Accepted for spotlight presentation at the Workshop on Machine Learning for Motion Planning ICRA 2021

Motion Planning via Bayesian Learning in the Dark

Carlos Quintero-Peña*, Constantinos Chamzas*, Vaibhav Unhelkar and Lydia E. Kavraki

***Abstract*— Motion planning is a core problem in many applications spanning from robotic manipulation to autonomous driving. Given its importance, several schools of methods have been proposed to address the motion planning problem. However, most existing solutions require complete knowledge of the robot's environment; an assumption that might not be valid in many real-world applications due to occlusions and inherent limitations of robots' sensors. Indeed, relatively little emphasis has been placed on developing safe motion planning algorithms that work in partially unknown environments. In this work, we investigate how a human who can observe the robot's workspace can enable motion planning for a robot with incomplete knowledge of its workspace. We propose a framework that combines machine learning and motion planning to address the challenges of planning motions for high-dimensional robots that learn from human interaction. Our preliminary results indicate that the proposed framework can successfully guide a robot in a partially unknown environment quickly discovering feasible paths.**

I. Introduction

Current motion planning methods provide efficient and diverse ways to compute the motion of a robot given perfect information about the environment [1]. However, the problem remains challenging when only noisy or partial information is available. If robots are to be deployed in houses, hospitals, nursing homes, and other unstructured environments, planning algorithms need to reason with partial information while being computationally tractable. In such environments, a service robot will typically rely on onboard sensors to create a representation of its world, which may lead to occlusions, limited point of view, or limited sensor precision. At the same time, most of these tasks are safety-critical since the robot might need to work alongside or with humans. Consequently, new methods that produce safe trajectories in the presence of sensing uncertainty need to be developed.

As an example, consider the Fetch robot shown in Fig. 1 performing a common manipulation task. It is tasked with picking an object that is located inside a box (shown in light blue). Due to limited visibility, the robot can only see the frontal face of the box, represented in the figure as colored voxels with occupancy information using Octomaps [2]. When planning from a starting configuration outside the box to the goal configuration inside, a motion planning algorithm may return the trajectory shown on the right. The resulting trajectory is in collision with the box but this information is not available to the robot at planning time.

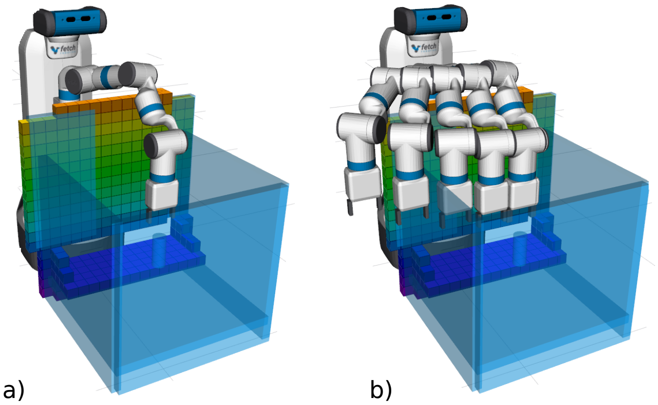


Fig. 1. **a)** An example where the robot only has access to partial workspace information (octomap) while the full geometric information lightly-blues shaded box is not visible. **b)** A standard motion planner produces an unsafe (in-collision) path when planning in the partially known environment.

In this paper, we present an exploratory work for motion planning that incorporates human expertise into the planning process to produce safe trajectories when only a partial representation of the environment is available. Our method, **B**ayesian **L**earning **IN** the **D**ark (BLIND), combines inverse reinforcement learning with motion planning in a novel way to solve motion planning problems for high degree-of-freedom (DOF) robots.

BLIND works by constructing a model of the task from past experiences that captures safety as perceived by the human. This model can be used to guide a motion planner to produce safe trajectories. The parameters of the model are learned by actively interacting with the human and inverse reinforcement learning. This allows BLIND to produce safe trajectories despite the missing workspace information.

The next section describes previous relevant work that address the problem of motion planning under sensing uncertainty and others that consider human-robot interaction. Sec. III describes the details of our proposed methodology. Sec. IV describes experiments performed in simulated environments showing a Fetch robot solving the task described in Fig. 1 and an ablation study. Finally, we describe conclusions and future work.

