Robot path planning using imprecise and sporadic advisory information from humans

G. A. Di Caro and E. Feo-Flushing

Carnegie Mellon University in Qatar, Doha
{gdicaro,efeoflus}@andrew.cmu.edu

Abstract. In environments featuring hazards (e.g., debris, holes in the ground), robot navigation can be challenging. Robot's sensors alone might be not able to guarantee timely detection of the threats. In such situations, the presence of nearby humans can be exploited to support safe robot navigation. The human can proactively provide advisory information and issue warnings. Unfortunately, verbally expressed human's inputs are usually quite imprecise or ambiguous when referring to spatial elements. We consider how to model the inherently imprecise and sporadic "human sensor" by using the formalism of imprecise probabilities, and how to use the model to build maps fusing robot sensor data and human inputs. Map information is used for path planning, searching for paths that balance survivability and efficiency (e.g., time). In a number of simulation scenarios we study the effectiveness of our approach compared to standard ways to build the map and perform path planning.

Keywords: Survivable path planning · Imprecise probabilities · HRI.

1 Introduction

For mobile ground robots, the navigation in *extremely cluttered, unknown, hazardous environments* can always be a challenge. In spite of recent advances in vision and range finding devices, the detection of elements such holes, stairs, puddles, fire, transparent windows and doors, electrified wires, can still be problematic. The problem is exacerbated in scenarios such as *post-disaster* ones, where the robot has to navigate in partially collapsed or semi-destroyed buildings areas, and environmental conditions are challenging.

An option to mitigate the risk of issues during navigation consists in *leveraging the presence of nearby human agents.* In fact, humans can exploit their cognitive and sensory abilities to easily spot threats, such as a puddle, that for the robot it might be difficult to detect timely and effectively. Here we are envisaging scenarios where autonomous robots and humans will team-up performing *collaborative missions.* Applications such as search and rescue and emergency-response, seem to be first candidates for featuring such mixed teams. In these scenarios it can be assumed that human agents will often be in the *proximity of robots* during mission deployment. Therefore, he/she can *proactively* "help" by providing information useful for path planning and navigation. At the same time, in these life-critical scenarios, humans will mostly be busy with their tasks, such as it is reasonable to state that inputs can indeed only be *sporadic*.

To these facts, we need to add up that human inputs are expected to be expressed in a *conversational language*, which is *inherently imprecise* [11]. E.g., if the robot is moving towards a target position and a puddle is in between, a nearby human may issue a warning by saying something like: "Hey, watch out, there's a puddle ahead of you!". What precisely that "ahead" means in terms of range and bearing is unspecified, as well as how large or deep the puddle is. In general, we can't really expect information being delivered in a precise form.

In this work, we start from the above types of scenarios and assumptions to address the following problem: how imprecise and sporadic humans inputs about the *spatial presence of potential hazards* for an autonomous robot can be effectively modeled and exploited to build and maintain an environment map for *safe path planning*. Humans inputs are assumed to be expressed in a *colloquial language*. Moreover, we assume that hazards might be *hardly, or not at all detectable by the robot sensors alone*. This implies that human inputs might be *necessary* to *plan* and *navigate safely*, avoiding hazards. Addressing the described problem, raises a number of general scientific questions:

- How do we model the inherently sporadic and imprecise *human sensor*?
- How do we *fuse* robot's sensors and human sensor information?
- How do we include fused, imprecise information in a *map* for *planning*?
- How do we use the map for finding paths that are *survivable* with respects to the hazards in the environment, and are *efficient* with respect to some predefined performance metric (e.g., time)?

The **contributions** of this work precisely consist in trying to provide answers to these fundamental questions, and include: modeling sporadic, verbally expressed human inputs using the formalism of *imprecise probabilities* [2]; building and updating an *imprecise probability map* that integrates robot sensory data and human inputs; a Rapidly Exploring Random Trees (RRT) algorithm specifically designed for searching for *survivable paths* on an imprecise probability map. *Performance* of proposed solutions is studied in simulation using ROS/Gazebo and compared with standard approaches. The study shows the overall effectiveness and flexibility of integrating imprecise probabilities and probabilistic maps for *survivable and efficient path planning* based on sporadic human inputs.

