POMDP Autonomous Vehicle Visibility Reasoning

Kyle Hollins Wray, Bernard Lange, Arec Jamgochian, Stefan J. Witwicki,
Atsuhide Kobashi, Sachin Hagaribommanahalli, and David Ilstrup
Alliance Innovation Lab Silicon Valley
Santa Clara, CA 95051, USA

***Abstract*—We present solutions for autonomous vehicles in limited visibility scenarios, such as traversing T-intersections, as well as detail how these scenarios can be handled simultaneously. The approach models each problem separately as a partially observable Markov decision process (POMDP). We propose an approach for integrating limited visibility within a POMDPs and implementing them on a physical robot. In order to address scalability challenges, we use a framework for multiple online decision-components with interacting actions (MODIA). We present the novel necessary architectural details to deploy MODIA on an actual robot. Moreover, we present a simple solution for multiple AVs to share their knowledge to overcome aspects of these limited visibility scenarios. The entire approach is demonstrated on a fully operational autonomous vehicle prototype acting in the real world at a T-intersection scenario.**

I. INTRODUCTION

Autonomous vehicles (AV) require the ability to reason about scenarios with limited visibility [1, 22, 6, 3]. The National Highway Traffic Safety Association determined that 94% of accidents in the U.S. are caused by human error [8]. The most prevalent cause (44% of cases) is a recognition error, in part a consequence of limited visibility. While high quality sensors (e.g., LIDARs and cameras) can greatly facilitate perception of the observable environment, AVs still cannot necessarily perceive the unobservable environment behind buildings, walls, and other obstructions [22]. They are capable, however, of detecting their own limited visibility [2]. To maximize safety, AVs must reason about these "known unknowns" and intelligently make decisions for when to go, stop, or edge forward slowly for visibility when entering an occluded T-intersection or passing an obstacle in the road.

This essential aspect of AV reasoning has only began to be explored recently. Approaches that introduce hand-crafted parameters [12] or a basic measure of risk [22] into the low-level motion planner's optimization or even the go/no-go mid-level decision-making [6] have been proposed. While these more engineered approaches provide a straight-forward means to slow the AV down, they are finely tuned for very specific scenarios (e.g., 4-way uncontrolled intersections). Therefore, the approach does not necessarily scale to the wide array of scenarios found in real-world driving and does not provide any general framework or method beyond these narrow autonomous driving situations.

The partially observable Markov decision process (POMDP) [9] provides a powerful model for sequential decision-making under limited visibility, sensor noise, and other forms of uncertainty. Specifically it can model known sensor limitations (e.g., limited visibility) through its probabilistic model of observing other vehicles. POMDPs have only recently been embraced as a solution for general AV decision-making [1, 2, 11]. Initial attempts also used POMDPs for limited visibility in impractically large continuous [2] and discrete [5] state spaces at n-way intersections [1, 7]. Recent POMDP algorithms allow scalable AV reasoning for multiple objectives [17, 14], leverage GPUs [15], and employ nonlinear programming techniques for generalized controllers [16, 21]. These algorithms still, however, cannot scale to handle a single large monolothic POMDP. To the best of our knowledge, all prior work was done in simulation, preserving concerns about the feasibility of POMDPs, and critically lacking the details to actualize them on a robot.

Fig. 1. Reasoning under limited visibility about both a T-intersection (left) and passing an obstacle (right) through MODIA implemented on a fully operational autonomous vehicle prototype acting on real public roads.