II. Related work

We consider the problem of motion planning under sensing uncertainty. Many motion planning methods model the problem as a Markov Decision Process (MDP) [3]. More specifically, partially observable MDPs (POMDPs) provide a principled way to model sensing uncertainties in the

*Authors contributed equally, and are ordered randomly

CC is supported by NSF-GRFP 1842494. This work has also been supported in part by NSF 1718478, NSF RI 2008720 and Rice University Funds.

All authors are affiliated with the Department of Computer Science, Rice University, Houston TX, USA {chamzas, carlosq, vaibhav.unhelkar, kavraki}@rice.edu

planning and execution stages [4]. Despite recent and ongoing advances in POMDP solvers [5], which have made POMDPs exceedingly tractable, the applicability of POMDPs remains largely limited to discrete state and action spaces. This requirement of discretization makes the direct application of POMDPs to motion planning tasks for high-DOF robots challenging. In this work, instead of discretizing the high-dimensional state-space, we construct a coarser MDP over a lower-dimensional space that can be used to guide a standard motion planner suitable for high-dimensional robots.

Linear-Quadratic Gaussian motion planning (LQG-MP) provides true state distributions for uncertain states and actions that can be used to perform robust planning [6]. However, it requires a Gaussian observation model, which may not hold when obstacle locations and shapes are partially unknown at planning time. Other approaches to planning under uncertainty assume that noisy versions of the obstacles are known and use them to find trajectories with a low probability of collision [7]–[9]. The aforementioned approaches are not suitable when planning using only occupancy information, since they require knowledge of the shape of the obstacles and estimates of the sensing uncertainty or they require Gaussian observation models.

Closer to our work are methods that perform planning in partially-known environments where obstacles are sensed through contact information [10], [11]. In these cases, planning and execution are interleaved to find valid paths based on contact feedback when the robot attempts to move on low probability-of-collision paths according to a belief. Similar to the work presented here, the sensing uncertainty comes from limitations in sensing, occlusions, or limited field of view from the robot's perspective, and the environment is typically represented as occupancy information. However, a critical assumption in these methods is that the robot is allowed to make contact with the environment. For safety-critical applications, this assumption may not hold since making contact may come with a high penalty (e.g., breaking a glass or hurting a human). Our method incorporates a human into the planning process avoiding the need to execute potentially unsafe trajectories.

The idea of utilizing human guidance to teach robots new skills and behaviors has also received significant attention in the last decade, with learning from demonstration (LfD) being a popular paradigm [12] augmented with active learning from human queries [13], [14]. However, these methods usually apply to discrete state-spaces [14] or plan only for the low-dimensional robot end-effector [13], [15]. In this work, we build upon these ideas but additionally seek to scale the challenges of high-dimensional state spaces, by planning directly in the high-DOF configuration space and incorporating human input as soft constraints. Finally, this work focuses on producing safe plans in incomplete environments while prior (LfD) methods largely focus on learning manipulation skills.

III. Proposed Method

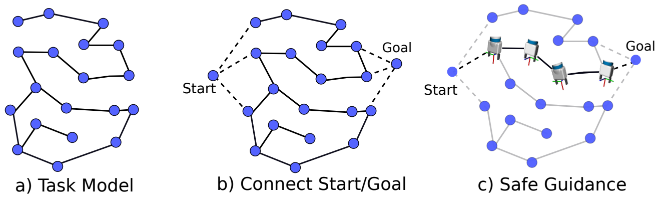
We consider a high-DOF robot with d controllable joints acting in a potentially unknown or partially known environment $\mathcal{W}$. The human and the robot are collocated and the human can observe the full environment. Our goal is to create a model for the task at hand alongside the feedback given by the human to produce safe trajectories, despite the incomplete workspace information. To accomplish this, we introduce BLIND, a methodology that leverages Bayesian inverse reinforcement learning to capture the human notion of safety and optimization-based motion planning.