2 Related work

In human-robot interaction, when it comes to communicate and model *spatial information* (e.g., the presence of an obstacle), a bulk of research works have aimed to to find methods that provide to the robot spatial information which is as precise as possible. One way to obtain this is by *multi-modality*, such as in [17, 10] where gestures and speech are combined. Here we restrict ourselves to the use of *verbally and colloquially expressed information*. The motivation is twofold: we want to rely on minimal assumptions, and at the same time we aim to address scenarios such as emergency-response where humans most often have their hands busy with their tasks and do not have time or training to provide "precise" information to a robotic agent.

Imprecision is inherently rooted in colloquial expressions. Hazard warning from a human is expected to come in the form of sentences such as: "Watch out, there is a hole in the ground!", or "Watch out for the hole in the ground, which is close, in front and to the left of you!". These sentences do not provide precise metric information about relative location, shape, and dimension of the hazard. This is common when using a natural language to describe *geometric and spatial relations.* An extensive body of research is precisely devoted to these aspects (e.g. [11]), especially in the context of GIS, both outdoor and indoor, and for interactive systems [15, 12]. For instance, the first sentence above expresses a purely *topological relationship*, but doesn't provide neither directional nor geometric information. These are given in the second sentence, in the form of *projective relations*, which are however still insufficient to precisely plan a survivable and efficient path. In general, the availability of qualitative spatial relations [9] might not be sufficient for performance-efficient path planning, they need to be transformed into quantitative ones, e.g., through an iterative loop of inputs capable altogether of precisely describing the spatial scene [12]. In our settings, we assume that human inputs, in addition to be imprecise are also sporadic, such that a sequence of inputs, or even a dialogue, are not contemplated. This sets our problem scenario and approach at the edges of current methods.

Scenarios featuring *hazards*, explicitly ask for performing *survivable path planning*: finding and strategically selecting the path that (at least) guarantees robot survivability with high probability. An environment presenting hidden hazards for the robot can be seen as non-strategically *adversarial.* [1] provides a survey of *adversarial robotics*, and points out that the *uncertainty* has received relatively little attention so far. In this work we emphasize that the robot has to make path planning decisions mostly based on one-shot information from the human. The uncertainty associated to this information is inherently large because of the imprecision in colloquial language, and it is also hard to model, since different humans have a different perception of spatial measures. In [8] the problem of *Safe Navigation in Adversarial Environments* is formalized. A number of algorithms and theoretical results are presented based on a game-theoretic approach on a map graph. The authors assume that relevant information about the opponent/threat can be incrementally gained over time, otherwise a generic risk-aversion approach is adopted. In our work we assume that sufficient information about the threat won't be incrementally gathered, neither from the human nor from on-board sensors. Our challenge is precisely how to strategically model and use imprecise information to balance risk minimization and performance optimization. Other works focus on multi-robot scenarios across adversarial environments, such as [13], that however are different from our scenarios since they can leverage redundancy and parallelism.

Survivability (or *traversability*) planning has been extensively studied also in general graphs, such as in the Canadian traveler problem [6], stochastic shortest path [3], and robust path optimization [4]. However, we cannot reduce our scenarios to these problems as we don't have precise probabilities between nodes, nor do the techniques for many of these problems work well practically for large and dense graphs such as occupancy grids.

To model the *uncertainty* related to human's inputs, instead of using the "typical" Bayesian approach we rely on *imprecise probabilities* [2], that are in a sense more general and reduce the requirement for setting up a probabilistic model of human's imprecision. Up to our knowledge, imprecise probabilities have found limited application in robotics. Imprecise probability maps are referred to in [5], but are grid maps with a Beta distribution to model cell occupancies, which is different from our imprecise probability map representation based on upper and lower estimates of occupancy per cell. Furthermore, these maps were not used in path planning, but to find the minimum number of inspections to certain the existence of a target. Instead, imprecise probabilities have been extensively used to model sensory information in the absence of reliable models. It's common to "approximate" imprecise probabilities by using Bayes approaches with very flat priors such as Dirichlet priors [7], instead, we face directly the issues related to lack of models for the conditional probabilities of a "human sensor".

