Efficient POMDP Behavior Planning for Autonomous Driving in Dense Urban Environments using Multi-Step Occupancy Grid Maps

Chi Zhang[1], Shirui Ma[2], Muzhi Wang[3], Gereon Hinz[4], Alois Knoll[5]

***Abstract*— Driving through dense urban environments is difficult for autonomous vehicles because they must reason about the unknown intentions of a large number of road users while also dealing with a variety of uncertain information, such as sensor noise and inaccurate predictions. The partially observable Markov decision process (POMDP) is a systematic method for planning optimal policies in stochastic environments. However, using POMDP in scenarios with a large number of road users necessitates significant computational effort. In this paper, we propose a scalable online POMDP behavior planner for autonomous driving in dense urban environments. We enable intention-aware POMDP planning while considering uncertainties by using Multi-step Occupancy Grid Maps (MOGM) to represent the current and predicted states of surrounding road users, as well as their uncertain intentions. Furthermore, MOGM is applied to create a more computationally efficient POMDP model by condensing the state space and reducing the number of calculations used for collision checks. We show by numerical experiments that our approach is computationally more efficient. We demonstrate that our approach is able to navigate a dense urban scenario involving a large number of road users.**

I. INTRODUCTION

In recent years, progress has been made in the fields of navigation, perception, decision-making, and motion planning towards human-level intelligence in autonomous vehicles. In comparison to an autonomous vehicle, a human can drive well in complex environments by better reasoning about other road users' hidden intentions and predicting stochastic future interactions with them. This human ability enables anticipative driving behavior, which is hard for a computer to achieve despite its shorter reaction time. One key challenge for behavior planning is to reason about other traffic participants' intentions. The complexity of reasoning grows when considering noisy measurements and long-term prediction errors. As shown in Fig. 1, the ego vehicle must plan driving policies that take into account interactions influenced by the various future intentions of other vehicles.

Planning under uncertainty is a promising way to address the above-mentioned challenges. The partially observable Markov decision process (POMDP) is a systematic method for planning optimal policies in a stochastic environment [1]. However, due to the curses of dimensionality and history [2], POMDP is difficult to apply in densely populated urban environments. Online POMDP planning algorithms construct a belief tree from a start state only to reachable states to approximate the optimal policy and thus reduce computational complexity [3]–[5]. Extensive works show the progress made in applying POMDP to autonomous vehicles to handle various driving scenarios in urban environments which require the reasoning of intentions and the consideration of different uncertainties [6]–[24]. Most approaches so far have been demonstrated in environments with a few road users. To apply the POMDP planner in highly dynamic and dense urban environments with large numbers of road users, its scalability needs to be investigated.

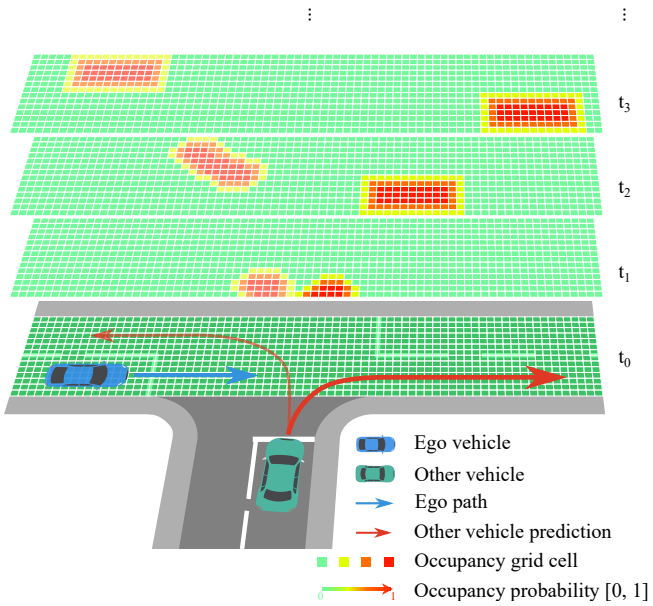

Fig. 1. Example of an MOGM with uncertain measurements, predictions, and intentions. Each green layer is one grid map of the MOGM at each time step. The color of the grid cell indicates the risk of this grid cell being occupied by a road user at this time step. In this example, another vehicle (green car) intends to drive into the junction. This vehicle has two possible intentions: turning left or turning right. Both intentions and their corresponding predicted states are represented in the MOGM by setting the grid cells as occupied (red grids) or possibly occupied (orange and yellow grids).

In this study, we focus on improving the scalability of a POMDP behavior planner by introducing an efficient MOGM-based POMDP model. The online POMDP solver relies on Monte Carlo sampling to construct the belief tree from the current state to the reachable state. A POMDP model is required as a black box simulator to generate a new episode, which includes transitioning from one state to the next and checking whether a state is a terminal state [5]. Such

[1]Chi Zhang, [2]Shirui Ma, [3]Muzhi Wang are with ZF Friedrichshafen AG, Friedrichshafen, Germany. (chi.zhang@zf.com, shirui.ma@zf.com, muzhi.wang@zf.com)

[4]Gereon Hinz, [5]Alois Knoll are with Chair of Robotics, Artificial Intelligence and Real-time Systems, Technical University of Munich, Germany. (gereon.hinz@tum.de, knoll@mytum.de)

a model needs to be called thousands to millions of times in a single planning cycle. Therefore, a more computationally efficient POMDP model can enhance the scalability of the POMDP planner.

Previous approaches typically build the POMDP state space and model using a feature-based representation of the environment, where the features of each road user, e.g., position, orientation and velocity, are set up directly in the state space. For each state transition of the ego vehicle and road users, a collision check is performed by simulating the interaction between each ego vehicle and road user pair, using the features defined in the state space. When the number of road users increases, the computational efforts of the collision checks also increase. Hence, feature-based approaches become increasingly inefficient for large numbers of road users. Increasing the efficiency of the collision check can reduce the computational effort for solving the POMDP model.