For a given motion planning problem, BLIND constructs and maintains a task model structure as a roadmap $\mathcal{G}$. The task model represents past robot experiences from similar tasks and it can either be constructed on the fly or queried from an existing database. In $\mathcal{G}$, vertices correspond to workspace projections of discretized trajectories from past experiences. In this paper, we use poses of the robot's end-effector as workspace projections. These vertices can be computed by discretizing trajectories previously used by the robot to solve similar tasks and keeping the pose of the end-effector for every configuration. However, depending on the task, other workspace projections may be used. Each vertex is also related to a temporal parameter that represents its relative temporal location along the original trajectory. An edge between two vertices in $\mathcal{G}$ implies a temporal dependency between the corresponding end-effector poses; that is, projections that appeared earlier have out-edges to later projections in the trajectories.

Alg. 1 shows the steps followed by BLIND. The algorithm assumes that roadmap $\mathcal{G}$ already exists. If this is not the case, it can be created by using a motion planner capable of providing different trajectories using $\mathcal{W}$, start and goal, e.g., a sampling-based motion planner [16] or an optimization-based motion planner with randomized initial trajectories [17]. Furthermore, the algorithm requires the available workspace information $\mathcal{W}$, the start, the goal, and a maximum number of human queries (max_q).

Given a new motion planning problem, BLIND starts by connecting the start and goal to the task model roadmap $\mathcal{G}$ (line 1). Then, BLIND queries the task model for a guidance P (line 3). The guidance corresponds to a sequence of end-effector poses along with their corresponding temporal dependencies from start to goal. This step is achieved by performing a graph search over the task model to find a path between start and goal (see Sec. III-A). This guidance is then used by a guided motion planning algorithm (line 4) to find a collision-free trajectory from start to goal with end-effector constraints given by P (see Sec. III-A).

The trajectory $\mathcal{T}$ returned by the planner is presented to the human as a candidate solution that she can accept or reject. In case she does not accept, BLIND asks the human for a detailed critique (line 8). That is, the human is asked to label each segment in $\mathcal{T}$ as good (D^+) or bad (D^-) (see Sec. III-B) according to her preferences. A segment is the volume swept by the robot between two consecutive configurations of



a) Task Model b) Connect Start/Goal c) Safe Guidance

Fig. 2. (Left) Task model $\mathcal{G}$ constructed from previous similar experiences. Each vertex corresponds to a robot end-effector-pose, edges encode temporal dependencies from the past trajectories. (Center) Given a new motion planning problem, the start and goal configurations are connected to the roadmap and a shortest path search is performed. (Right) The result is a sequence of end-effector poses (guidance) that can help to guide a motion planner to find safe trajectories.

the discretized trajectory. The critique is used by an inverse reinforcement learning algorithm to compute the posterior of a reward function that best explains the labeled data (line 9). By sampling from this posterior, BLIND updates the edge costs of the roadmap to capture the user preference (line 10).

Algorithm 1: BLIND

```
    input  : Roadmap $\mathcal{G}$, incomplete workspace $\mathcal{W}$,
             start, goal, $max_q$
    output : Collision-free Trajectory $\mathcal{T}$ or $\emptyset$ when fail
 1  $\mathcal{G}$' ← ConnectToRoadmap ($\mathcal{G}$, start, goal) ;
 2  for $i$ = 1,...,$max_q$ do
 3      $P$ ← GuidanceSearch ($\mathcal{G}$') ;
 4      $\mathcal{T}$ ← GuidedMP ($\mathcal{W}$, $P$, start, goal) ;
 5      if HumanAccepts ($\mathcal{T}$) then
 6          Return $\mathcal{T}$;
 7      else
 8          $(D^+, D^-)$ ← Critique ($\mathcal{T}$) ;
 9          $\Pr(R|D^+, D^-)$ ← BIRL ($\mathcal{G}$', $(D^+, D^-)$) ;
10          $\mathcal{G}$' ← UpdateRoadmap ($\Pr(R|D^+, D^-)$) ;
11  Return $\emptyset$ ;
```

BLIND contains two key novel components: on one hand it maintains a task model that captures the human preference and that is used to guide a motion-planner to follow safe trajectories. On the other hand, it models safety as a reward over the task model that can be learned using inverse reinforcement learning from a human. The following sections describe these methods in detail.