3 Problem scenario: description and challenges

We consider a generic scenario where a mixed *human-robot multi-agent team* is deployed. Each agent is given a set of tasks to deal with, together with a map that provides some (incomplete) description of the environment (i.e., maps are only partially reliable). Robotic agents are autonomous mobile robots. We don't make any particular assumption on robot's capabilities. The main point be that while navigating in the environment robot's sensors alone might be insufficient to timely and precisely detect environmental elements that constitute a *hazard* to the robot. E.g., the described scenario well-fits a post-disaster rescue mission.

The given map provides some initial support but for searching and localization, and gets dynamically updated on the field through SLAM operations. In order to reach out assigned targets on the map, a robot performs *path planning* and *local navigation* in the *partially observable and partially known environment.* Given the presence of unknown hazards, the robot face the challenge to find navigation paths that are both *survivable* and *efficient.* The notion of path survivability serves to quantify the *risk* for the robot, by specifying the probability that a selected path will be safely navigable given robot's characteristics.

In this work we aim to set up a mixed initiative system that leverages the presence of nearby humans to provide to robots information that can *complement data from on-board sensors* and can eventually help to *avoid threats.* The challenges come from the fact that we assume that humans' inputs are *sporadic*, often *one-shot* information, and are given in a rather *colloquial*, inherently *imprecise* way. In other words, we regard the human as an *imprecise, expensive, external sensor that can be accessed in a very limited way.*

Our assumptions are quite restrictive and general at the same time, and set the need to perform a conversion of the sporadic human input into *quantitative data*. More precisely, we need to define the conversion into a model that specifies distances, shapes, and so on, with some uncertainty. One obvious way of proceeding would consists in the adoption of a *Bayesian approach*: make a model of the "human sensor", start with some prior map, revise it with human input based on the human sensor model, plan based on the updated map. This is one of the standard approaches in modern probabilistic robotics [18]. One difficulty in this respect lays in the *lack of a reliable model for the generic "human sensor"* to use in Bayes' update formula. Moreover, this way of proceeding works well when repeated updates are available, such that revised, more accurate posteriors can iteratively be computed. However, we are assuming sporadic inputs from the a quite imprecise human sensor. To face these issues, here we propose to use a model based on *imprecise probabilities*, which is discussed in the next section.

4 Map representation using imprecise probabilities

We consider maps in the form of *occupancy grids*. Bayesian methods have been widely employed to create and maintain maps as probabilistic occupancy grids based on inputs from "common" sensors (e.g., range finders). Updating the map involves updating the occupancy probability of a cell in the map given a new piece of evidence. Evidence is provided by a sensor with a known probability distribution model: $P(O|S) = \frac{P(S|O)P(O)}{P(S)}$ where the random variables O represents the probability that a specific cell in the occupancy grid features the presence of an obstacle or hazard, and S is the observed value returned by the sensor.

In order to perform the update and revise the posterior conditional probability, a *probabilistic model of the sensor and of the environment* is required. In the case of a *human sensor*, how do we model it? It becomes problematic: we need a probability distribution for the observation made by the human, and a probability distribution of the human observation given the occupancy value of the cell. How do we reliably build them for a single individual accounting for the inherent imprecision and vagueness of his/her expected inputs? We need to give a precise, quantitative meaning to statements such as "in front of you", "near you, to your left", "ahead of you", "a large puddle", that do not contain (and are not expected to contain) precise statements such as "there's a hole 3.22m 23° east your current heading, and it's shaped as a circle of 0.8m radius".