As mentioned in [16] and [22], a large number of future states of road users can be calculated in advance and saved in a lookup table to improve the efficiency of the planner. We extend this concept by using MOGM to represent all future states and probabilistic intentions of other road users. MOGM contains multiple occupancy grid maps (OGMs) to represent the free and potential collision areas at each planning time step, up to the planning horizon. An OGM discretizes the road surface into grid cells indicating the state of a particular area, i.e., occupied or not occupied. We reduce the computational demands of the collision check by applying MOGM to check whether the ego vehicle is located in areas that are occupied by other road users.

To enable the planning of driving policies in the uncertain environment, we consider the uncertain measurements and predictions of road users with extended occupancy area by using a two-dimensional Gaussian approximation. Their multiple intentions and the estimated probabilities are incorporated by extending the occupied grid cells to save the uncertain intention information. Finally, in the POMDP solver, the MOGM-based POMDP model is used as the black box simulator to build the belief tree and obtain optimized driving policies.

The main contributions of this study are:

- an extension of the MOGM that includes estimation of road users' uncertain intentions to enable intention-aware planning based on grid maps,
- the introduction of an efficient MOGM-based POMDP model for reducing computational effort, which improves the scalability and the efficiency of the POMDP planner,
- the evaluation and analysis of computation time for our MOGM-based POMDP planner in scenarios with large numbers of road users.

The remainder of this paper is organized as follows: Works related to environment representation and uncertainty-aware POMDP planning are presented in Sec. II. In Sec. III, the generation of the MOGM and its combination with a POMDP model to plan driving strategies are introduced. Sec. IV explains how the POMDP problem is solved. Evaluations are shown in Sec. V. Finally, the conclusions and future work are summarized in Sec. VI.

II. RELATED WORK

In this section, we compare feature-based and grid-based approaches to representing the environment and predictions. Following that, the progress of POMDP-based planners that consider various sources of uncertainty is summarized and discussed.

A. Environment Representation

The perception and prediction module of an autonomous driving function provides the current and predicted environment information to the planning module [25]. The environment can be represented in different forms.

1) Feature-based method: The feature-based method applies a set of features to represent the environment. The features can be discrete, continuous, or a mixture of both. The discrete feature is used to classify detected objects into different categories such as vehicles, pedestrians or cyclists [26]. A list of bounding boxes over different time steps is frequently used to describe the current and future states of detected objects with continuous variables like position, orientation, and velocity [27]. The prediction for one object can also contain multiple intentions. In such cases, the feature list includes both discrete features such as intentions and estimated probabilities to indicate whether a vehicle intends to turn right or left, as well as continuous variables to represent the trajectories associated with each intention [28].

2) Grid-based method: Another common representation is OGM [29]. An OGM discretizes the road surface into grid cells, where each grid cell shows whether an obstacle is present. An OGM grid cell can be expanded to include more information, such as object classification, to allow for more sophisticated decision-making [30]. However, a single OGM is insufficient to represent moving objects in a dynamic environment. [31] extends the OGM to temporal occupancy grids with several layers, where each layer shows which cells were occupied during a specific time interval. MOGM is used to represent predictions, with each OGM representing a time step of the prediction [32].

B. Decision-making Under Uncertainty

POMDP is a general probabilistic method for decision-making under various uncertainties [1], which is widely applied in autonomous driving for handling complex driving scenarios in urban environments.

1) Uncertain measurements and predictions: Uncertain measurements caused by sensor noise are the most commonly considered source of uncertainty when modeling the POMDP as a behavior planner. The state uncertainty is typically represented by a Gaussian distribution [6], [8]. The prediction uncertainty increases over time. Several studies explicitly include prediction uncertainty in the POMDP planning process [9], [15].

2) Uncertain intention and interaction: Recent research focuses on optimizing driving strategies involving uncertain intentions and interactions with other road users, as well as providing driving strategies that account for possible future observations of scenarios. The ability of the POMDP formulation has been demonstrated for scenarios where the ego vehicle needs to negotiate and interact with other road users, such as on-ramp merging [6], lane changes [7], [10], unsignalized intersections [12]–[14], roundabouts [15], [16], crosswalks and bus stops [17], [18].

3) Uncertain appearance: Urban environments involve static and dynamic objects that limit the field of view (FoV) of the ego vehicle. Previous works expand the POMDP model to account for the uncertain probability of potentially occluded vehicles and pedestrians appearing in such environments [19]–[23]. The idea is to introduce phantom road users to represent the potentially existing pedestrians and vehicles which have certain probabilities of appearing from occluded areas. The phantom road users are incorporated into a POMDP formulation's state space and probabilistic transition model.

4) Scalability of the POMDP planner: POMDP problems face both the curse of dimensionality and the curse of history [2], making it very hard to scale up with a large POMDP model. However, recently developed open-source POMDP solvers based on Monte Carlo tree searches have been released to address this issue [3]–[5]. Parallel computing with CPU and GPU improves the POMDP solver efficiency even further [33]. The authors of [34] incorporate a learning-based heuristic to guide the tree construction process of the POMDP solver, enabling more efficient planning without searching too deep. Besides targeting solver, another direction for enhancing the scalability of POMDP planners is the model design. The authors of [24] reduce the computational cost of POMDP-based decision-making algorithms by utilizing domain knowledge in a policy tree. However, because the POMDP model and solver are coupled in one algorithm, their approach cannot benefit from further POMDP solver developments.

III. APPROACH

In this section, we first explain the structure and the generation of the MOGM. Secondly, the POMDP model based on the MOGM is introduced.

A. Multi-step Occupancy Grid Maps

We follow the assumption that the estimated intentions and their corresponding future states of each road user are fixed in one planning cycle [22]. This information is delivered from the prediction module of the autonomous vehicle, where the future states of other traffic participants are represented as bounding boxes. The corners of the bounding boxes are represented by the Frenet coordinates s and d, where s indicates the arc-length along the ego lane center, and d is the orthogonal distance to the ego lane center, as shown in Fig. 2.

