2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Scalable Decision Making with Sensor Occlusions for Autonomous Driving

Maxime Bouton,[1] Alireza Nakhaei,[2] Kikuo Fujimura,[2] and Mykel J. Kochenderfer[1]

***Abstract*— Autonomous driving in urban areas requires avoiding other road users with only partial observability of the environment. Observations are only partial because obstacles can occlude the field of view of the sensors. The problem of robust and efficient navigation under uncertainty can be framed as a partially observable Markov decision process (POMDP). In order to bypass the computational cost of scaling the formulation to avoiding multiple road users, this paper demonstrates a decomposition method that leverages the optimal avoidance strategy for a single user. We evaluate the performance of two POMDP solution techniques augmented with the decomposition method for scenarios involving a pedestrian crosswalk and an intersection.**

I. Introduction

Autonomous vehicles are expected to drive through crowded urban environments efficiently and reliably. Such environments often involve physical obstacles that occlude the field of view of the sensors, which makes avoiding other road users challenging. In this paper, we focus on an unsignalized crosswalk and an unsignalized intersection with traffic in both directions. Both scenarios have areas that are not observable by the car due to an obstacle on the side of the road as illustrated in Fig. 1. Providing appropriate behavior requires scalable algorithms that reason about the potential locations of pedestrians and other vehicles along with their motion over time.

Prior approaches for accounting for sensor occlusions involved resolving the worst case scenario. By specifying an upper bound on the speed of the agents potentially coming from the occluded area, one can provide a speed profile that will guarantee safety [1]. Another approach is to set a threshold on the level of uncertainty acceptable to make safe decisions [2]. Providing such tailored solutions is unlikely to scale to complex tasks and does not address the problem of avoiding an unknown number of users. In cases where the uncertainty is high, a worst-case approach might lead to suboptimal behaviors such as standing still if the car fails to find a feasible strategy to avoid collisions [3].

Anticipating uncertainty and optimally trading off between gathering information and reaching the goal can be addressed by modeling the problem as a partially observable Markov decision process (POMDP) [4]. This mathematical framework for sequential decision making under uncertainty has been successfully applied to autonomous driving scenarios [5]–[8]. Brechtel [5] introduces sensor occlusions in the POMDP formulation but assumes that the number of agents is known and constant. A similar crosswalk scenario with a single pedestrian has also been considered [9].

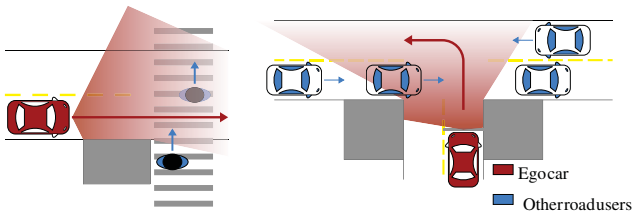

Fig. 1: The ego car (in red) must decide on the acceleration to apply along the given path. The gray blocks are obstacles occluding the ego car field of view and the shaded area represents the part of the environment that is not occluded. Occlusion checking is done using ray tracing techniques.

Prior work can be categorized based on whether they use online or offline algorithms. Online planners compute the best action to take at execution time. They often use sampling techniques and can handle higher dimensional state spaces but require much more computational power during execution. Offline planners, in contrast, compute an approximately optimal policy prior to execution. Finding a representation of the problem that makes the algorithm tractable is still a challenge. The proposed representations lead to an exponentially increasing number of states with the number of other road users considered in the problem and is not easily transferable to other scenarios.

This paper presents a scalable method for approximately solving POMDPs for avoiding multiple occluded users through utility fusion [10]. This decomposition approach consists of computing the optimal state action utility for each user in the environment independently of the others and fusing these utilities to approximate the global problem of avoiding multiple users. This technique has been successfully applied to optimize aircraft collision avoidance systems [11].

