

Thinking in front of the box: Towards intelligent robotic action selection for navigation in complex environments using image-schematic reasoning

Mihai Pomarlan¹, Stefano De Giorgis², Maria M. Hedblom³, Mohammed Diab⁴ and Nikolaos Tsiogkas^{5,6}

¹*Institute of Artificial Intelligence, University of Bremen, Germany*

²*University of Bologna, Via Zamboni 32, Bologna (BO), Italy*

³*Jönköping Artificial Intelligence Laboratory, Jönköping University, Sweden*

⁴*Imperial College London, London, UK*

⁵*Department of Mechanical Engineering, KU Leuven, Celestijnenlaan 300, B-3001 Heverlee (Leuven), Belgium*

⁶*Core Lab ROB, Flanders Make, Gaston Geenslaan 8, 3001 Heverlee, Belgium*

Abstract

One of the problems an agent faces when operating in a partially known, dynamic, sometimes unpredictable environment is to keep track of aspects of the world relevant to its task, and, if possible, restrict its attention to only these aspects. We present our first steps towards constructing a system that combines image schematic knowledge and reasoning with reactive robotics, and which enables perception that focuses on, and keeps track of, relevant entities and relationships. While our approach is more reasoning intensive than is usual in reactive robotics, the formalism we use for inference is fast and allows an agent to adjust, in real time, the complexity of its action selection procedures according to the complexity of the relevant part of the environment. We illustrate our approach with a few simulated examples of robots performing navigation tasks. In some examples, interaction with obstacles is necessary to complete the navigation tasks, adding complexity to the scenario.

1. Introduction

Among the problems that a robotic agent has to deal with, navigation seems to be one of the most robustly solved. As an example, it was only in 2019 that, after 15 years of operation, the Mars Rover finally was declared “dead” because of dust preventing its solar cells from operating. Through all that time, it had to move autonomously on the martian surface; the communication lag with Earth was too long to make teleoperation feasible.

Clearly, algorithmic support exists to generate paths in environments with many obstacles. Algorithms such as A^* [1] or its version adapted to changing environments D^* [2] can efficiently search for shortest paths in (discretized) two-dimensional spaces. Randomized planning algo-


The Eighth Joint Ontology Workshops (JOWO'22), August 15-19, 2022, Jönköping University, Sweden

✉ pomarlan@uni-bremen.de (M. Pomarlan); stefano.degiorgis2@unibo.it (S. De Giorgis); maria.hedblom@ju.se (M. M. Hedblom); m.diab@imperial.ac.uk (M. Diab); nikolaos.tsiogkas@kuleuven.be (N. Tsiogkas)

🆔 0000-0002-1304-581X (M. Pomarlan); 0000-0003-4133-3445 (S. De Giorgis); 0000-0001-8308-8906 (M. M. Hedblom); 0000-0002-5743-5190 (M. Diab); 0000-0003-2842-7316 (N. Tsiogkas)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

gorithms such as RRT[3] can handle spaces with more degrees of freedom in a “probabilistically complete” fashion, eventually finding a path, if one exists, and in practice they find one quickly.

However, what such algorithms address is the problem of finding a path in an environment that is conceptualized as beyond the robot’s ability to change. Also, an understanding of the environment in terms of its possibilities for action and change are beyond the scope of a navigation module. This becomes important when path and trajectory planning and control methods, such as RRT[3], A*[1], TEB[4] etc., fail to provide a solution. If there is no deeper understanding of the reasons of the failure, there is little indication of how to overcome it.

For instance, boxes could be pushed out of the way; other agents (humans or autonomous agents) standing in the way might be asked to leave; and a box that was not a problem before should be pushed away to make room for another and so on. Such actions may also have consequences for other tasks the robot may be pursuing. To handle such situations, a robot would rely on higher level planning systems, such as state transition planning systems in which actions are treated as atomic components [5] described symbolically in something such as Planning Domain Definition Language (PDDL) [6].

However, planning operations are time-inefficient with their complexity growing exponentially in relation to the situation as both perception systems and action execution systems require constant updating to changes in the environment and the agent’s own states. It is important in such cases to keep track of the relevant data about the environment, and, if possible, only the relevant data – as anything else is a potentially costly distraction. Further, in order for a planning method to work it will need knowledge of both the environment as it is, and of possibilities of action in general. In consequence, methods based on planning alone are seldom enough to deal with the complex and dynamic real-world scenarios. For instance, in a situation in which a box is blocking the planned trajectory, planning alone will not be able to adapt to this change, unless the system has a deeper understanding of what this blockage means, how it effectively can bypass it or move it, and, likely, it will also need to focus on a comparatively small part of the environment for the planning search to be feasible. Thus, for difficult situations extending over longer time periods or that require unexpected manipulations of the environment, conventional planning methods will not be enough. At the same time, for many routine operations where repeated perceptual search and drawing complex inferences are unnecessary, the solution can instead be sought on the behavior level.