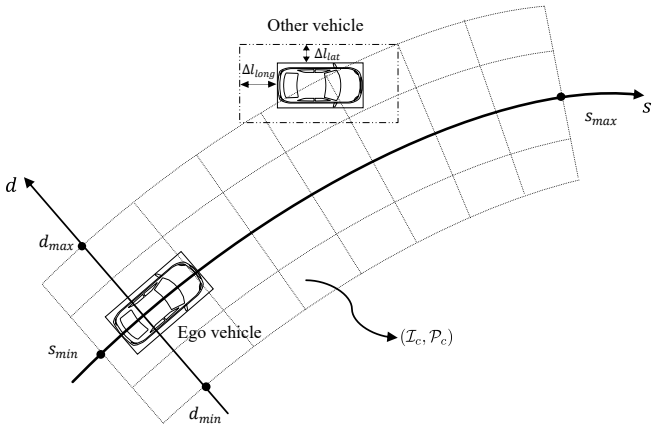

Fig. 2. Construction of the MOGM. The ego vehicle and the predicted states of other vehicles are represented by bounding boxes. Δl_{long} and Δl_{lat} indicate the extended risk area.

Using the intentions and bounding boxes, we construct an MOGM to represent all possible intentions and future states of road users. In order to account for uncertain measurements and predictions, we extend the bounding boxes of road users with risk areas approximated by a two-dimensional Gaussian function. The probabilities of the grid cells being occupied by the bounding boxes are stored in the MOGM.

1) Uncertainty approximation using Gaussian function: The occupancy probability p_c of an MOGM grid cell is set to one when it is occupied by bounding boxes. We consider the uncertainty of the measurements and predictions by extending the bounding box with risk areas (see Fig. 2). The occupancy probability p_c is approximated by a two-dimensional Gaussian function:

$$p_c = \exp\left(-\left(\frac{(\Delta l_{long})^2}{2{\sigma_{long}}^2} + \frac{(\Delta l_{lat})^2}{2{\sigma_{lat}}^2}\right)\right), \qquad (1)$$

where $\Delta l_{long}, \Delta l_{lat} \in \mathbb{R}$ are the longitudinal and lateral distances to the bounding box boundary, and the variances σ_{long}, $\sigma_{lat} \in \mathbb{R}$ determine the risk distribution around the road users' bounding boxes in the longitudinal and lateral directions, respectively. In this work, we choose $\sigma_{long} = 1$, and $\sigma_{lat} = 0.5$.

In order to improve the representation of the risk distribution around the bounding boxes, we extend the bounding box of a road user with N_b longitudinal and lateral extensions $\Delta\mathcal{L}_{long}$, $\Delta\mathcal{L}_{lat}$, where $N_b \in \mathbb{N}^+$, $\Delta\mathcal{L}_{long}, \Delta\mathcal{L}_{lat} \in \mathbb{R}^{N_b}$.

2) Multiple intentions: Every road user $i \in \{1, ..., N\}, N \in \mathbb{N}^+$, has been assigned a set of intentions $\mathcal{I}^i = \{\iota_1^i, ..., \iota_J^i\}$, where $J \in \mathbb{N}^+$ is the number of intentions of road user i (see Fig. 1). Each intention is associated with a set of predicted states over time. We store $\iota_j^i \in \mathcal{I}^i$, $j \in \{1, ..., J\}$ in the grid cells of the MOGM, if they geometrically overlap (i.e., are occupied) with the bounding boxes of predicted states from the intention ι_j^i.

3) MOGM generation: We discretize the planning time horizon $H \in \mathbb{R}^+$ into $K \in \mathbb{N}$ steps. As shown in Fig. 2, the

Algorithm 1: OGM generation for time step k

Input : N road users, longitudinal extensions $\Delta\mathcal{L}_{long}$, lateral extensions $\Delta\mathcal{L}_{lat}$
Output: OGM $\mathcal{M}_k$

1 $\mathcal{M}_k \leftarrow$ `initializeOGM`(k)
2 **foreach** *road user* $i \in \{1, ..., N\}$ **do**
3 $\quad\mathcal{I}^i \leftarrow$ `getIntentions`(i)
4 $\quad$**foreach** $\iota \in \mathcal{I}^i$ **do**
5 $\quad\quad b_k \leftarrow$ `getPredictedState`(ι, k)
6 $\quad\quad\mathcal{B}_k \leftarrow$ `extendBoxes`$(b_k, \Delta\mathcal{L}_{long}, \Delta\mathcal{L}_{lat})$
7 $\quad\quad$**foreach** $c_k \in \mathcal{M}_k$ **do**
8 $\quad\quad\quad\mathcal{I}_{c_k}, \mathcal{P}_{c_k} \leftarrow c_k$
9 $\quad\quad\quad$**if** `isOccupied`(c_k, b_k) **then**
10 $\quad\quad\quad\quad p_{c_k,\iota} \leftarrow 1$
11 $\quad\quad\quad\quad$`addToList`$(\mathcal{I}_{c_k}, \iota)$
12 $\quad\quad\quad\quad$`addToList`$(\mathcal{P}_{c_k}, p_{c_k,\iota})$
13 $\quad\quad\quad$**else if** `isOccupied`$(c_k, \mathcal{B}'_k)$ **then**
14 $\quad\quad\quad\quad p_{c,\iota} \leftarrow$ `maxRisk`$(\Delta\mathcal{L}_{long}, \Delta\mathcal{L}_{lat})$
15 $\quad\quad\quad\quad$`addToList`$(\mathcal{I}_{c_k}, \iota)$
16 $\quad\quad\quad\quad$`addToList`$(\mathcal{P}_{c_k}, p_{c_k,\iota})$
17 $\quad\quad\quad$**end if**
18 $\quad\quad$**end foreach**
19 $\quad$**end foreach**
20 **end foreach**
21 **return** $\mathcal{M}_k$

OGM $\mathcal{M}_k$ at time step $k \in \{0, ..., K\}$ is constructed on the ego lane center. In the longitudinal direction, $\mathcal{M}_k$ starts at s_{min} and ends at s_{max} along the ego lane center, while in the lateral direction, $\mathcal{M}_k$ covers d_{max} to the left and d_{min} to the right of the ego lane center.