This paper proposes a novel solution for limited visibility reasoning in AVs using POMDPs. An exemplar limited visibility scenario is considered: T-intersections. We propose a novel architectural implementation of a mathematical framework called MODIA [20, 13]. Our novel *MODIA architecture* fills in the critical details necessary to deploy POMDPs and MODIA on a real robot that were previously lacking. It consists of small well-defined POMDP decision problems (DPs) that are solved offline. When vehicles are perceived online, DPs are instantiated as decision components (DCs). DCs recommend an action at specific arbitration points along the route, with conflicts resolved by an executor arbitration function (e.g., take the safest action). Virtual vehicles, imagined just outside of the field-of-view, are also created and instantiate DCs to allow for reasoning about possible imperceptible vehicles. Multiple collaborative AVs are also able to share information, automatically affecting virtual vehicles and improving POMDP decision-making. As shown in Figure 1, this MODIA architecture is successfully demonstrated on a real AV prototype.

Our contributions are: (1) a general architecture for limited visibility POMDP decision-making in AVs; (2) a novel POMDP solution for T-intersections; (3) a demonstration of POMDPs in MODIA on a real AV.

II. Background

A. The POMDP Model

A POMDP is defined by a tuple $\langle S, A, \Omega, T, O, R\rangle$ [9, 10]. S is a set of n states. A is a set of m actions. Ω is a set of z observations. $T : S \times A \times S \to [0, 1]$ is a transition, with state s and action a, the successor s' follows $T(s, a, s') = Pr(s'|s, a)$. $O : A \times S \times \Omega \to [0, 1]$ is the observation likelihoods, with action a and successor s', observation ω follows $O(a, s', \omega) = Pr(\omega|a, s')$. $R : S \times A \to \mathbb{R}$ is a reward, with state s and action a the reward is $R(s, a)$.

In POMDPs, the agent does not observe the true state of the world, instead maintaining a belief over the true state. Formally, any belief b is a distribution over the state, denoted $b \in B \subseteq \triangle^n$ using the standard n-simplex $\triangle^n$ or any belief subset B. For a belief b, after performing an action a, and making observation ω, the updated belief $b'_{ba\omega}$ follows:

$$Pr(s'|b, a, \omega) = Pr(\omega|b, a)^{\text{-}1} O(a, s', \omega) \sum_{s\in S} T(s, a, s')b(s)$$

with $b'_{ba\omega} = [Pr(s'_1|b, a, \omega), \ldots, Pr(s'_n|b, a, \omega)]^T$ and normalizing constant $Pr(\omega|b, a)^{\text{-}1}$. An initial belief is $b_0 \in B$.

A policy defines which action the agent will take in each belief. Formally, a policy is $\pi : B \to A$. The objective is to maximize expected discounted reward for an infinite horizon. Formally, we seek to find an optimal policy π^* that maximizes the value $V : B \to \mathbb{R}$ at the initial belief $V(b_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R^t | b_0, \pi^*]$ with discount factor $\gamma \in [0, 1)$ and R^t denoting the random variable for the reward at time t. The model enables us to write a Bellman optimality equation for the value of reachable belief b:

$$V(b) = \max_{a\in A} \Big(\sum_{s\in S} b(s)R(s, a) + \gamma \sum_{\omega\in\Omega} Pr(\omega|b, a)V(b'_{ba\omega}) \Big).$$

B. The MODIA Framework

The multiple online decision-components with interacting actions (MODIA) framework [20] enables scalability in real-world POMDP decision-making systems by separately solving each decision-making subproblem, such as negotiating with other vehicles or passing an obstacle. MODIA is defined as the tuple $\langle \mathcal{D}, \{\mathcal{C}^t\}, \epsilon\rangle$. $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ is a set of k decision-making problem (DP) POMDPs, known a priori. Each DP POMDP $\mathcal{D}_i$ has the same action space $A = A_i$. Online, whenever a new DP $\mathcal{D}_i$ scenario is detected at time t, a decision-making component (DC) $\mathcal{C}^t_j$ is instantiated as a copy of the original $\mathcal{D}_i$ including its policy $\pi^t_j = \pi_i$. An active DC maintains a belief b^t_j and recommends actions $\pi_j(b^t_j)$. $\mathcal{C}^t = \{\mathcal{C}^t_1, \ldots, \mathcal{C}^t_\ell\}$ is the set of all active DCs at time t, unknown a priori. $\epsilon : A^* \to A$ is the executor function mapping any action recommendations to a final action performed by the system. At each time step t, all active DCs $\mathcal{C}^t_j \in \mathcal{C}^t$ recommend an action $\pi_j(b^t_j)$. So for recommendations $\overline{a}^t = \langle \pi_1(b^t_1), \ldots, \pi_\ell(b^t_\ell)\rangle$, the executor arbitrates which action should be performed by $\epsilon(\overline{a}^t)$.