A. *Using the task model to guide motion planning*

In BLIND, the task model $\mathcal{G}$ captures past similar experiences *and* the human preference for the task. Past experiences are captured on the roadmap vertices by end-effector poses from trajectories previously planned in similar tasks. The human preference is captured by the cost of edges between vertices in $\mathcal{G}$ (see Sec. III-B). From an MDP point of view, the cost of each edge is equivalent to the negative reward of performing the action represented by the edge. These rewards (costs) can be learned through inverse reinforcement learning to capture the human preference. Therefore, finding the shortest path from start to goal results in a sequence of vertices that represent past experiences and has the maximum accumulated reward (we call this sequence guidance). Fig. 2 shows a schematic of this process.

A key insight in BLIND is that guidance can be used by a motion planner to produce a trajectory from start to goal that is collision-free and passes through the set of end-effector poses given by the guidance. Below, we describe how an optimization-based planner can be used to perform such a task.

Optimization-based motion planners [18]–[20] optimize a cost function over the trajectory while ensuring collision-free motions. In TrajOpt [20], for instance, the collision avoidance is achieved by keeping a positive signed distance between robot links and obstacles in the workspace. Other behaviors such as joint limits, dynamics, and end-effector constraints can be incorporated as additional terms in the optimization formulation. This gives rise to a non-convex optimization problem that can be solved using sequential convex optimization, where each non-convex term is linearized around a nominal trajectory and a locally convex version of the problem is solved at every iteration [20], [21].

This can be achieved using TrajOpt with additional pose constraints from the guidance (see [20]). Formulation (1) shows "Guided TrajOpt":

$$
\begin{aligned}
&\underset{x_0,\dots,x_T}{\text{minimize}} && \sum_{t=0}^{T-1} \|x_{t+1}-x_t\|^2 && \text{(1a)}\\
&\text{subject to} && x_0 = x_{St}, && \text{(1b)}\\
& && x_T = x_G, && \text{(1c)}\\
& && \text{sd}(A_{it}, O_j) \geq d_s \ \forall i, j, t && \text{(1d)}\\
& && F_k^{-1}\text{FK}(x_\tau) = 0 \ \forall (k, \tau) \in P && \text{(1e)}
\end{aligned}
$$

where the variables $x_t \in \mathcal{C} \subseteq \mathbb{R}^d$ are waypoints of a discretized trajectory $(t = 0, \dots, T)$ in configuration space; x_{St}, x_G are given start and goal configurations respectively, **sd**( ) is the signed distance between convex shapes, A_{it} is the i-th robot link at timestep t, O_j is j-th obstacle and d_s is a safe distance. In Equation 1e, F_k denotes the k-th target pose from the guidance set that needs to be enforced at its corresponding timestep for the robot's end-effector and FK(x_τ) is the pose of the end-effector at configuration x_τ. τ is the timestep of the corresponding pose. In Equation 1e, F_k denotes the k-th target pose from the guidance set that needs to be enforced at its corresponding timestep for the robot's end-effector and FK$_k(x_t)$ is the pose of the end-effector the robot at configuration x_t.

B. *Learning safety from human interaction*

The roadmap $\mathcal{G}$ provides a model for the task where costs between vertices encode how "good" it is to go from one end-effector pose to the next. Therefore, the task model can be seen as an MDP where vertices correspond to states, edges correspond to actions, costs of edges are negative rewards, and state-action values are computed as the cost-to-goal using the Bellman-Ford algorithm. The notion of safety is captured by a reward function (R) over the edges that can be learned from interaction with the human using inverse reinforcement learning. The reward function is calculated as a linear combination of features [22] over the vertices (v_1, v_2) of each edge.

$$R(v_1, v_2) = w \cdot \phi(v_1, v_2)$$

Where w is a vector of weights and ϕ is a feature function. Inspired by [14], we maintain a belief over reward functions given a set of critiques from the user and update them to capture safe trajectories using Bayesian Inverse Reinforcement Learning (BIRL) [23]. Given user critiques in the form of good segments and bad segments, the probability of edges belonging to the set of good/bad labels can be written as follows:

$$\Pr(a_i \in E(s_i) \mid R) = \frac{1}{Z_i} \exp^{\alpha Q(s_i, a_i, R)}$$
$$\Pr(a_i \notin E(s_i) \mid R) = 1 - \frac{1}{Z_i} \exp^{\alpha Q(s_i, a_i, R)}$$

where s_i, a_i are the i-th state and action (pose, edge) of the candidate trajectory respectively, $Q(s_i, a_i, R)$ is the state-action value and $E(s_i)$ corresponds to the set of optimal actions at each state s_i, i.e., $E(s) = \arg\max_a Q(s, a)$.