A cell in $\mathcal{M}_k$ can be occupied, partly occupied, or unoccupied. We consider a partly occupied cell as occupied. Formally, $\mathcal{M}_k$ can be represented as a matrix of tuples:

$$\mathcal{M}_k = \begin{bmatrix} c_k(1,1) & \dots & c_k(1,Y) \\ \vdots & \ddots & \vdots \\ c_k(X,1) & \dots & c_k(X,Y) \end{bmatrix}, \tag{2}$$

where $X, Y \in \mathbb{N}^+$ indicate the number of rows and columns of $\mathcal{M}_k$. Each grid cell $c_k \in \mathcal{M}_k$ stores a tuple $(\mathcal{I}_{c_k}, \mathcal{P}_{c_k})$, $\mathcal{I}_{c_k} \subseteq \bigcup_{i=0}^{N} \mathcal{I}^i$ is a set of intentions, from which the predicted states occupy the grid cell c_k, and $\mathcal{P}_{c_k}$ is a set of occupancy probabilities calculated by the predicted states from $\mathcal{I}_{c_k}$.

The generation of $\mathcal{M}_k$ is described in Alg. 1. Initially, we construct an empty OGM such that intentions $\mathcal{I}_{c_k}$ and occupancy probability $\mathcal{P}_{c_k}$ of each grid cell are empty sets (line 1). We obtain the predicted states at time step k from all intentions of each road user (see lines 2 to 4). Each predicted state is represented by a bounding box b_k (line 5). We extend b_k to a set of bounding boxes $\mathcal{B}_k$ with N_b longitudinal and lateral extensions $\Delta\mathcal{L}_{long}$, $\Delta\mathcal{L}_{lat}$ (line 6). For each grid cell c_k that is occupied by b_k, the intention and occupancy probability $p_{c_k,\iota} = 1$ are inserted into the intention set $\mathcal{I}_{c_k}$ and occupancy probability set $\mathcal{P}_{c_k}$, respectively (see lines 7 to 12). If c_k is not occupied by b_k, we check whether any bounding boxes in $\mathcal{B}_k$ occupy the grid cell c_k and insert the maximum occupancy probability $p_{c_k,\iota}$ calculated by (1) to $\mathcal{P}_{c_k}$ and the intention to $\mathcal{I}_{c_k}$ (see lines 13 to 16).

B. POMDP Behavior Planning

1) Preliminaries: A POMDP is a probabilistic method that models the sequential decision process of a system (often denoted as agent) under uncertain conditions. A POMDP is defined by the tuple $(\mathcal{X}, A, O, T, Z, R, \gamma)$ where $\mathcal{X}$, A, O represent the state, action, and observation spaces, respectively. The transition model T is a conditional probability function $T(\chi, a, \chi') = P(\chi' \mid \chi, a)$ modeling the probability of a system transition from the state $\chi \in \mathcal{X}$ to the state $\chi' \in \mathcal{X}$ when action $a \in A$ is executed. Similarly, the observation model Z is a conditional probability function $Z(o, a, \chi') = P(o \mid \chi',\ a)$ describing the probability of receiving observation $o \in O$ after taking action $a \in A$ and transitioning to state $\chi' \in \mathcal{X}$. The reward $R(\chi,\ a)$ is the immediate reward generated by performing the action $a \in A$ from the state $\chi \in \mathcal{X}$. Finally, a factor $\gamma \in [0, 1)$ discounts future rewards [35].

In a partially observable environment, the agent has only incomplete knowledge of the system state. Hence, a belief state $\beta(\chi)$ is maintained to reflect its internal knowledge of the system and estimate the true state. The policy $\pi : \beta \rightarrow a$ is a mapping from a belief β to an action a. Therefore, the solution to a POMDP problem is an optimal policy π^* that maximizes the expectation of accumulated rewards over time:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{k=0}^{\infty} \gamma^{t_k} R\left(\chi_{t_k}, \pi\left(\beta_{t_k}\right)\right) \mid \beta_0, \pi\right]. \tag{3}$$

2) State and observation space: The state space is a representation of the driving scenario in the POMDP model. It contains the state of the ego vehicle χ_{ego} and the state of other road users χ_i. The state $\chi \in \mathcal{X}$ is a vector $\chi = [\chi_{ego}, \chi_1, \chi_2, ..., \chi_N]^T$, where $\chi_{ego} = [x_{ego}, y_{ego}, \theta_{ego}, v_{ego}, r_{ego}]^T$ is the ego state which includes the position (x_{ego}, y_{ego}), orientation θ_{ego}, speed v_{ego} and intended driving path r_{ego}. The states χ_i with $i \in \{1, 2, \dots, N\}, N \in \mathbb{N}^+$ represent road users in the scenario. Optionally, phantom objects can be added to the state space to allow the model to plan driving behavior while taking into account the uncertain probability that potentially occluded traffic participants will appear [21]. We introduce an MOGM-based state model to represent other road users. The state of a road user i is defined as $\chi_i = \iota_j^i,\ \iota_j^i \in \mathcal{I}^i$, $j \in \{1, ..., J\},\ J \in \mathbb{N}^+$, where ι_j^i is one of the intentions of the road user i, e.g., turn left or right at an intersection.

3) Action: We follow the path-velocity decomposition method [36] for planning the longitudinal driving policies along the intended ego path r_{ego}. Possible driving behaviors of the ego vehicle are represented by longitudinal accelerations: $A = \{+1.5\ m/s^2, 0\ m/s^2, -1.5\ m/s^2\}$.