The novel architecture to implement MODIA on a robot with multiple POMDPs is presented in the next section.

III. Autonomous Vehicle Architecture

The AV must make second-to-second decisions along a fixed high-level route [18] using static information about the map (e.g., detailed road information) and dynamic perceived sensor information (e.g., vehicles). Stop, edge forward, and go decisions must be made which control the motion along the trajectory [19]. To accomplish scalable online decision-making we employ MODIA.

Three key aspects of this architecture are detailed: (A) MODIA architecture, (B) executor decisions at arbitration points, and (C) virtual vehicles. Once established, we present an example of a DP that reasons under constrained visibility: (D) the T-intersection scenario. Figure 2 provides an architectural overview.

A. Implementing the MODIA Framework on an AV

In order to implement MODIA on a robot, it needs to be linked to a route plan, perception, and trajectory control. The result of high-level route planning is an ordered sequence of arbitration points $P = \langle P_1, \ldots, P_r\rangle$. At each arbitration point $P_k = \langle p^\ell_k, p^a_k\rangle$ has a physical location $p^\ell_k \in \mathbb{R}^2$ and an associated mid-level action $p^a_k \in A$. Here we consider stop, edge, and go actions $A = \{a_s, a_e, a_g\}$. The low-level trajectory control uses these points as an input to its constrained optimization continual planner. Mid-level decision-making controls the points' actions online, continually adjusting them to safely and efficiently reach its destination.

Each category of scenarios the robot must make decisions about, such as traversing a T-intersection or passing an obstacle, is defined as a DP. Each DP $\mathcal{D}_i$ is parameterized by features θ_i of the map, localization, and/or perception's detections. For example, a T-intersection scenario can have $\Theta_i = \langle \theta^\ell_i, \theta^n_i\rangle$ with another vehicle's relative incoming lane $\theta^\ell_i = \{\ell_l, \ell_r\}$ (left/right) and the number of its crossing lanes $\theta^n_i \in \mathbb{N}$. These Θ_i parameterize the transition, observation, and reward functions to allow for minor customizations, such as to model different shapes of T-intersection. These DPs are solved offline and the resulting policy π_i is stored.

Monitors $\mathcal{M} = \langle \mathcal{M}_1, \ldots, \mathcal{M}_k\rangle$ are process nodes $\mathcal{M}_i$ for each type of DP $\mathcal{D}_i$. They convert raw perception features F into abstracted beliefs B_i and arbitration points P relevant for $\mathcal{D}_i$, acting as the function $\mathcal{M}_i : F \to B_i \times P$. For example, position and velocity for AV localization F_{av} and a perceived other vehicle F_{ov} each return vectors in $\mathbb{R}^4$, then $F = F_{av} \times F_{ov}$ and the monitor is $\mathcal{M}_i : \mathbb{R}^4 \times \mathbb{R}^4 \to B_i \times P$. Each monitor $\mathcal{M}_i$ has a clear instantiation and termination condition for DCs as a function of features detected by perception $f^t_j \in F$, localization, and/or the map. This includes the creation of virtual vehicles presented here.