The objective of this work is to demonstrate a generic POMDP approach for decision making applied to autonomous driving scenarios with sensor occlusions. By first modeling and solving the single user problem, this paper demonstrates a decomposition method to extend it to multiple users. Two different offline planners are explored: QMDP [12] and SARSOP [13]. Simulation results show that both methods outperform manually designed strategies for the two scenarios presented in Fig. 1. In both scenarios, the ego car decides on the acceleration to apply along a given path.

*This work was supported by the Honda Research Institute.

[1] Maxime Bouton and Mykel J. Kochenderfer are with the Department of Aeronautics and Astronautics, Stanford University, Stanford CA 94305, USA, {boutonm,mykel}@stanford.edu.

[2] Alireza Nakhaei and Kikuo Fujimura are with the Honda Research Institute, 375 Ravendale Dr., Mountain View, CA 94043, USA, anakhaei,kfujimura@hra.com.

The organization of this paper is as follows. Section II describes how to model the single user avoidance problem as a POMDP and how to find an approximately optimal policy. The multiple user problem and the method used to scale the solution are discussed in Section III. Experiments and analysis of the resulting policies are described in Section IV and Section V, respectively.

II. Single User Model

Previous work showed how to model autonomous driving problems as a partially observable Markov decision process (POMDP) [5]–[8]. A POMDP is defined by the tuple $(\mathscr{S}, \mathscr{A}, \mathscr{O}, T, O, R, \gamma)$. The sets $\mathscr{S}$, $\mathscr{A}$, $\mathscr{O}$ are a finite set of states, a finite set of actions, and a finite set of observations, respectively. The transition model T is a conditional probability function $T(s,a,s') = p(s' \mid s,a)$ modeling the likelihood of ending in the state s' at the next time step after taking action a while in state s. At each time step, the agent makes an observation $o \in \mathscr{O}$ that contains partial information about the state due to uncertainty in the sensor measurement or due to occlusions. The observation is modeled by the conditional probability function $O(o,s',a) = p(o \mid s',a)$. Finally, R represents the reward function. At each step, the agent receives a reward $R(s,a) \in \mathbb{R}$ for executing action a in state s. Rewards in the future are discounted exponentially by a factor γ.

In contrast with a Markov decision process (MDP), the state of the environment cannot be sensed directly. The agent maintains a belief that reflects its internal knowledge of the state. The belief state is a probability distribution over all possible states, $b : \mathscr{S} \mapsto [0,1]$, and $b(s)$ represents the probability of being in state s. The belief b is updated after taking action a and observing o using the following equation:

$$b'(s') \propto O(o \mid s',a) \sum_s T(s' \mid a,s) b(s) \tag{1}$$

The solution to a POMDP is an optimal policy π^* that, if followed, maximizes the expected discounted sum of immediate rewards from any given belief. The optimal policy can be extracted from the optimal utility function $U^*(b,a)$.

A. Scenario Modeling

The crosswalk and intersection problems can be formulated as a POMDP.

1) Action space: The ego vehicle must control its acceleration profile along a given path (straight path for the crosswalk scenario or left-turn for the intersection as illustrated in Fig. 1). Strategic maneuvers such as hard braking, moderate braking, maintaining constant speed, and accelerating can be represented by a finite set of acceleration and deceleration actions: $\{-4\,\mathrm{m/s^2}, -2\,\mathrm{m/s^2}, 0\,\mathrm{m/s^2}, 2\,\mathrm{m/s^2}\}$ for the crosswalk and $\{-4\,\mathrm{m/s^2}, -2\,\mathrm{m/s^2}, 0\,\mathrm{m/s^2}, 1.5\,\mathrm{m/s^2}, 3\,\mathrm{m/s^2}\}$ for the intersection.

2) States and transition model: The state summarizes all aspects of the environment that can impact the future. For both scenarios, the state includes the poses of the ego car and the other user to avoid. The pose of the ego car is defined as its position along the given path and its longitudinal velocity: $p_{\text{ego}} = (s_{\text{ego}}, v_{\text{ego}})$. The pose is updated at each time step following a deterministic constant acceleration model.