Algorithm 2: Calculate collision probability at time step k

Input : OGM $\mathcal{M}_k$, State $\chi = [\chi_{ego}, \chi_1, ..., \chi_N]^T$
Output: Collision probability $p_{collision}$

1 $b_{ego} \leftarrow \texttt{buildBoundingBox}(\chi_{ego})$
2 $\mathcal{C}_{k,ego} \leftarrow \texttt{findOccupiedCells}(b_{ego}, \mathcal{M}_k)$
3 $\mathcal{P}_{collision} \leftarrow \emptyset$
4 **foreach** *occupied cell* $c_{k,ego} \in \mathcal{C}_{k,ego}$ **do**
5 $\mathcal{I}_{c_k}, \mathcal{P}_{c_k} \leftarrow c_{k,ego}$
6 **foreach** *road user* $i \in \{1, \dots, N\}$ **do**
7 $\iota \leftarrow \chi_i$
8 **if** $\iota \in \mathcal{I}_c$ **then**
9 $p_{c_k,\iota} \leftarrow \mathcal{P}_{c_k}$
10 $\texttt{addToList}(\mathcal{P}_{collision}, p_{c_k,\iota})$
11 **end if**
12 **end foreach**
13 **end foreach**
14 $p_{collision} \leftarrow \texttt{max}(\mathcal{P}_{collision})$
15 **return** $p_{collision}$

4) State transition with MOGM: The state transition from the current state χ to the next state χ' is determined by the transition models of the ego vehicle and other road users. The intended ego path remains unchanged such that $r'_{ego} = r_{ego}$. We apply a point mass model in (4) to predict the ego movement along the intended ego path r_{ego}. The position of the ego vehicle s_{ego} is the arc-length along r_{ego}. The new ego position s'_{ego} and velocity v'_{ego} are predicted by the chosen action a and the step size Δt:

$$\begin{bmatrix} s'_{ego} \\ v'_{ego} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_{ego} \\ v_{ego} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} a. \quad (4)$$

The x'_{ego}, y'_{ego} and θ'_{ego} in the updated ego state χ'_{ego} are obtained by getting the Cartesian position from r'_{ego} according to the path geometry based on the position s'_{ego}. With our assumption that a road user's intention does not change within a planning cycle, the states of road users χ_i remain the same in all planning steps.

5) Terminal condition check with MOGM: When a state transitions to the next state χ', a terminal condition check for χ' is needed. We consider a state to be terminated when the planning horizon is reached during the construction of the belief tree, or a collision occurs between the ego vehicle and a road user. Alg. 2 performs the calculation of the collision probability. Firstly, the ego state is matched to the occupied cells $\mathcal{C}_{k,ego}$ in the OGM $\mathcal{M}_k$ at time step k (see lines 1 and 2). Then, the collision probability is checked for every cell $c_{k,ego}$ that is occupied by the ego vehicle (line 4). For each road user i, we get the intention ι from χ_i (see lines 6 and 7). If $c_{k,ego}$ contains the intention ι of road user i, the collision probability $p_{c_k,\iota}$ is saved in the list $\mathcal{P}_{collision}$ (see lines 8 to 10). Finally, the maximum collision probability $p_{collision}$ from the list $\mathcal{P}_{collision}$ is returned (see lines 14 and 15). If $p_{collision} = 1$, the state is denoted as the terminal state.

6) Reward: The reward function of this study includes the objectives of safety, speed, and comfort:

$$R = R_{safety} + R_{speed} + R_{comfort}. \quad (5)$$

To emphasize safety, we assign rewards depending on the collision probability $p_{collision}$:

$$R_{safety} = \begin{cases} -100000, & \text{if } p_{collision} = 1 \\ -10000 \cdot p_{collision}, & \text{otherwise.} \end{cases} \quad (6)$$

The ego vehicle is also encouraged to maintain the desired velocity $v_{desired}$ following the ego path r_{ego}:

$$R_{speed} = \begin{cases} -200\left(v_{desired} - v_{ego}\right), & \text{if } v_{desired} \geq v_{ego} \\ -2000\left|v_{desired} - v_{ego}\right|, & \text{otherwise.} \end{cases} \quad (7)$$

To obtain comfortable driving policies, acceleration is penalized with $R_{comfort} = -300 \cdot a^2$.

IV. SOLVING POMDP MODEL

We apply the Toolkit for approximating and Adapting POMDP solutions In Real time (TAPIR) to solve our MOGM-based POMDP model [5]. TAPIR is based on Monte Carlo sampling to construct a belief tree to approximate the optimal policy online. In this section, we briefly summarize the core idea of the TAPIR algorithm and how the approximated optimal policy is obtained using our MOGM-based POMDP model.

A. Belief State Tracking

We apply unweighted particles to maintain the internal belief of the POMDP planner about the current belief state [14]. In the beginning of the belief tree construction, the initial belief b_0 is initialized by sampling particles $\chi \in \mathcal{X}$ to reflect the system's expectation of the possible states of the ego vehicle and road users. To determine which intention of a road user is contained in a particle, a sample is drawn based on the probability distribution of the different intentions of each road user. The position and the dimension of road users can be accessed from the MOGM; only the intention needs to be sampled to generate the initial belief β_0 in the MOGM-based state space.

B. Belief Tree Construction

The belief tree is constructed based on the Monte Carlo tree search process [5]. A particle is randomly chosen from the initial belief β_0 as the start of an episode. An episode contains a state, action, reward, and observation. By invoking the POMDP model, the chosen particle transitions continuously from its current state to other possible states, resulting in new episodes. Our MOGM-based POMDP model serves as the black box simulator for generating new episodes to construct the belief tree. Finally, the approximated optimal policy π is obtained from the belief tree at the end of each planning cycle.

TABLE I

APPLIED PARAMETERS IN THE SIMULATION.