DCs are the online use of a DP $\mathcal{D}_i$'s policy π_i detected by the monitor $\mathcal{M}_i$. Each DP can have many instantiated DCs. For example, a DP for a single T-intersection vehicle can have one DC for each such perceived (or virtual) vehicle. At each time step t, DCs $\mathcal{C}^t_j$ obtain its belief and arbitration point from the monitor $\mathcal{M}_i(f^t_j) = \langle b^t_j, P^t_j\rangle$. $\mathcal{C}_j$ recommends action $\pi_i(b^t_j)$ for arbitration point $P^t_j = \langle p^{t,\ell}_j, p^{t,a}_j\rangle$.

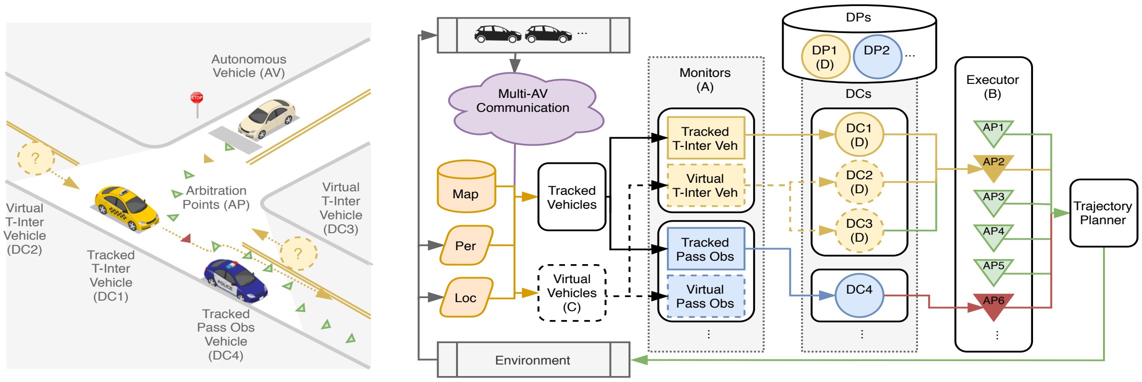

Fig. 2. An architectural overview of MODIA (right) grounded in an illustrative example (left): A concurrent encounter of T-intersection and pass obstacle scenarios with limited visibility left and right. Left: There is one perceived moving vehicle (yellow) and one pass obstacle vehicle (blue). Two virtual vehicles (yellow dotted circles) are created from the limited field of view. Arbitration points (triangles) show their arbitrated actions (green = go, yellow = edge, red = stop). Right: All rounded rectangles are distinct process nodes. Raw data (Map, Per, Loc) is used to create tracked and virtual vehicles; multi-AV communication compressed data is also integrated. Monitors for each DP type create beliefs from this vehicle information. DCs are maintained, instantiated from a DP database, each recommending an action for an arbitration point. The executor arbitrates at these points for trajectory planning.

B. Executor Decisions at Arbitration Points

Given the action recommendations of all DCs at time t, the executor must now arbitrate actions at *all* arbitration points $P_k \in P$ given its action recommendations $\overline{a}_k^t$. Here we consider an executor arbitration function ϵ that selects the most conservative recommendation [20]. That is, stop is preferred to edge which is preferred to go. Formally, for each arbitration point $P_k \in P$, the action recommendations for the point are given by $\overline{a}_k^t = \{p_j^{t,a}, \forall \mathcal{C}_j | p_k^\ell = p_j^{t,\ell}\}$. The executor applies its arbitration function at each point $p_k^a \leftarrow \epsilon(\overline{a}_k^t)$ by:

$$\epsilon(\overline{a}_k^t) = \begin{cases} a_s, & \text{if } a_s \in \overline{a}_k^t \\ a_e, & \text{if } a_s \notin \overline{a}_k^t \wedge a_e \in \overline{a}_k^t \\ a_g, & \text{otherwise} \end{cases} \quad (1)$$

noting that if $\overline{a}_k^t = \emptyset$, the third case implicitly assigns go a_g.