Moreover, how do we make a model that reliably *generalize to different people*? Do the above expressions acquire a different quantitative meaning depending on the person and perception of dangers, distances, orientations? Likely, we would need to empirically build/learn a model for each different human agents the robot could interact with. This is quite unpractical and very prone to errors.

In our scenario, the robot can't neither gather multiple data from the human (one-shot input in the limit), nor get very reliable information from its own sensor (at least about hazards). Under these assumptions, the lack of a reliable sensor model in Bayesian updating can have a significant impact on the reliability of the resulting map, and therefore on path planning. A wrong model would create an initial bias that would hardly be canceled out later on, which is instead the case when multiple consecutive data inputs are made available.

In order to deal with these issues, we propose the use of *imprecise probabilities* as a replacement of Bayesian modeling. We will still have to set up *some* approximate model for human's inputs, however the impact of our approximations are expected to be less critical than using a purely Bayesian approach. Imprecise probabilities allow us to model the inherent imprecise nature of the human input by not forcing us to assign a precise probability to an event, but rather define a *range*. In the crudest sense, imprecise probabilities, can be thought of as *interval probabilities* with an upper estimated probability and a lower estimated probability associated with each event [2]. A Bayesian probability value can be thought of as the most likely value for the probability of an event as shown by the distribution curve, while the imprecise probability is a uniformly distributed uncertainty between the lower and upper probability values. Imprecise probabilities also provide us a way to automatically capture the *confidence* in the probability of an event through the range in upper and lower probability, which is not possible with Bayesian probabilities. The closer the upper and lower probabilities are to each other the more the confidence in the probability.

Imprecise probabilities are a generalization of Bayesian probability, as they converge to Bayesian probability when the upper and lower probabilities are the same. The upper probability is considered to be an estimate of the *plausibility* of a certain event, where the lower probability is considered to be the *certainty* of an event [14]. Unlike with Bayesian probabilities, where we have one value for the probability of occupancy per cell, with imprecise probabilities we make use of *two separate occupancy maps* to represent imprecise probability occupancy grids: the *belief map*, with all the lower probabilities of the cell being occupied, and the *plausibility* map with the upper probabilities of the cell being occupied. To have a consistent view of the world, the maps are updated as new sensor readings and human information arrive, which is discussed in the next section.

5 Map updating with on-board sensors and human inputs

In our problem scenario, the main task of the robot is to perform survivable path planning and navigation, given that the environment is partially observable, partially known, and may include hazards. At this aim, the robot needs to dynamically build and update a map representation of the surrounding environment. The robot starts with a environment map which is only partially filled. The on-board sensors are used to update this map, which is in the form of a *probabilistic occupancy grid*, as well as to localize, by doing SLAM (e.g., using common ROS packages). We refer to this environment map as the *sensor map* (SM). This is the map used by the *local planner* for navigation. If / when human inputs become available, the provided information is used to create/update the *human input map* (HIM), which is an *imprecise probability map* of the environment consisting of two maps, one of upper and one of lower probabilities. Following each human input, SM and HIM are *fused* to create an *imprecise probability map* (IPM), a rich representation including both on-board and human sensory information. The IPM is further converted to a *probabilistic occupancy map for planning* (POMP), which is passed to a planning algorithm, a modified RRT, to find the most survivable path towards an assigned target pose. In the following we explain how the HIM is constructed from human inputs, which is illustrated in Figure 1. In the next section we detail how the two maps in the HIM are fused to obtain a single POMP environment map for survivable planning.

In our model, human inputs are provided to the robot by *speech*. We can use APIs of available speech recognition systems to convert the input from natural language to a *spatial specification data structure*. In practice, we have a simple parsing system that extracts the properties of an imprecisely described hazard or obstacle and assign them as fields of a custom data structure, that includes:

- *Object*: The type and size / shape of object (e.g., large hole).
- *Range*: Distance of obstacle/hazard from the robot (e.g., close, far).
- *Direction*: Direction in which the obstacle/hazard is located with respect to robot's current orientation (e.g., left, right).
- *Message*: Additional useful information (e.g., be careful!).