In the crosswalk scenario, it is assumed that the pedestrian follows a straight path on the crosswalk. Its pose is described by its position along the crosswalk and its velocity. The pedestrian follows a stochastic transition model. At each time step it can change its velocity by a random amount in the set: $\{-1\,\mathrm{m/s}, 0\,\mathrm{m/s}, 1\,\mathrm{m/s}\}$ [8]. Its position is updated using a deterministic point-mass constant velocity model.

Both the position and the velocity of the pedestrian are discretized regularly with a resolution of $1\,\mathrm{m}$ and $1\,\mathrm{m/s}$. The velocity of the pedestrian is bounded up to $2\,\mathrm{m/s}$. An additional pedestrian pose is added to model the case where the pedestrian is absent from the scene: p_{absent}. The discretization results in thirty three possible positions and eight velocities for the ego car and eleven positions and three possible velocities for the pedestrian. By multiplying all possible combinations of ego car positions, velocities, pedestrian positions and velocities, the total number of states amounts to 1.01×10^4

In the intersection scenario, the pose of the other car is described by its position along the lane, its longitudinal velocity, and its lane. The lane models the location of the other car on the road and its intention (going straight or turning). At each time step, the other car can change its acceleration by a random amount in the set $\{-0.5\,\mathrm{m/s}, 0\,\mathrm{m/s}, 0.5\,\mathrm{m/s}\}$. The position and velocity spaces are again discretized but with a larger resolution of $2\,\mathrm{m}$ and $2\,\mathrm{m/s}$ because greater speeds are involved at intersections. Again, an extra state for modeling an absent car is added. By mulitplying all possible combinations of ego car poses, other car poses and lane, the total number of states amounts to 2.15×10^4

3) Observation model: The observation variable represents what the ego car can sense about the state. We can reasonably assume that its own pose is perfectly observable and that it can get noisy measurements of the agent pose in the visible areas. The observation space is similar to the state space except that all states $(p_{\text{ego}}, p_{\text{ped}})$ (or equivalent for the intersection problem) where the other user is occluded are aggregated in the same observation $(p_{\text{ego}}, p_{\text{absent}})$. The observation model can be described as follows:

- A user in a non-occluded area will always be detected
- If a user is in an occluded area it will not be detected
- The measured position and velocity of a detected user are uniformly distributed around the true state at $\pm 1\,\mathrm{m}, \pm 1\,\mathrm{m/s}$ for the crosswalk and $\pm 2\,\mathrm{m}, \pm 2\,\mathrm{m/s}$ for the intersection.

For the intersection scenario, we assume that the lane of the other car is observable. The focus of this work is to analyze the performance of the approach regarding occlusion handling rather than uncertainty in the intentions of other drivers, which has been addressed in previous works [5]–[8].

4) Reward Model: The ego vehicle receives a unit reward for reaching a final position and receives a penalty for collision. The value of the collision penalty can be tuned through simulation as described in Section IV to balance risk averse behaviors and efficiency.

B. Solving for the optimal policy

Once the model has been fully specified, one must compute the optimal belief action utility function $U^*(b,a)$. The optimal action to execute from belief b is then given by $\pi^*(b) = \arg\max_{a \in \mathscr{A}} U^*(b,a)$. In general, computing the exact optimal utility function for a POMDP is intractable [14], and one must rely on approximation techniques instead. Methods for computing the optimal utility function for a POMDP may be either offline or online [15]. Online methods compute the policy at execution time in the environment. They often involve sampling and consider only the states reachable from the current belief. Online methods can handle higher dimensional state spaces, but they are often computationally expensive. Offline methods compute the policy prior to execution in the environment. They cannot handle large state spaces but are computationally inexpensive to execute in the environment. In this paper, we use two offline methods that compute approximately optimal policies: QMDP [12] and SARSOP [13].