C. Virtual Vehicle Reasoning

POMDPs enable belief-based reasoning about a vehicle's existence even if it has never actually been perceived by perception. Single monolithic POMDPs for AV decision-making [2] define a state space for up to some maximum number of possible vehicles x [3]. Perceived vehicles ($x_p \leq x$) are reasoned about in the same model as imagined potential vehicles outside of view ($x_v \leq x - x_p$) [5].

In MODIA, x_p perceived vehicles are instantiated as separate POMDP DCs. It is this decomposition that enables MODIA's scalability—that is, linear in the number of possible vehicles x. Virtual vehicles are the mechanism in the MODIA architecture that handle any x_v imagined vehicles outside of view. Each x_v virtual vehicle is also a separate POMDP DC.

LIDAR, stereo cameras, and other sensors provide an occupancy grid representing robot-frame knowledge about current sensor imperceptibility. We create virtual vehicles at the edge of the detectable visibility range along all lanes. For example at a T-intersection, two virtual vehicle DCs will always exist, one for each incoming lane.

D. T-Intersection Scenario

The T-intersection POMDP $\langle S_T, A_T, \Omega_T, T_T, O_T, R_T \rangle$ is a DP that models AV-to-vehicle interaction when the AV arrives at the stop line of a T-intersection. Left and right incoming lanes can be obscured, producing virtual vehicles.

State Space: $S_T = S_{av}^\ell \times S_{av}^t \times S_{ov}^\ell \times S^g$ denote the AV's location (before-stop/at-stop/before-gap/at-gap/terminal/goal), the AV's time at the location (short/long), the other vehicle's location (empty/before-stop/at-stop/before-gap/at-gap/goal), and if a gap exists when the AV arrives (yes/no).

Action Space: $A_T = A = \{a_s, a_e, a_g\}$ for both DPs.

Observation Space: $\Omega_T = \Omega_{av}^m \times \Omega_{ov}^m \times \Omega_{ov}^g$ denote if the AV successfully moved (yes/no), if the other vehicle successfully moved (yes/no), and if a gap is detected (yes/no).

Transition Function: $T_T : S_T \times A \times S_T \rightarrow [0, 1]$ multiplies parameterized probabilities of quantifiable events within the state-action space including: (1) the AV and/or other vehicle moving forward, (2) time incrementing, (3) entering a terminal state, (4) the other vehicle slowing down for a crossing AV, (5) the gap's existence toggling based on other vehicle movement, etc.

Observation Function: $O_T : A \times S_T \times \Omega_T \rightarrow [0, 1]$ also multiples parameterized probabilities including: (1) localizing correctly within the AV's location state factor, (2) correctly detecting the other vehicle that does exist, (3) correctly matching the other vehicle to its location state factor, (4) observing the terminal or goal state, (5) detecting the gap correctly based on predictions, etc.

Rewards: $R_T : S_T \times A \rightarrow \mathbb{R}$ are rewards defined by: (1) 0 for the goal state, (2) -1000 for any other terminal state, and (3) -1 for all other state-action pairs.

Monitor: $\mathcal{M}_T$ uses tracked vehicle $f_{ov}^{tv} \in F^{tv}$ and virtual vehicle $f_{ov}^{vv} \in F^{vv}$ features to instantiate and terminate its DCs $\mathcal{C}_j$: (1) instantiation within a constant distance from the intersection (e.g., 50 meters), and (2) termination upon complete traversal of the intersection lane.

T-Intersection

Fig. 3. Experiments demonstrating MODIA on a fully operational autonomous vehicle prototype at a T-intersection. The right three figures show the abstract map (black), stop line (red), path (green), and points along the route (blue) corresponding to each of the three pictures to its left.

IV. Experiments

We evaluate an implementation of the architecture and POMDP solutions on a real autonomous vehicle acting in Santa Clara, California.