Parameter	Value	Description
d_{min}	$-5\ m$	Min. lat. coordinate on the MOGM
d_{max}	$5\ m$	Max. lat. coordinate on the MOGM
s_{min}	$-20\ m$	Min. long. coordinate on the MOGM
s_{max}	$80\ m$	Max. long. coordinate on the MOGM
X	200	Number of the MOGM rows
Y	20	Number of the MOGM columns
$\Delta\mathcal{L}_{long}$	$\{1.5, 3.0\}$	Long. extensions of bounding box
$\Delta\mathcal{L}_{lat}$	$\{0.5, 1.0\}$	Lat. extensions of bounding box
F	$2\ Hz$	Planning frequency
γ	0.95	Discount factor
D	10	Maximal tree depth
H	$10\ s$	Planning horizon
K	11	Number of planning time steps

V. EVALUATION

We evaluate our approach on a system with an Intel Core i7-6820HQ CPU running at 2.70 GHz. The evaluations are carried out in a proprietary simulation platform that allows configuration of static buildings, dynamic vehicles, and pedestrians. To eliminate the influence of other road users' intelligent behavior in the evaluation, we control them via predefined behaviors that do not consider collision avoidance. We use our previous feature-based POMDP model (TAPIR-POMDP) [21] as a baseline planner. Our approach – a POMDP behavior planner with the MOGM-based POMDP model – is denoted as MOGM-POMDP. To demonstrate the planner's ability to plan driving policies in a dense urban environment with occlusions, we create the MOGMOA-POMDP by extending the MOGM-based POMDP model with occlusion awareness, as proposed in [21]. We provide a supplementary video[1] of the evaluated scenarios.

The parameters applied in the simulation are listed in Tab. I. Firstly, we evaluate the average generation time of MOGM with different numbers of objects. Then, we set up a dense traffic scenario to carry out a qualitative evaluation for comparing driving strategies using the aforementioned planners. To compare the computational efficiency of the MOGM-POMDP with that of the TAPIR-POMDP, we measure the computation time for generating a new episode when constructing the belief tree with different numbers of road users. The speedup is defined as the average computation time of the MOGM-POMDP compared to that of the TAPIR-POMDP. Furthermore, we illustrate the overall performance by comparing the number of active belief nodes in the belief tree within a planning cycle. The speedup is used to compare the average number of active nodes of the MOGM-POMDP to that of the TAPIR-POMDP.

A. MOGM Generation Time

To evaluate the computation time of the MOGM, we set up 10 to 70 objects located on the ego lane. In this evaluation, we configure the simulation such that all objects are detected and provided to generate the MOGM. Tab. II shows the average generation time of MOGM in one planning cycle. It can be seen that the MOGM generation time ranges from 6.4 ms to 43.2 ms, increasing linearly with the number of objects involved in the scenario.

[1]Video: https://github.com/GitChiZhang/MOGM-POMDP

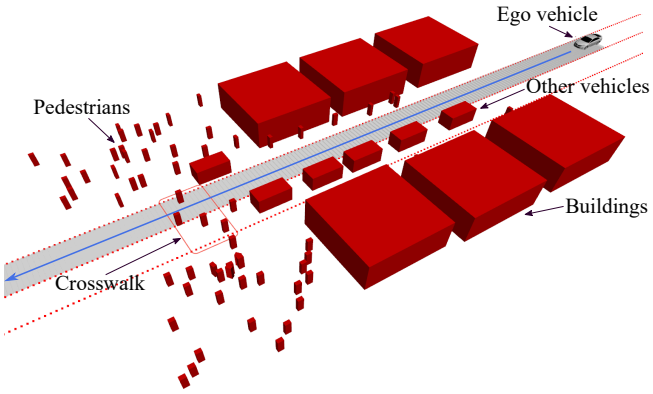

Fig. 3. The ego vehicle intends to drive through the dense urban environment with buildings along the street and many pedestrians moving around. The total number of static and dynamic objects in the scenario is 80.

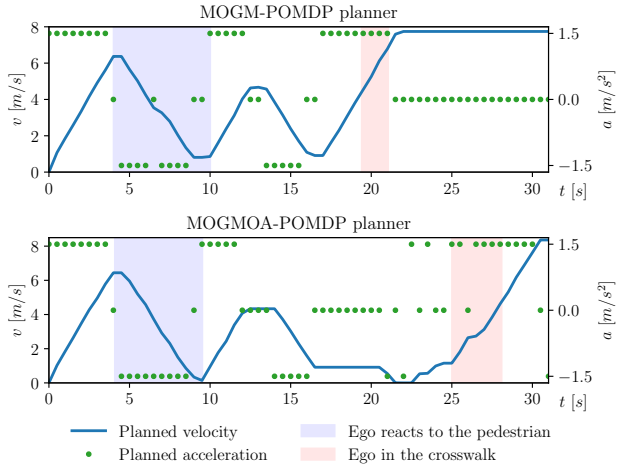

Fig. 4. Comparison of planned driving strategies for handling the dense urban environment.

B. Performance in a Dense Urban Environment

Fig. 3 illustrates a dense urban scenario with buildings alongside the street and lots of pedestrians walking around. The total number of static and dynamic objects in the scenario is 80. In this scenario, there are several pedestrians crossing a crosswalk. One pedestrian, not on the crosswalk, will abruptly cross the street from the middle of the road. The walking speed of all pedestrians ranges from 0.1 m/s to 1.5 m/s to reflect the diversity of the traffic situations.

TABLE II

MOGM GENERATION TIMES.