The likelihood of the labeled data can be expressed as:

$$\Pr(D^+, D^- \mid R) = \prod_{(s_i, a_i) \in D^+} \Pr(a_i \in E(s_i) \mid R) \prod_{(s_i, a_i) \in D^-} \Pr(a_i \notin E(s_i) \mid R)$$

To generate samples from the posterior using the labeled candidate trajectory, we use the Monte Carlo Markov Chain (MCMC) policy walk algorithm from [23]. Sampling from the MCMC allows us to estimate the posterior distribution of reward functions over the edges of $\mathcal{G}$.

IV. Experiments

In our experiments, we compared the performance of BLIND with the following three baselines:

- Plain-TrajOpt (P-TRAJOPT): This baseline is the plain TrajOtp algorithm without any modifications.
- Random-BLIND (R-BLIND): This variation of BLIND utilizes the task model graph $\mathcal{G}$ but not the human critiques. Instead, the costs of edges in $\mathcal{G}$ are set randomly for every new attempt.
- Penalized-BLIND (P-BLIND): This variation of BLIND instead of BIRL uses a simple heuristic to update the costs of edges in $\mathcal{G}$. The heuristic simply adds a large cost for every segment that was labeled as unsafe by the human.

We designed a realistic scenario with an incomplete environment to examine the performance of BLIND. We emulated the performance of a human by utilizing a collision checker that has access to the full geometric information of the environment. The robot starts from the left side of the box and tries to place its arm inside the box while avoiding collision with the box obstacle. As shown in Fig. 1a) only part of the box is detected from the robot sensors (3D-camera) and is represented as an occupancy grid. The rest of the box is not detected by the robot and it must learn to avoid it leveraging human queries.

We generated motion planning problems in incomplete environments similar to [24]. Between different problems, we vary the position of the box by $\pm 10cm$ along the X and Y-axis, and its angle relative to the robot base by $\pm 15^o$. Both the start and the goal are found through IK-sampling that place the end-effector on the left-side and inside of the box respectively with a tolerance of $\pm 10cm$.

To create the task model graph $\mathcal{G}$ we generated 10 environments with the aforementioned procedure and used P-TRAJOPT only with the incomplete sensed information to generate 10 trajectories. The end-effector poses from these 10 trajectories were composed in a task-model roadmap key-pose as shown in Fig. 3 a). The same task-model roadmap was used for all BLIND variants in all cases.

For this task we defined 5 simple reward features for each edge on the task-model roadmap.

$$\begin{aligned}
\phi(v_a, v_b) =& [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5] \\
\phi_1(v_a, v_b) =& |x_a + x_b| \\
\phi_2(v_a, v_b) =& |y_a + y_b| \\
\phi_3(v_a, v_b) =& |z_a + z_b| \\
\phi_4(v_a, v_b) =& \mathbb{1}_{(v_a, v_b) \in M} \\
\phi_5(v_a, v_b) =& \mathbb{1}_{(v_a, v_b) \notin M}
\end{aligned}$$

where x_i, y_i, z_i are the spatial coordinates of end-effector pose i and M is the set of known regions from the sensors. More specifically ϕ_1, ϕ_2, ϕ_3 encode spatial information of the edges while ϕ_4, ϕ_5 encode whether the edge vertices lie in known or unknown regions.

To evaluate the methods, we created a test set of 100 problems with the procedure described above. Given a new problem, all methods propose a trajectory and query the human for approval. If the human approves the trajectory, it is executed. If the human does not approve the trajectory, the human labels the invalid trajectory parts, and a new trajectory is proposed until approval. P-TRAJOPT and R-BLIND disregard the critique while P-BLIND and BLIND utilize it. P-TRAJOPT is initialized with a different random trajectory each time to avoid proposing the same trajectory.