The QMDP approach solves the problem under the assumption that the state becomes fully observable after one time step. We can then use the value iteration algorithm to solve for the optimal state-action utility function $U^*(s,a)$ assuming full observability. In order to incorporate state uncertainty when using the policy online, the belief action utility is approximated as follows:

$$U^*(b,a) \approx \sum_{s \in \mathscr{S}} b(s) U^*(s,a) \tag{2}$$

It has been shown that the resulting utility function is an upper bound on the true solution and that this solution tends to have difficulty solving problems where information gathering is important [15].

The second approach is Successive Approximations of the Reachable Space under Optimal Policies (SARSOP). It is a point-based value iteration algorithm that represents the utility function with a finite set Γ of α-vectors associated with actions. This algorithm efficiently explores the belief space to build a collection of α-vectors. The belief-action utility function can then be approximated as follows

$$U^*(b,a) \approx \max_{\alpha \in \Gamma} (\alpha \cdot b) \text{ where } \alpha \in \mathbb{R}^{|\mathscr{S}|},\ b \in [0,1]^{|\mathscr{S}|} \tag{3}$$

Since SARSOP plans in the belief space, it inherently incorporates state uncertainty in the solution.

In the crosswalk scenario, we can visualize the resulting policy for a two dimensional slice of the state space. Fig. 2 fixes two of the state variables, the velocity of the ego car and the velocity of the pedestrian. We can visualize which action the ego car should take if it is at position x along the road and has a belief that the pedestrian is at position y along the road. In Fig. 2 we assume no state uncertainty regardless of occlusions. We can see that the SARSOP policy is more conservative since there is a larger red zone where the ego car has to brake than for the QMDP policy.

When using the policy, the ego vehicle only has a noisy estimate of the pedestrian position and velocity. It is important to identify whether the policy is robust to state uncertainty. Fig. 3 shows the policy when the pedestrian position is a Gaussian centered on the true state. Depending on the tracking performance, the standard deviation of the belief may be larger. As we increase the state uncertainty, we can see the QMDP policy changing shape until the red zone spreads vertically to every pedestrian position. This plot shows regardless of the location of the pedestrian, the vehicle will brake when reaching the position 15 m. As uncertainty increases, the QMDP policy becomes less efficient. Interestingly, the SARSOP policy barely changes until the standard deviation exceeds 1 m, which corresponds to the level of uncertainty in the model used for planning. Finally, when the standard deviation increases to 2 m, the red zone starts spreading vertically but in a less drastic way than for QMDP. We can conclude that using a belief state planner such as SARSOP can result in more robust policies and avoid suboptimal behavior when uncertainty is high.

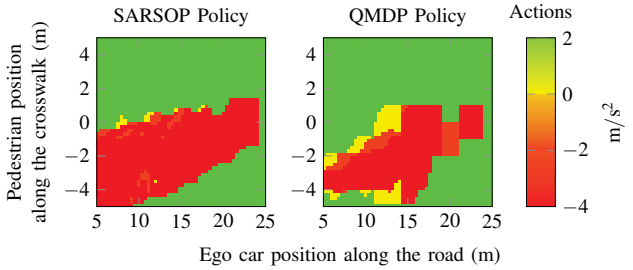

Fig. 2: Representation of the policies obtained using SARSOP (on the left) and QMDP (on the right) assuming no state uncertainty (belief is concentrated on one single state) for a fixed ego velocity at 5.0 m/s and fixed pedestrian speed at 1.0 m/s. The end of the horizontal axis corresponds to the beginning of the crosswalk. The penalty for collision is set to -10.

III. Scaling to Multiple Road Users

Extending the POMDP formulation to avoiding multiple users is straightforward. Adding an additional agent requires introducing two new variables to the state space for the crosswalk scenario and three for the intersection scenarios. As a consequence, the state space would grow from 1.01×10^4 to 2×10^5 states for the crosswalk scenario and from 2.15×10^4 to 7×10^6 states for the intersection scenario. Both problems would become intractable for SARSOP [16]. Utility fusion can help approximate solutions to this problem.