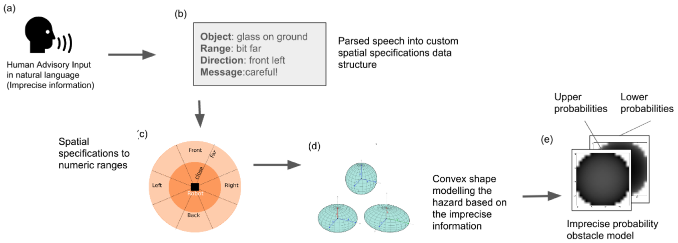

Fig. 1. Processing human inputs: (a) speech recognition, (b) parsing input to spatial specification structure, (c) quantifying imprecise inputs by given templates, (d) using convex shape to model hazard, (e) converting model to imprecise probabilities map.

Spatial specifications need to be converted into numerical ranges before they can be used to update the imprecise probability maps. Conversions are performed by *template matching*. The first conversion regards the *direction* of the obstacle/hazard with respect to the robot. We have adopted a simple template with 8 slices (Figure 1(c)). More slices can be used for increased precision, although more slices means more "priors", which is precisely what we want to avoid.

The conversion of the notion of *distance* is more problematic than that of direction, being more subjective. In an earlier contribution [16] an experimental study with 8 subjects was performed to analyze the variability of the estimated distance in relation to a moving robot. A positive correlation between the notion of "close" distance and the velocity of the robot was observed. As robot's velocity increases, the perception of a "close" distance seems to increase. Data collected during the experiment have been used to create a quantitative model for translating human's qualitative expressions of distances into numeric values.

Size and shape of the hazard are something that we cannot easily determine from the human input, especially if such information is not explicitly provided.

Our approach is to model the hazard as a symmetrical shape, as our uncertainty about shape properties is high and no preferential direction of smaller uncertainty is given. Hence, we model the presence of the hazard with an *ellipsoid*, a shape that has convenient properties: it's parametric, convex, symmetric and stretchable along the two axes. The uncertainty model that we adopt has a confidence that decreases towards the edges of the hazard. To represent this, we use the lower hemisphere of the ellipsoid to model the upper probabilities and the upper hemisphere to model the lower probabilities (see Figure 1(e)).

Once the human input has been quantified in the ellipsoid model, this is used to generate a new HIM. The SM is used by the localization algorithm to set robot's pose and where to "place" the ellipsoid in the HIM. In turn, the IPM gets updated as follows. For each environment cell, the upper probability value gets the max value between the current cell value in the SM and the corresponding cell value from the upper probability map of the HIM. The lower probabilities in the IPM are set as the min value between the cell value in SM and the corresponding cell value from the lower probability map of HIM. These updating criteria ensure consistency in the definition of imprecise probabilities and allow to consistently *fuse human and sensor inputs.*

6 Survivable path planning

The goal of the robot is to find the most survivable path(s) to navigate to an assigned pose. At this aim, two main steps need to be accomplished. First, create a *unified planning map*, the POMP map, from the IPM. Second, *find the most survivable paths* using the POMP. The creation of a single POMP map is motivated by the fact that doing path planning with two different sets of values (as in IPM) can quickly become cumbersome and would require a metric for survivability to be extracted from both the upper and lower probabilities of a cell, and combined somehow. Instead, having one single map representing the survivability metric for each cell makes computation faster and makes it easier to modify existing path planning algorithms to include the notion of survivability. For these reasons, we create the POMP map from the two imprecise probability maps in IPM by the following non-linear combination (for each cell in the map):

$$P = \underline{P}\Big(1 - (\overline{P} - \underline{P})\Big) \tag{1}$$

The formula defines the *survivability P of a cell* as the lower probability, which is the current belief, times the confidence in the probability values of the cell, which is given by the complement of the difference in probabilities between upper and lower probabilities. The planning map resulting from the imprecise probability map representation is published to the planner, that aims to find a path whose overall probability of crossing a hazard is acceptably low.