Num. object	10	20	30	40	50	60	70
Aver. time (ms)	6.4	11.4	18.2	25.0	29.7	36.4	43.2

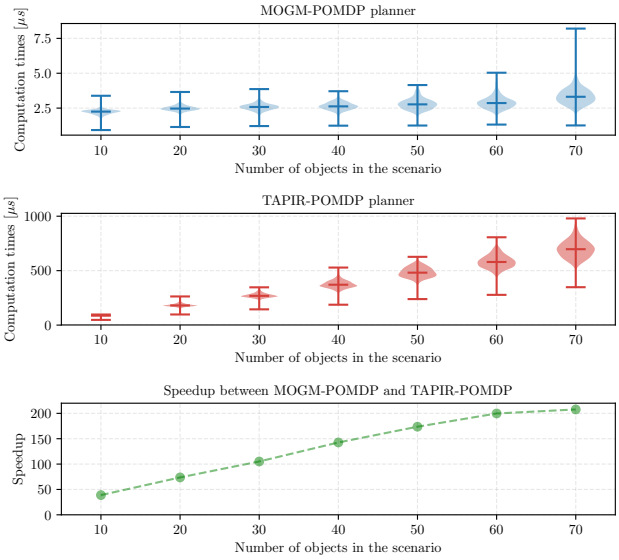

Fig. 5. Comparison of the computation time of the MOGM-POMDP model and the TAPIR-POMDP model for generating a new episode. The speedup is calculated by comparing the average computation time of the MOGM-POMDP to that of the TAPIR-POMDP.

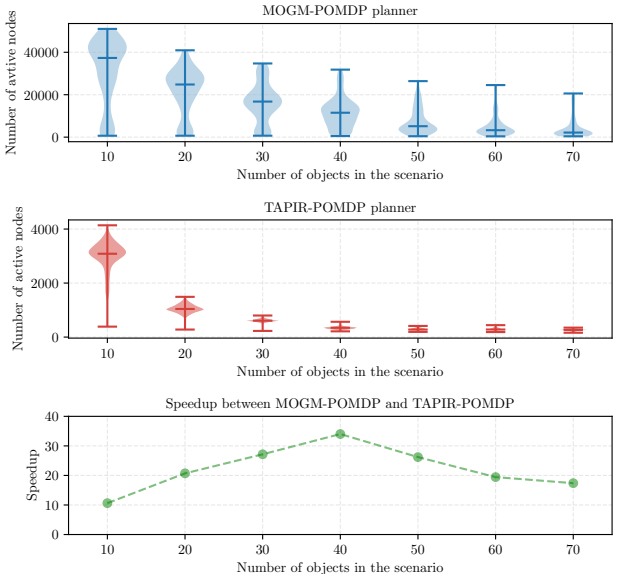

Fig. 6. Comparison between the active nodes of the MOGM-POMDP and TAPIR-POMDP in the belief tree construction. The speedup is calculated by comparing the average number of active nodes of the MOGM-POMDP to that of the TAPIR-POMDP.

Apart from the pedestrians, there is a parked vehicle near the crosswalk that obscures the ego vehicle's FoV.

We perform a qualitative evaluation of the planners in the aforementioned scenario. Due to a large number of road users in the scenario, the TAPIR-POMDP is unable to sample enough belief nodes to provide safe policies for the given planning frequency of 2 Hz. In contrast to the TAPIR-POMDP, both the MOGM-POMDP and MOGMOA-POMDP can provide safe driving policies to navigate through the scenario. Fig. 4 shows the acceleration and velocity profile of the planned driving strategies. At time $t = 4.10\ s$, the MOGM-POMDP stops accelerating and reacts to the abruptly crossing pedestrian by further reducing the speed. The MOGM-POMDP then waits for pedestrians crossing the road before driving through the crosswalk. The MOGMOA-POMDP demonstrates various driving strategies with occlusion awareness for dealing with the same scenario. At time $t = 17.50\ s$, the MOGMOA-POMDP continues to drive cautiously to reduce the risk of pedestrians suddenly appearing from the blind spot of the ego vehicle. Once the ego vehicle has a sufficient FoV, it accelerates and drives through the crosswalk.

C. Effect on Episode Generation

In this evaluation, we measure the computation time for generating a new episode in the tree construction. As shown in Fig. 5, the MOGM-POMDP significantly reduces the average computation time for generating a new episode. The MOGM-POMDP's mean computation time varies from 2.5 μs to 3.5 μs depending on the number of objects in the state space, whereas the TAPIR-POMDP has a mean computation time of 50 μs to 600 μs. It can also be seen that increasing the number of objects in the state space has a larger impact on the TAPIR-POMDP. Consequently, the MOGM-POMDP achieves a speedup of 50 to 200 times over the TAPIR-POMDP in calculating a new episode. There are two reasons for the improvement. First, the state space and observation space of the MOGM-POMDP are smaller than those of the TAPIR-POMDP, which reduces the matching time of a particle to the belief tree. Second, unlike the TAPIR-POMDP, which performs collision checks by simulating the interaction between each ego vehicle and object pair, the MOGM-POMDP only checks the new ego vehicle state within the corresponding occupancy map.

D. Effect on Belief Tree Construction

The evaluation result in Fig. 6 shows that for both the MOGM-POMDP and TAPIR-POMDP, the number of active belief nodes decreases as the number of road users involved in the state space increases. The performance of TAPIR-POMDP drops dramatically when more than ten objects are considered. The MOGM-POMDP outperforms the TAPIR-POMDP by a factor of 10 to 35. Major performance gains are seen when the number of objects is between 20 and 60 with a speedup of between 20 and 35.

VI. CONCLUSION AND FUTURE WORK

In this study, we present a scalable POMDP planner that can plan safe driving behaviors in a dense urban environment. We incorporate the surrounding road users' uncertain measurements, predictions, and intentions into the MOGM to enable intention-aware planning with uncertain risk consideration. Furthermore, an efficient POMDP model based on the MOGM is introduced as a black box simulator for the POMDP online solver. According to the evaluation results, the MOGM-based POMDP model is approximately 50 times faster than the baseline POMDP model. The overall performance of our approach improves on the performance

of the baseline POMDP planner by a factor of at least 10. Simulations show that the MOGM-POMDP planner can plan safe driving policies in an urban environment involving a large number of road users.