For each user i in the environment, the ego vehicle maintains a belief b_i on its state and computes the optimal belief action utility $U^*(b_i,a)$ assuming i is the only user to avoid. It measures the expected accumulated reward of taking a and then following the optimal policy associated with user i. Utility fusion involves combining the utilities associated with each user to approximate the global optimal utility function $U^*(b,a)$. More formally, it requires defining a function f such that:

$$U^*(b,a) = f(U^*(b_1,a),...,U^*(b_n,a)) \tag{4}$$

where n is the number of users to avoid. The action to execute in the multi-user problem is still given by

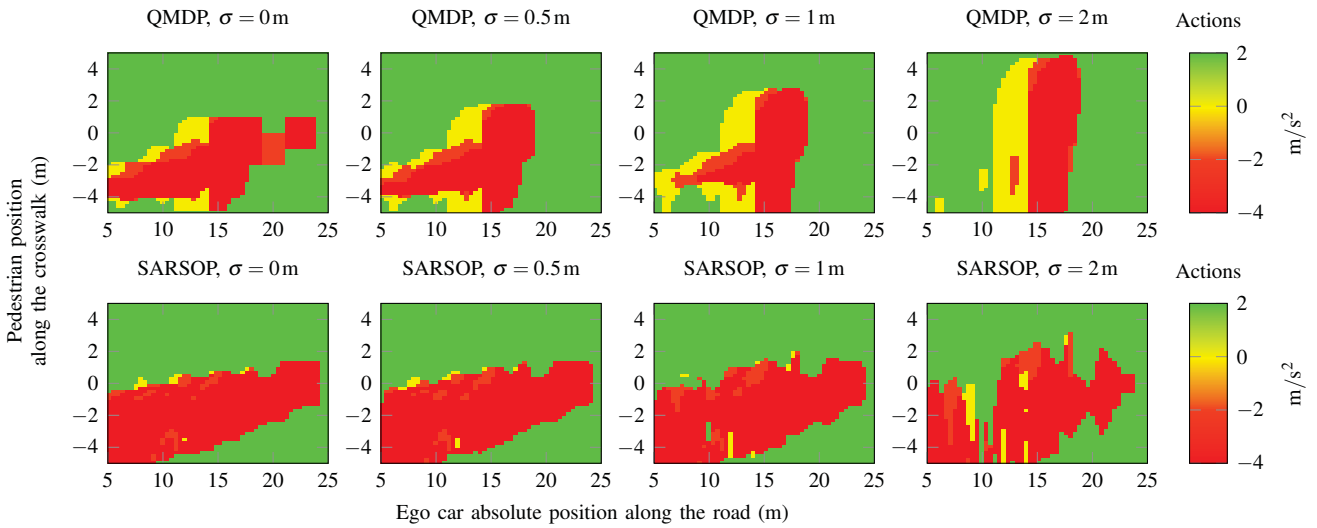

Fig. 3: Representation of the policies obtained using QMDP (top row) and SARSOP (bottom row) with increasing uncertainty for a fixed ego velocity at 5.0 m/s and fixed pedestrian speed at 1.0 m/s. The belief is assumed to be a Gaussian distribution over the position of the pedestrian with standard deviation σ.

$\arg\max_{a\in\mathscr{A}} U^*(b,a)$. Any function f could be used for fusing the utility function. One could train a parameterized nonlinear function in simulation to have suitable performance on a specific problem [11]. In this work, we investigated two fusion methods found in the literature [11]. The first sums the utility functions, and the second one computes the minimum belief action utility over all users:

$$U^*(b,a) = \sum_i U^*(b_i,a) \tag{5}$$

$$U^*(b,a) = \min_i U^*(b_i,a) \tag{6}$$

The first method results from the underlying assumption that all users to avoid are independent. It equally weights the utility of a user in a safe zone of the environment and a user in a more dangerous zone. In the second method, taking the minimum will consider the user with the lowest utility. Given this specific reward function, it considers the user that action a is most likely to harm. This approach is more risk averse as reflected in the results in Section V.