To find a path that is highly survivable in this respect, we modified a sampling based planning algorithm, the popular *Rapidly-exploring Random Tree* (RRT).

As preprocessing, we dilate the obstacles by the radius of the robot to define robot's configuration space. This ensures that we only sample points that the robot can safely be positioned at. Similarly, the *local planner* dilates obstacles to create a local cost map for navigation.
In the RRT pseudo-code to the side, the *max risk* term denotes the max value of survivability that the planner is willing to sample before considering a cell unsurvivable. The tree T contains all the nodes that are sampled and have a survivable path to them from the current pose.

```
T.init()
currentRisk = min(Map)
While(PathNotFound):
   point = SampleMap(currentRisk, targetBias)
   If currentRisk >= maxRisk:
      break
   If point == None or iter >= maxIter:
      currentRisk += stepSize
      Iter = 0
   onTree = FindClosest(T, point)
   If Distance(point, onTree) > delta:
      point = PointInDirection(point)
   If !Collision(onTree, point):
      T.add(point, parent = onTree)
      If point == target:
         break
   Iter += 1
```

The current risk to start out is set to the lowest possible risk from the whole map. If we use up all the lowest risk cells, we will have to increase the risk by some predefined step value, until we either find a path to the target or we reach the max risk threshold. At each iteration, we first sample from the map by only looking at cells with a risk less than or equal the current risk, and use this as the random point for growing the tree. Every n samples, the random point equals the target cell, to encourage exploring towards the target. The closest node in the tree to the random point is found. If the random point is within an L2 metric distance `delta` and there's no obstacle in line of sight, the random point is added to T. If it is not a point `delta` away along the closest node, the random point vector is checked for collision with any obstacles between them, if not, it is added. Otherwise, we move onto the next iteration. If a max number of iterations is reached, the current risk level is increased, and new iterations start, so as to continue exploring the next most survivable cells.

Hence, by systematically sampling for the most survivable cells first, we can nearly guarantee (in probability) to find a path that is the most survivable. Yet, the risk level might be too high to use, which is in the end a *strategic decision*. In this case, the robot could be entitled to ask the human for *further input or better specifications*, that would help, for instance, to reduce the size of the ellipsoid modeling the hazard, or being more precise defining its eccentricity.

7 Implementation, results, and discussion

The described model and path planning algorithm have been implemented in ROS and evaluated in simulation using Gazebo. The Turtlebot 2, equipped with a depth camera, has been adopted as robot platform.

The implementation consists of two main components: (i) the imprecise 2D costmap IMP, which consists of two 2D costmaps – corresponding to the lower and upper probability maps – and combines the human input with the sensory data; (ii) the modified RRT planner, that uses the POMP probabilistic cost map derived from the IMP. Both components are integrated into `move_base`, the ROS package for mobile robots navigation. We configure `move_base` to use our RRT implementation as global planner, and the POMP map for global planning. Instead, the local planner uses a map equivalent to the SM one. The robot localizes itself with the depth sensor using the `amcl` localization driver.

The purpose of the simulation study is to evaluate *dependability* and *efficiency* of the proposed approach. The *key performance index* is quantified by the ability to find paths that guarantee robot *survivability* (facing the presence of hazards) and at the same time reach out a target pose in a *short time*.

At this aim, we have set up simulated worlds featuring relatively large obstacles randomly placed in an area of 15×15 m^2. In the *cluttered* reference scenario the density of obstacles is higher than in the *semi-cluttered* case. In both scenarios a large pile of debris is placed in the center of the world. The pile plays the role of the hazard that the robot can potentially fail to detect while moving.[1] To study path planning, four different start and end poses are defined, located at opposite corners. Each simulation begins with the robot in a start pose, and with the end pose at the opposite corner set as a navigation goal to `move_base`.