To evaluate the performance of the methods we count the number of queries per problem and the success rate. As one query we refer to the human Accepting/Rejecting and optionally providing a Critique. In other words, the number of queries is equivalent to how many times line 5 is called. Note that at least 1 human query is needed to verify the safety of the path. Motivated by the cost of querying a human, the problem instance is considered a failure if it requires more than ten queries.

The average number of queries needed for the 100 test

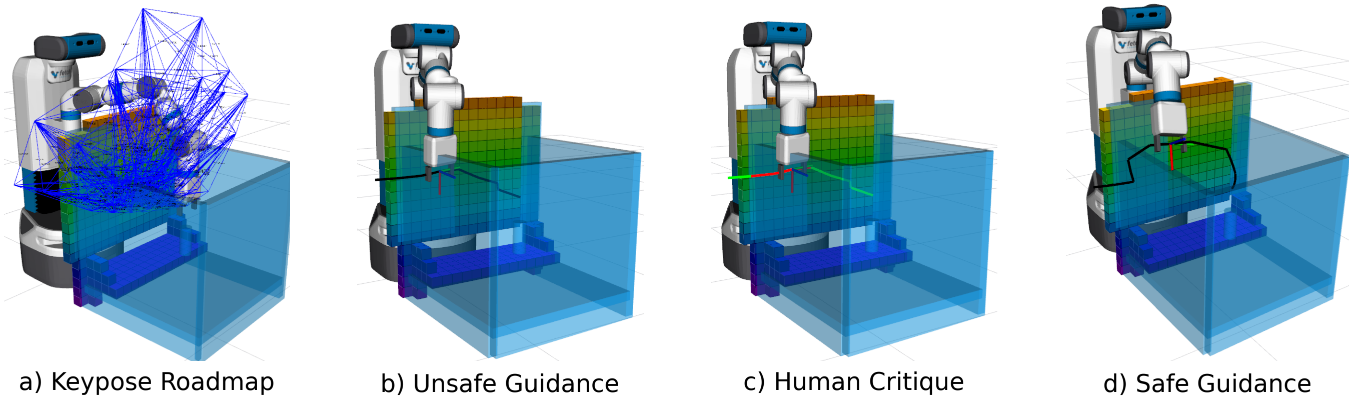


Fig. 3. **a)** The task-model roadmap that was constructed from 10 example trajectories. **b)** A misleading guiding path (end-effector poses sequence) which fails to avoid the obstacles. The line traces the end-effector position in space. **c)** A critique provided by a human. Green lines denote good segments while red lines denote a bad segment. Note that the human annotates the joint-space trajectory which corresponds to the end-effector segments visualized. **d)** A retrieved guidance (end-effector poses) that correctly guides around the box obstacle

TABLE I
AVG HUMAN QUERIES AND SUCCESS RATE

Method	Human Queries Mean (±std)	Success Rate
P-TRAJOPT	7.88 ± 3.78	0.24
R-BLIND	6.33 ± 3.05	0.67
P-BLIND	3.63 ± 3.19	0.87
BLIND	2.96 ± 3.17	0.86

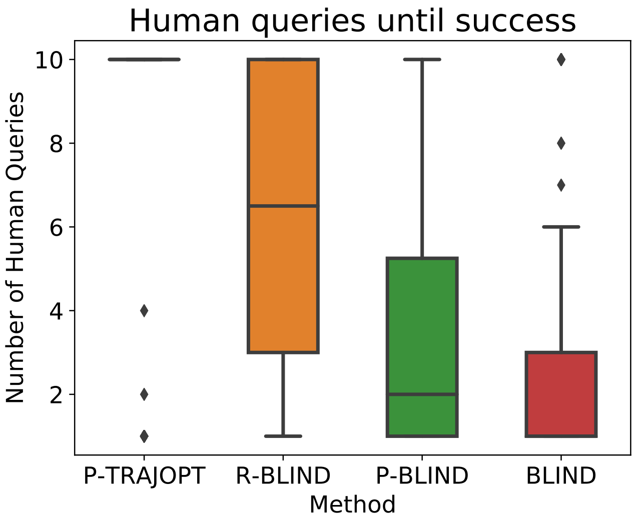


Fig. 4. Number of human queries needed until a safe path was found. A maximum of 10 human queries was allowed.