One issue raised by the decomposition method is that it requires knowing the number of users present in the environment. From the modeling in Section II, the ego vehicle can keep a belief on the position and presence of a user in an occluded area. Going further to reduce the complexity of the problem, we will keep one common belief for all invisible users. All the users that are absent or in an occluded area are treated as the same in the decomposition method. The underlying motivation for this approach is that the maneuver to avoid several occluded road users should not be different from avoiding a single occluded user.

Finally, it can be deduced from Eq. (4) that the computational cost of querying the multi-user policy online grows linearly with the number of agents. In order to compute $U^*(b,a)$, one must compute the utilities associated with each detected user and the utility associated with each invisible user. Computing the utility function for a single user can be achieved in the order of milliseconds for QMDP and a tenth of a second for SARSOP. Utility decomposition methods allows to handle an arbitrary number of users to avoid.

IV. Experiments

To evaluate the performance of the policies, we compare them in simulation with manually designed baseline policies. For the crosswalk scenario, the policy involves coming to a full stop at the level of the crosswalk (where there is full visibility), check at each time step if a pedestrian is crossing. Once the crosswalk is clear, the ego vehicle accelerates. A similar strategy is designed for the intersection scenario: stop in the zone of full visibility and then check that the time to collision (TTC) with other cars is below some threshold and cross. For both scenarios, once the car starts crossing it keeps accelerating until it reaches the goal position.

All policies are evaluated according to two metrics:

- safety: average number of collisions
- efficiency: average time to reach the goal position

In the POMDP formulation, we can tune the reward function to balance one objective over another. Increasing the collision cost will favor risk averse behavior, while decreasing it will lead to a more aggressive driving. To have a fair comparison with the baseline, we first evaluate the POMDP policies from QMDP and SARSOP alone with different values for the collision cost in the reward function. Another variable in the simulation is the function used for the decomposition method. Once a suitable reward cost and a suitable fusion function are found, we run Monte Carlo simulations to compare all the strategies. The evaluation framework allows us to measure the performance of the policies in the multiple user formulation with the decomposition method.

To avoid biasing the results, the evaluation models are different from the one used to find the optimal policy and are also higher fidelity. In the simulation, a flow of pedestrians or cars is generated and controlled by a probability of appearance (set to 0.01) at every time step which corresponds to three pedestrians per run in average. The

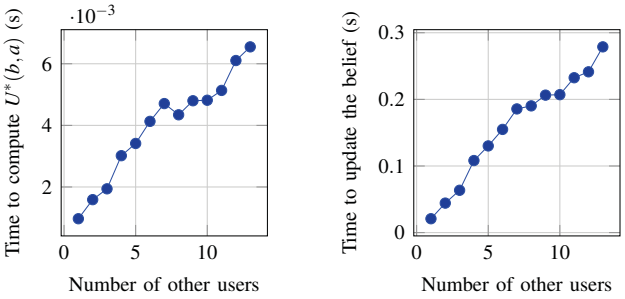

Fig. 4: The computation time required to compute the belief utility function (QMDP) and to update the belief both increase linearly with the number of other road users detected.

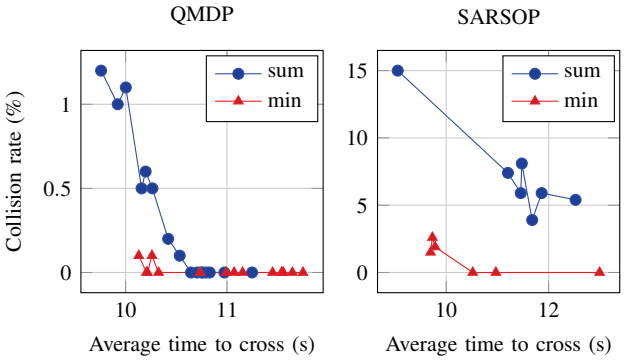

Fig. 5: Evolution of the performance of QMDP (left) and SARSOP (right) as the collision cost is varied for the crosswalk scenario. The two fusion functions from Section III are being analyzed.

pedestrians are following a constant speed of $1\,\mathrm{m/s}$, and the cars are following the intelligent driver model [17]. At each simulation step, the ego car takes a measurement of the scene. Occlusion checking is done using ray-tracing. If the line going from the front of the ego car to a user intersects with an obstacle, then the user is occluded. It is assumed that the vehicle can identify all the non-occluded users and track them independently. The simulated sensor gives a noisy measurement of the position and velocity of other users with Gaussian noise specified in Table I. The ego car can access its own pose exactly.