To assess dependability and efficiency of our approach, we have considered four different planning and navigation setups. The first three correspond to somehow *standard* approaches to the problem (including Bayesian updating), not using imprecise probabilities. The last one corresponds to our model. More specifically the setups are the following. (a) The pile of debris has been mapped and the robot uses the default `move_base` global planner (`GlobalPlanner`). (b) The pile of debris has not been mapped but, while navigating, the robot can detect it without external noise, and can update accordingly (in a Bayesian way) the local probabilistic occupancy map. (c) The pile of debris has not been mapped, frequent (0.5 Hz) but rather noisy sensory information is given to the robot, and the default `move_base` configuration is used with Bayesian updating (this may simulate repeated inputs from a human sensor). (d) The pile of debris has not been mapped, the robot is given one sporadic advisory information (representing the human input), and `move_base` is configured to use the imprecise probability framework and the RRT planner as global planner. Other navigation parameters are common to all four setups, and have been tuned accordingly.

In each simulation, the performance metric is the time it takes to arrive to the end pose. For each start and end pair, and setup, we performed 20 simulation runs to account stochasticity. Since setup (a) represents an ideal situation we use its average performance to benchmark the other setups. In the plots, setup (b) is indicated as *standard*, setup (c) as *noisy sensing*, and setup (d) as *imprecise*.

In Figure 2 we report the observed navigation times (normalized to the ideal setup average). Results suggest that the imprecise probabilities framework achieves good performance relative to the other setups, being comparable in me-

[1] Since the pile can be actually sensed to some extent by the simulated depth sensor, we have added Gaussian white-noise for the sensor data corresponding to the pile. In this way we can simulate a situation of failure detection in the experiments.

dian to the standard one and better than the noisy one. Moreover, it also lowers the variance of the navigation performance. The claim is that with sporadic inputs and imprecise probabilities modeling (supported by minimal prior models) a performance comparable if not better than methods requiring more data, and more accurate models can be achieved. Moreover, the imprecise framework can be successfully applied even when a hazard is not detectable at all by the robot, a situation that would see the other methods fail badly.

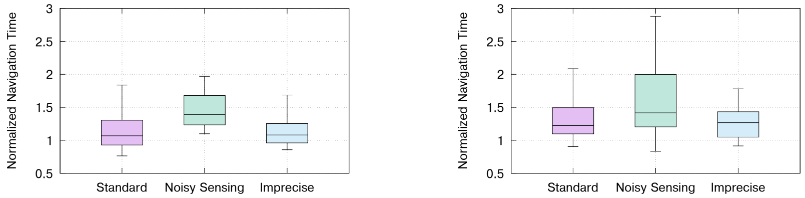

Fig. 2. Navigation results in semi-cluttered (left) and cluttered (right) environments.

Figure 3 show the 2D density map of robot positions for all the simulation runs corresponding to the same setup. In these density maps we observe that the trajectories in the imprecise setup are more spread and, in general, longer with respect to other setups. Despite this apparent suboptimality, the imprecise probabilities framework allows to effectively tackle scenarios where the robot sensors would not be able to accurately spot the hazards while providing minimal performance degradation and variance reduction.

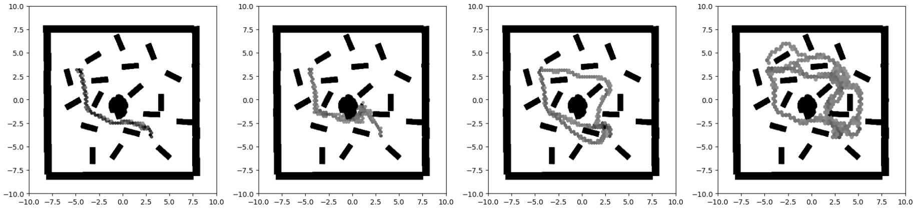

Fig. 3. Trajectories with baseline, standard, noisy, imprecise setups (from left to right).