MODIA is compared with two established baseline algorithms: Ignorant and Naive [20]. The two baselines represent the two extremes of rule-based systems [4] to serve as a lower and upper bounds for AV behavior. The Ignorant baseline simply ignores virtual vehicles. This represents how quickly it can go through the scenarios while following the law. The Naive baseline always carefully edges forward into the intersection or passed an obstacle, instead of using virtual vehicles. This represents a cautious AV that can slowly make it through the scenario and follow the law. Both baselines use straight-forward rules, lacking the detailed POMDP reasoning about limited visibility in MODIA.

Each setting and algorithm are analyzed by three metrics: its speed profile, visibility profile, and time to complete the scenario(s). Speed profiles measure the distance traveled along route versus speed. They represent the effect the algorithms have on controlling arbitration point actions. Visibility profiles measure the distance traveled along route versus the virtual vehicle's probability of not existing. They represent the amount of visibility reasoning within virtual vehicle DCs.

Figure 4 shows the speed and visibility profiles. Table I states additional information about them. Figure 3 demonstrates MODIA successfully traversing the scenario.

T-Intersection Experiment: The T-Intersection experiment considers the AV arriving at the stop line under an occlusion to the left and right. In Figure 4 (a), the speed profile shows MODIA and the two baselines all stopping at the stop line at around 15 meters. The baselines then produce aggressive and conservative behavior, respectively. The visibility profile shows that they lack any awareness about limited visibility. We observe that MODIA is aware of its limited visibility, as it slowly edges forward into the intersection cautiously. MODIA has instantiated two virtual vehicle DCs, one for each incoming lane. It slows briefly while the POMDPs gain confidence that there are no other real vehicles. Only then does it safely proceed forward. This illustrates POMDP DCs, virtual vehicles, and successfully navigating a T-intersection under limited visibility.

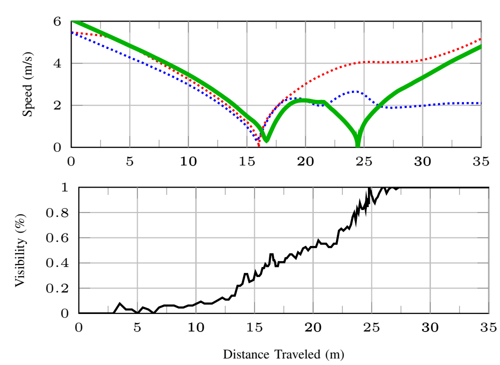

Fig. 4. Results from the experiments on a fully operational autonomous vehicle prototype at a T-intersection. MODIA (green lines) is compared with baselines Ignorant (red lines) and Naive (blue lines). Metrics include: Speed Profiles (top row) and Visibility Profiles (bottom row).

Experiment Setting	MODIA			Baselines	
Name	#V	$\lvert\mathcal{C}\rvert$	T	Ignorant T	Naive T
T-Intersection	2	2	21.7	14.6	22.6

TABLE I
Experiment setting information. Metrics include the number of possible interacting vehicles (#V) and number of instantiated DCs ($\lvert\mathcal{C}\rvert$), and the time (in seconds) to complete the scenario(s) for MODIA and the baselines (T).

V. Conclusion

This paper proposes a novel architecture for implementing POMDPs on an actual autonomous vehicle within the MODIA framework. A general method is described for creating virtual vehicles. A novel POMDP solution for limited visibility reasoning at a T-intersection is presented that uses virtual vehicles. Experiments demonstrate the proposed architecture's success in safely navigating limited visibility scenarios on a real fully operational autonomous vehicle prototype acting in the real world.

References

[1] Maxime Bouton, Alireza Nakhaei, Kikuo Fujimura, and Mykel J. Kochenderfer. Scalable decision making with sensor occlusions for autonomous driving. In *2018 IEEE International Conference on Robotics and Automation*, pages 2076–2081, 2018.

[2] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Probabilistic decision-making under uncertainty

for autonomous driving using continuous POMDPs. In *17th International IEEE Conference on Intelligent Transportation Systems*, pages 392–399, 2014.