After each measurement, the ego car can update its belief according to Eq. (1) using the Gaussian observation model. Given the continuous nature of the evaluation environment, the states in the ego car belief will not exactly match with the discretization used in Section II. As a consequence, we use multi-linear interpolation to compute the corresponding belief in the discrete state space.

As shown in Section IV, computing the action scales linearly with the number of users. We can notice that the belief update operation scales linearly as well and is almost fifty times more expensive than computing the QMDP action. For SARSOP, the computation time is of the same order of magnitude than for the belief update. We selected conservative policies that lead to a safe behavior by setting the collision cost to -1.5 for QMDP and -30 for SARSOP to compare against the baseline strategy for the crosswalk scenario, and a similar experiments for the intersection lead to a collision cost of -1.6 for QMDP and -17 for SARSOP. To produce the results in Table II we chose the minimum function for fusing the utilities.

TABLE I: Simulation parameters

Parameter	Value
Position sensor standard deviation	$0.5\,\mathrm{m}$
Velocity sensor standard deviation	$0.5\,\mathrm{m/s}$
Cars maximum speed	$8\,\mathrm{m/s}$
Pedestrians maximum speed	$2\,\mathrm{m/s}$
Simulation time step	$0.1\,\mathrm{s}$
Decision frequency	$0.5\,\mathrm{s}$
Belief update frequency	$0.1\,\mathrm{s}$

V. Results

Before comparing the POMDP policies with the baseline, we analyzed the influence of the collision cost on the performances. Fig. 5 shows the trade-off between safety and efficiency. For both POMDP planners, as we increase the cost of colliding with another user, the policy becomes safer (fewer collisions) but less efficient (longer time to cross). In addition, the simulations were repeated for the two proposed fusion functions. We can see on both graphs that computing the minimum of the utilities over all users leads to safer policies. Moreover in both graphs there is a region of domination of the minimum function over the summation. For SARSOP, all the points are dominated whereas for QMDP there is a region where both functions leads to a safe policy but the minimum function is more conservative.

A similar optimization procedure was followed for the baseline strategies. The two parameters to tune are the threshold on the time to collision and the verification time. For the crosswalk scenario, checking that the time to collision is below $10\,\mathrm{s}$ during 10 decision steps (of $0.5\,\mathrm{s}$) leads to the safest strategy. For the intersection scenario, it is reached for a TTC threshold of $6\,\mathrm{s}$ and a verification time of 8 decision steps. Table II shows the performance of the policies for both scenarios. The results are averaged over one thousand simulations. We also compared them against a random policy.

TABLE II: Performance comparison

	Collision rate (%)	Time to cross (s)
Crosswalk		
Random	54.45 ± 2.56	12.03 ± 10.66
Baseline	0.1 ± 0.04	18.58 ± 5.39
QMDP	0.0 ± 0.0	10.61 ± 3.76
SARSOP	0.0 ± 0.0	10.51 ± 4.44
Intersection		
Random	28.00 ± 2.27	14.68 ± 6.80
Baseline	0.1 ± 0.04	13.46 ± 3.04
QMDP	0.0 ± 0.0	6.20 ± 2.108
SARSOP	0.7 ± 0.83	4.38 ± 0.1342

For the crosswalk scenario, the results show that the two POMDP policies outperform the baseline on the two metrics. The manually designed baselines did not lead to a safe strategy whereas the QMDP policy and the SARSOP policies did not result in any collisions for the crosswalk scenario. Regarding efficiency, they both outperform the baseline as well by more than $8\,\mathrm{s}$.