In this paper, we explore how much decision-making power can be created exclusively from the perception inputs and behavioral outputs of the robot, on which very simple inference mechanisms operate. We do not claim a slower, deliberative layer that performs planning is unnecessary. We claim that such a layer should be supported by a behavioral layer that keeps its focus on relevant parts of an environment, where this relevance is decided by the task a robot performs, and the situations it finds itself in. Our work is inspired by the reactive, behavior-based robotics approach pioneered by Rodney Brooks [7] with two-fold novel contributions.

Firstly, we employ inference techniques that, while tractable and much simpler than planning, allow for more dynamic rewiring of behavior than what is traditionally seen in reactive robots. In particular, our methods allow the robot to adjust the set of entities and relationships it considers relevant in a given situation. Secondly, we build on recent work on the formalization of image schemas [8] (sensori-motor patterns of embodied experience [9]) and encode knowledge for robotic perception/attention reconfiguration. The hypothesis we pursue in the long term

is that the information present in the image-schematic knowledge will allow our system to reconfigure “the rules” of environmental changes, as opposed to an approach that would hard-code information about objects and their interactions.

In this paper, we take a first step in this direction. We are here interested in how to automatically detect potential disturbances to the nominal task execution from a robot. This requires the robot to decide which elements of the environment are interesting based on the executed task and accordingly monitor them. In short, what we are tackling in this paper is:

1. a robot navigation problem, namely identify elements of a certain “saliency” in the navigation environment (paths, blockage, etc.) and ask questions about their status;
2. managing the action-selection loop, in this case restricted to selecting what perception queries to perform;
3. cost-efficiency problem, where the method of inference is close to the behavior-level, to enable the robot to respond quickly to changes in the environment;
4. commonsense reasoning problem, since the knowledge for inference is expressed in image-schematic terms to assist in its reuse and adaptability of the robot.

2. Theoretical Foundation and Related Work

2.1. Embodied Cognition and Image Schemas

Embodied cognition is a commonly applied research paradigm for cognitive robotics (e.g. see [10]). It proposes that all forms of cognition, and by formal extension, robotic behavior, can be traced back to abstracted information in the embodied experiences [11]. Through repeated exposure to particular spatiotemporal relationships between objects, agents, and environments, salient features are extracted and formed into conceptual schemas that can be repeatedly reused for analyzing similar scenarios. These abstracted sensori-motor patterns are called *image schemas* and have been defined by Mandler as “...dynamic analog representations of spatial relations and movements in space” [12, p. 591]. To this end, they have been applied to describe information transfer in analogical reasoning [13] and to describe the cognition structure in event conceptualization [14].

While stemming from cognitive linguistics, image schemas have seen increased popularity to solve issues surrounding symbol grounding and commonsense reasoning problems in artificial intelligence [8] and more interestingly, due to their relationship to affordances and functional relationships [15], research to integrate them into cognitively-inspired robotics systems to simulate more intelligently behaving robots (e.g. [16, 17]).

Due to their interdisciplinary foundation, there exists no agreed-upon list of image schemas. Likewise, the most optimal method to formally represent these abstract concepts in terms of their internal taxonomy and relational meronymy remains uncertain (for examples see [8, 18, 19]).

In this paper, we consider a minimal scenario, as exposed in detail in Sec. 4.3, namely a robot moving on wheels from a starting point to some destination, which should be able to move unless it encounters some blockage in the form of an obstacle (object) in front of it, entering in a relation of contact with it. To represent this apparently simple scenario, from a cognitive robotics perspective, it requires the introduction of some of the most relevant image schemas, here presented together with their conceptual components:

SOURCE_PATH_GOAL: movement from a source or starting point, following a series of contiguous locations leading to a goal or endpoint [9]. In its dynamic form, it is present in all forms of movement of an entity [20].

CONTACT: the physical relationship in which the surfaces of two entities are touching.

SUPPORT: CONTACT between two objects in the vertical dimension [21], assuming some consequences related to the forces exerted by the above and below entities being such that the supported entity does not fall.

BLOCKAGE: the complex construct in which planned movement is prevented by an obstacle. It can be described as a force vector encountering a barrier and then taking any number of directions [9].

2.2. Task and Motion Planning

Task and Motion Planning (TAMP) plans the tasks to be done by the robot, both at an abstract level, considering only atomic actions, and at a geometric and/or control level that considers how these actions are to be carried out in detail. TAMP techniques are used to adapt to the actual state of the environment, and comply with constraints both at task and geometric levels.

There are two dominant approaches in the task planning domain, one based on search algorithms and the other based on knowledge and reasoning. The former mainly uses PDDL [22] to describe the world. Although this way of description can easily handle tasks with many actions, and integrate the geometric (motion) constraints [23], it makes a closed world assumption, i.e., if some facts about the world are not known or change, a planner may not be able to find a solution. This limitation means that robots are not able to begin a task until all objects in the environment are known and the actions the robot can do on them are completely defined. The latter, knowledge enabled approach, has emerged as a new domain of planning, aiming at making the robot able to flexibly perform manipulation tasks like [24]. This approach can easily integrate the knowledge from the environment and adapt the action to be done accordingly.