In this work, the MOGM is discretized into grids of fixed size. We plan to improve the efficiency of generating the MOGM by introducing a variable resolution of the grid. When the grid cells have a larger distance to the ego vehicle or are not occupied by any other road users, their size can be increased, whereas smaller grid cells can be used to more accurately represent occupancy by objects. Furthermore, we plan to validate our approach using real vehicle data.

REFERENCES

[1] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[2] J. Pineau, G. Gordon, and S. Thrun, "Anytime point-based approximations for large POMDPs," *Journal of Artificial Intelligence Research*, vol. 27, pp. 335–380, 2006.

[3] D. Silver and J. Veness, "Monte-carlo planning in large POMDPs," *Proc. of advances in neural information processing systems*, vol. 23, 2010.

[4] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "Despot: Online POMDP planning with regularization," *Proc. of advances in neural information processing systems*, vol. 26, pp. 1772–1780, 2013.

[5] D. Klimenko, J. Song, and H. Kurniawati, "Tapir: A software toolkit for approximating and adapting POMDP solutions online," in *Proc. of the Australasian Conference on Robotics and Automation, Melbourne, Australia*, vol. 24, 2014.

[6] J. Wei, J. M. Dolan, J. M. Snider, and B. Litkouhi, "A point-based MDP for robust single-lane autonomous driving behavior under uncertainties," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 2586–2592.

[7] S. Ulbrich and M. Maurer, "Towards tactical lane change behavior planning for automated vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2015, pp. 989–995.

[8] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1288–1302, 2014.

[9] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 1671–1678.

[10] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, "A belief state planner for interactive merge maneuvers in congested traffic," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 1617–1624.

[11] W. Song, B. Su, G. Xiong, and S. Li, "Intention-aware decision making in urban lane change scenario for autonomous driving," in *Proc. of the IEEE International Conference on Vehicular Electronics and Safety*, 2018, pp. 1–8.

[12] M. Bouton, A. Cosgun, and M. J. Kochenderfer, "Belief state planning for autonomously navigating urban intersections," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 825–830.

[13] M. Sefati, J. Chandiramani, K. Kreisköther, A. Kampker, and S. Baldi, "Towards tactical behaviour planning under uncertainties for automated vehicles in urban scenarios," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1–7.

[14] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller, "Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction," *Journal of IEEE Transactions on Intelligent Vehicles*, vol. 3, pp. 5–17, 2018.

[15] W. Liu, S.-W. Kim, S. Pendleton, and M. H. Ang, "Situation-aware decision making for autonomous driving on urban road using online POMDP," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2015, pp. 1126–1133.

[16] H. Bey, M. Sackmann, A. Lange, and J. Thielecke, "POMDP planning at roundabouts," in *Proc. of the IEEE Intelligent Vehicles Symposium Workshops*, 2021, pp. 264–271.

[17] M. Schratter, M. Bouton, M. J. Kochenderfer, and D. Watzenig, "Pedestrian collision avoidance system for scenarios with occlusions," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 1054–1060.

[18] S. M. Thornton, F. E. Lewis, V. Zhang, M. J. Kochenderfer, and J. C. Gerdes, "Value sensitive design for autonomous vehicle motion planning," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 1157–1162.

[19] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, and C. Stiller, "A POMDP maneuver planner for occlusions in urban scenarios," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 2172–2179.

[20] Y. Wang, Y. Guo, and J. Wang, "A hierarchical planning framework of the intersection with blind zone and uncertainty," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2021, pp. 687–692.

[21] C. Zhang, F. Steinhauser, G. Hinz, and A. Knoll, "Improved occlusion scenario coverage with a POMDP-based behavior planner for autonomous urban driving," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2021, pp. 593–600.

[22] X. Lin, J. Zhang, J. Shang, Y. Wang, H. Yu, and X. Zhang, "Decision making through occluded intersections for autonomous driving," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2449–2455.

[23] P. Schörner, L. Töttel, J. Doll, and J. M. Zöllner, "Predictive trajectory planning in situations with hidden road users using partially observable Markov decision processes," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2019, pp. 2299–2306.

[24] L. Zhang, W. Ding, J. Chen, and S. Shen, "Efficient uncertainty-aware decision-making for automated driving using guided branching," in *Proc. of the Int. Conf. on Robotics and Automation*, 2020, pp. 3291–3297.

[25] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[26] N. Zimmerman, C. Schlenoff, and S. Balakirsky, "Implementing a rule-based system to represent decision criteria for on-road autonomous navigation," in *Proc. of the AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems*, 2004.

[27] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net," in *Proc. of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 3569–3577.

[28] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *Proc. of the Int. Conf. on Robotics and Automation*, 2019, pp. 2090–2096.

[29] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," 2013, arXiv preprint arXiv:1304.1098.

[30] M. Schreier, "Environment representations for automated on-road vehicles," *at-Automatisierungstechnik*, vol. 66, no. 2, pp. 107–118, 2018.

[31] D. Arbuckle, A. Howard, and M. Mataric, "Temporal occupancy grids: A method for classifying the spatio-temporal properties of the environment," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, 2002, pp. 409–414.

[32] N. Mohajerin and M. Rohani, "Multi-step prediction of occupancy grid maps with recurrent neural networks," in *Proc. of the the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 600–10 608.

[33] P. Cai, Y. Luo, D. Hsu, and W. S. Lee, "Hyp-despot: A hybrid parallel algorithm for online planning under uncertainty," *The International Journal of Robotics Research*, vol. 40, pp. 558–573, 2021.

[34] P. Cai, Y. Luo, A. Saxena, D. Hsu, and W. S. Lee, "Lets-drive: Driving in a crowd by learning from tree search," 2019, arXiv preprint arXiv:1905.12197.

[35] M. J. Kochenderfer, *Decision making under uncertainty: theory and application*. MIT press, 2015.

[36] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The international journal of robotics research*, vol. 5, no. 3, pp. 72–89, 1986.