For the intersection scenario, The SARSOP policy presented more collisions than the baseline and QMDP. The

solver was given a maximum of twelve hours to complete the planning but converged to an aggressive policy. A possible explanation is that such policy would be efficient to avoid one user since the chances of colliding are fairly rare, but it does not adapt to the multiple users scenario. Further reward engineering or parameter tuning could certainly be done to achieve better performances. In contrast, the QMDP policy leads to a less efficient strategy but it is safer than SARSOP. It is still safer and more efficient than the baseline policy.

The few resulting collisions for the baselines are due to users arriving in a critical zone right after the ego car has made the decision to cross. In contrast with the POMDP approach, the ego car is not able to change its decision dynamically. It highlights the difficulty of engineering a good safety criterion such that all future decisions are safe.

In addition to providing a safe and dynamic policy, smart behavior emerged from the POMDP planning. If no road users are present in the environment, the baseline policy always indicates to come to a stop and check the TTC for the given number of steps resulting in a suboptimal behavior whereas both POMDP policies indicate to slow down without necessarily stopping. Another major difference between the two approaches is how occlusions are handled. In the manually designed policy, one must specify the information gathering actions whereas in the POMDP approach they emerge from the planning indifferently for both scenarios. For these reasons, the proposed approach is more efficient but also less scenario specific.

VI. Conclusion

This paper discussed a generic POMDP approach for decision making with sensor occlusions. Although POMDP formulations can explicitly account for uncertainty in the locations of undetected road users, it is challenging to scale them to avoiding multiple road users. We addressed this issue by leveraging the solution to the single user decision making problem through utility fusion. We analyzed the performance of two different offline POMDP planners to solve autonomous driving scenarios involving a crosswalk and an intersection. Experiments confirmed that the computation time evolves linearly with the number of agents detected by the ego vehicle. Simulations showed that the proposed method is safer and more efficient than heuristic policies for the two scenarios of interests.

In the current scenarios, uncertainty arises from occluded parts of the environment due to fixed obstacles. An interesting extension would be to consider moving obstacles of different shapes. By including the obstacle characteristics in the state space, the policy could generalize to a variety of different scenarios.

References

[1] R. Alami, T. Simon, and K. M. Krishna, "On the influence of sensor capacities and environment dynamics onto collision-free motion plans," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[2] J. Miura, Y. Negishi, and Y. Shirai, "Adaptive robot speed control by considering map and motion uncertainty," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 110–117, 2006.

[3] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[4] M. Hauskrecht, "Value-function approximations for partially observable Markov decision processes," *Journal of Artificial Intelligence Research*, vol. 13, pp. 33–94, 2000.

[5] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2014.

[6] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[7] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, "Intention-aware motion planning," in *Algorithmic Foundations of Robotics X*, 2012.

[8] M. Bouton, A. Cosgun, and M. J. Kochenderfer, "Belief state planning for autonomously navigating urban intersections," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[9] S. M. Thornton, F. E. Lewis, V. Zhang, M. Kochenderfer, and J. C. Gerdes, "Value sensitive design for autonomous vehicle motion planning," *IEEE Intelligent Vehicles Symposium (IV)*, 2018, (in review).

[10] S. J. Russell and A. Zimdars, "Q-decomposition for reinforcement learning agents," in *International Conference on Machine Learning (ICML)*, 2003.

[11] J. P. Chryssanthacopoulos and M. J. Kochenderfer, "Decomposition methods for optimized collision avoidance with multiple threats," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 398–405, 2012.

[12] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *International Conference on Machine Learning (ICML)*, 1995.

[13] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Robotics: Science and Systems*, 2008.

[14] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and related stochastic optimization problems," *Artificial Intelligence*, vol. 147, no. 1-2, pp. 5–34, 2003.

[15] M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.

[16] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "DESPOT: online POMDP planning with regularization," *Journal of Artificial Intelligence Research*, vol. 58, pp. 231–266, 2017.

[17] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, p. 1805, 2000.