problems along with the success rate is shown in Table I, while Fig. 4 has the box plots of the number of queries needed for each method. BLIND outperformed all the baselines both in terms of the number of queries and success rate. As expected due to most of the environment being unknown to the robot, P-TRAJOPT was unable to find a safe path in most cases. BLIND outperformed the other variants demonstrating that the learned reward successfully produces safe paths.

V. DISCUSSION

In this work, we proposed BLIND, a method that can learn to produce safe trajectories with human guidance in incomplete environments. Our preliminary results show that the method can successfully incorporate feedback given by the human to produce safe trajectories with only a few interactions and large missing parts of the environment. Furthermore, our method to learn and update the human notion of safety proved to be better than using random edge costs of the task model or a simple penalization strategy. In future work, we would like to improve BLIND by applying it to more varied environments and learning more general features. We would also like to implement a user interface and evaluate BLIND with human users in a realistic partial visibility setting for the robot.

REFERENCES

[1] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation.* MIT Press, 2005.

[2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[3] J. Burlet, O. Aycard, and T. Fraichard, "Robust motion planning using markov decision processes and quadtree decomposition," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, pp. 2820–2825 Vol.3, 2004.

[4] J. van den Berg, S. Patil, and R. A. and, "Motion planning under uncertainty using iterative local optimization in belief space," 2012.

[5] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "Despot: Online pomdp planning with regularization," in *Advances in Neural Information Processing Systems* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.

[6] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized Path Planning for Robots with Motion Uncertainty and Imperfect State Information," *Int. J. Rob. Res.*, vol. 30, p. 895–913, June 2011.

[7] B. Luders, M. Kothari, and J. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *AIAA guidance, navigation, and control conference*, p. 8160, 2010.

[8] B. Axelrod, L. P. Kaelbling, and T. Lozano-Pérez, "Provably safe robot navigation with obstacle uncertainty," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1760–1774, 2018.

[9] C. Quintero-Peña, A. Kyrillidis, and L. E. Kavraki, "Robust optimization-based motion planning for high-DOF robots under sensing uncertainty," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[10] B. Saund and D. Berenson, "Motion planning for manipulators in unknown environments with contact sensing uncertainty," in *International Symposium on Robotics Research (ISRR)*, November 2018.

[11] B. Saund, S. Choudhury, S. Srinivasa, and D. Berenson, "The blindfolded robot: A bayesian approach to planning with contact feedback," in *International Symposium on Robotics Research (ISRR)*, October 2019.

[12] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[13] A. Singh, L. Yang, C. Finn, and S. Levine, "End-To-End Robotic Reinforcement Learning without Reward Engineering," in *Robotics: Science and Syst.*, 2019.

[14] Y. Cui and S. Niekum, "Active reward learning from critiques," in *IEEE Int. Conf. Robot. Autom.*, pp. 6907–6914, IEEE, 2018.

[15] M. Palan, G. Shevchuk, N. Charles Landolfi, and D. Sadigh, "Learning Reward Functions by Integrating Human Demonstrations and Preferences," in *Robotics: Science and Syst.*, 2019.

[16] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[17] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[18] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[19] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4569–4574, 2011.

[20] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[21] R. Bonalli, A. Cauligi, A. Bylard, , and M. Pavone, "GuSTO: guaranteed sequential trajectory optimization via sequential convex programming," in *2019 IEEE Conf. on Robotics and Automation*, 2019.

[22] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, p. 1, 2004.

[23] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, (San Francisco, CA, USA), p. 2586–2591, Morgan Kaufmann Publishers Inc., 2007.

[24] C. Chamzas, Z. Kingston, C. Quintero-Pena, A. Shrivastava, and L. E. Kavraki, "Learning Sampling Distributions Using Local 3D Workspace Decompositions for Motion Planning in High Dimensions ," *IEEE Int. Conf. Robot. Autom.*, 2021.