[3] Constantin Hubmann, Nils Quetschlich, Jens Schulz, Julian Bernhard, Daniel Althoff, and Christoph Stiller. A POMDP maneuver planner for occlusions in urban scenarios. In *2019 IEEE Intelligent Vehicles Symposium*, pages 2172–2179, 2019.

[4] Kichun Jo, Junsoo Kim, Dongchul Kim, Chulhoon Jang, and Myoungho Sunwoo. Development of autonomous car–Part II: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Transactions on Industrial Electronics*, 62 (8):5119–5132, 2015.

[5] Xiao Lin, Jiucai Zhang, Jin Shang, Yi Wang, Hongkai Yu, and Xiaoli Zhang. Decision making through occluded intersections for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference*, pages 2449–2455, 2019.

[6] Stephen G. McGill, Guy Rosman, Teddy Ort, Alyssa Pierson, Igor Gilitschenski, Brandon Araki, Luke Fletcher, Sertac Karaman, Daniela Rus, and John J. Leonard. Probabilistic risk metrics for navigating occluded intersections. *IEEE Robotics and Automation Letters*, 4(4):4322–4329, October 2019.

[7] Markus Schratter, Maxime Bouton, Mykel J. Kochenderfer, and Daniel Watzenig. Pedestrian collision avoidance system for scenarios with occlusions. In *2019 IEEE Intelligent Vehicles Symposium*, pages 1054–1060, 2019.

[8] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. National Highway Traffic Safety Association, 2015.

[9] Richard D. Smallwood and Edward J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[10] Edward J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations Research*, 26(2):282–304, 1978.

[11] Justin Svegliato, Kyle Hollins Wray, Stefan J. Witwicki, Joydeep Biswas, and Shlomo Zilberstein. Belief space metareasoning for exception recovery. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1224–1229, 2019.

[12] Ömer Şahin Taş and Christoph Stiller. Limited visibility and uncertainty aware motion planning for automated driving. In *2018 IEEE Intelligent Vehicles Symposium*, pages 1171–1178, 2018.

[13] Kyle Hollins Wray. *Abstractions in Reasoning for Long-Term Autonomy*. PhD thesis, University of Massachusetts, Amherst, MA, 2019.

[14] Kyle Hollins Wray and Shlomo Zilberstein. Multi-objective POMDPs with lexicographic reward preferences. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1719–1725, 2015.

[15] Kyle Hollins Wray and Shlomo Zilberstein. A parallel point-based POMDP algorithm leveraging GPUs. In *Proceedings of the 2015 AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents*, pages 95–96, 2015.

[16] Kyle Hollins Wray and Shlomo Zilberstein. Generalized controllers in POMDP decision-making. In *2019 IEEE International Conference on Robotics and Automation*, pages 7166–7172, 2019.

[17] Kyle Hollins Wray, Shlomo Zilberstein, and Abdel-Illah Mouaddib. Multi-objective MDPs with conditional lexicographic reward preferences. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 3418–3424, 2015.

[18] Kyle Hollins Wray, Luis Pineda, and Shlomo Zilberstein. Hierarchical approach to transfer of control in semi-autonomous systems. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 517–523, 2016.

[19] Kyle Hollins Wray, Dirk Ruiken, Roderic A. Grupen, and Shlomo Zilberstein. Log-space harmonic function path planning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1511–1516, 2016.

[20] Kyle Hollins Wray, Stefan J. Witwicki, and Shlomo Zilberstein. Online decision-making for scalable autonomous systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4768–4774, 2017.

[21] Kyle Hollins Wray, Akshat Kumar, and Shlomo Zilberstein. Integrated cooperation and competition in multi-agent decision-making. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 4751–4758, 2018.

[22] Ming-Yuan Yu, Ram Vasudevan, and Matthew Johnson-Roberson. Occlusion-aware risk assessment for autonomous driving in urban environments. *IEEE Robotics and Automation Letters*, 4(2):2235–2241, April 2019.