8 Conclusions

Navigation in partly known environments featuring hazards hard to detect (e.g., post-disaster scenarios) is still a challenge for a mobile robot. We have considered how a robot can navigate avoiding hazards by exploiting inputs from a human agent. We have modeled the human as a sensor proactively providing sporadic (e.g., one-shot) and inherently imprecise information about spatial elements (e.g., hazards). To overcome difficulties adopting a Bayesian approach in this setting, we adopted the formalism of imprecise probabilities to model the human sensor and maintain an imprecise probability map where we fuse all available sensory information. The map provides the input to a modified version of the RRT planning algorithm, that finds paths that are at the same time survivable and efficient. The model and the algorithm have been implemented in

ROS and studied in simulation using Gazebo considering cluttered environments featuring hazards hard to detect for the robot alone. The combination of human sensor and imprecise probabilities has shown the ability to robustly find survivable and efficient (short traveling time) paths. Our approach has also shown a systematically good performance compared to more “standard” approaches to the problem, using Bayesian updating, that however may need way more data inputs and may not be able to avoid the hazard when it is not detectable by on-board sensors (without the external help of a human).

References

1. Agmon, N.: Robotic strategic behavior in adversarial environments. In: Proc. of IJCAI. pp. 5106–5110 (2017)
2. Augustin, T., Coolen, F.P., de Cooman, G., Troffaes, M.C.: Introduction to imprecise probabilities. John Wiley & Sons (2014)
3. Bertsekas, D.P., Tsitsiklis, J.N.: An analysis of stochastic shortest path problems. Mathematics of Operations Research **16**(3), 580–595 (1991)
4. Bertsimas, D., Sim, M.: Robust discrete optimization and network flows. Mathematical programming **98**(1-3), 49–71 (2003)
5. Bertuccelli, L.F., How, J.: Robust UAV search for environments with imprecise probability maps. In: 44th IEEE Conf. on Decision and Control & European Control Conf. (CDC-ECC). pp. 5680–5685 (2005)
6. Fried, D., Shimony, S.E., Benbassat, A., Wenner, C.: Complexity of Canadian traveler problem variants. Theoretical Computer Science **487**, 1–16 (2013)
7. Hutter, M.: Robust estimators under the Imprecise Dirichlet Model. In: Proc. of 3nd Int. Symp. on Imprecise Prob. and their Appl. (ISIPTA). pp. 274–289 (2003)
8. Keidar, O., Agmon, N.: Safe navigation in adversarial environments. Annals of Math. and Artificial Intelligence (AMAI) **83**(2), 121–164 (2018)
9. Kordjamshidi, P., Hois, J., van Otterlo, M., Moens, M.F.: Learning to interpret spatial natural language in terms of qualitative spatial relations. In: Representing space in cognition, pp. 115–146. Oxford UP (2013)
10. Lei, Z., Gan, Z.H., Jiang, M., Dong, K.: Artificial robot navigation based on gesture and speech recognition. In: IEEE SPAC. pp. 323–327 (2014)
11. Levinson, S.C.: Space in language and cognition: Explorations in cognitive diversity. Cambridge University Press (2003)
12. Levit, M., Roy, D.: Interpretation of spatial language in a map navigation task. IEEE Trans. on Sys., Man, and Cyb., Part B: Cybernetics **37**(3), 667–679 (2007)
13. Lyu, Y.H.: Implications of motion planning: optimality and k-survivability. Ph.D. thesis, Dartmouth College, Dept. of Computer Science (2016)
14. Miranda, E.: A survey of the theory of coherent lower previsions. International Journal of Approximate Reasoning **48**(2), 628–658 (2008)
15. Moratz, R., Tenbrink, T.: Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations. Spatial cognition and computation **6**(1), 63–107 (2006)
16. Pillai, R.: A mixed initiative system for survivable path planning in unknown cluttered environments. Senior Thesis, Computer Science, CMU in Qatar (2018)
17. Skubic, M., Perzanowski, D., Blisard, S., Schultz, A., Adams, W., Bugajska, M., Brock, D.: Spatial language for human-robot dialogs. IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **34**(2), 154–167 (2004)
18. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)