The increasing emphasis on real-world applications has led researchers to develop algorithms and systems that more closely match realistic planning problems in which manipulation skills play a significant role. Manipulation problems are referred to as problems in which robots handle objects using a set of primitives, e.g., pushing, picking, or placing. Due to task constraints, the limitation of generic motion planning emerges, and the robot is required to displace objects when there are no feasible solutions between two robot configurations. This requires TAMP to deal with different types of robotic manipulation problems, ranging from single or multiple collaborative mobile robots navigating among movable obstacles to complex higher-dimensional table-top manipulation problems carried out by dual-arm robots or mobile manipulators.

The key challenge in real scenarios that include manipulation skills is to make the robot adaptive to the changes that happen in the world. This requires a sophisticated system that updates the state of the entities in the environment and feed the planning system with the changes that occur. In this line, researchers have investigated how to keep updating the state of the environment. A manipulation planning framework with perception capability has been proposed [25] and [26]. The former integrates a multi-modal sensory system to infer the

line-of-sight, and non-line-of-sight objects and store the records in the database as experiential knowledge to adapt the robot behaviour in similar situations based on the new states of the world. The latter optimizes over Cartesian frames defined relative to target objects. The resulting plan remains valid even if the objects are moving and can be executed by reactive controllers that adapt to these changes in real-time. In [27] learning manipulation skills from a single demonstration are proposed where a robot is shown a manipulation skill once and should then use only a few trials on its own to learn to reproduce, optimize, and generalize that same skill.

2.3. Situational awareness and assessment

For a robot to successfully operate in the world and achieve its goals, a general understanding of the surrounding environment and the dynamics of the processes taking place there, are required. In a similar fashion, humans that execute tasks in the real world, are using cognitive processes to form an understanding of their surroundings that allows them to take appropriate actions towards their goals. This task-specific knowledge and understanding is formally referred to as *situation awareness* (SAW) [28], and is studied in the scope of understanding how humans perform tasks so that accidents can be minimized. It is divided in three levels:

Level 1: Refers to the perception of elements from the environment that are relevant to the executed task.

Level 2: Refers to the comprehension of the current situation based on the perceived elements, and any relations that are formed among them.

Level 3: Refers to the projection of the current situation to the future, allowing the individual to make predictions regarding the possible evolution of it.

Given the situation awareness, an individual can plan its actions accordingly, so that they achieve their goals. It must be noted that the three levels of situation awareness are not forming a linear relationship, i.e. a complete knowledge of one is not required to form the other. On the contrary, they all work in parallel, continuously updated, as the situation evolves. Once the individual needs to make a decision, they can use the current best estimate of each level to guide their decision process.

To achieve situation awareness, an active process of gathering knowledge is required. Such an active process is referred as *situation assessment* (SAS). During situation assessment, an individual focuses their attention towards the elements of the environment that are important for the task at hand. In addition, they make the required connections between perceived elements given background knowledge. In an analogous way, an artificial agent, such as a robot, can assess the situation given its task, by using sensors to measure the environment, and algorithms to process the sensor data.

3. Problem setting

In this paper we focus only on reconfiguring an agent's perception, as opposed to also its actions. Our setup then is that a "monitoring" agent, implementing our image-schematic approach,

watches over a collection of robots each navigating to its own goal. The robots do not interact with the monitoring agent. Even so, the agent will keep track of what is relevant for navigation, and seek to form an understanding of the relevant relationships between objects such that it can foresee and suggest solutions for problems in navigation.

3.1. Competency Questions

The operation of our monitoring agent involves reasoning to direct its perceptive attention and update its understanding of the world. To develop this reasoning procedure, we have first defined a set of competency questions it must address. As part of its behavior, a robot must focus its perception on aspects of the environment that are plausibly important (or, “relevant”) for the achievement of its goals. Or, in the case of our monitoring agent, it should be able to take the perspective of the robots it monitors. Following are the competency questions we defined:

1. which entities are directly involved in the robot’s goals?
2. what relevant relations exist between entities directly involved in the robot’s goals?
3. what are other relevant entities and relationships in the environment?
4. are there entities/relationships that cease to be relevant?

Questions 1 and 2 are there to provide a basic understanding of a situation. When navigating, there is a trajectory and a goal for example, and relevant relations pertain to their spatial arrangement. However navigation situations can become more complex, hence the need for question 3, which allows entities to enter into the agent’s attention. Thus, an obstacle becomes relevant, and so may be an other entity that prevents the obstacle from being moved out of the way. Question 4 is important because it provides a cleanup mechanism for the agent’s attention.

4. Proposed approach

We implement our monitoring agent as a combination of perception procedures that are called depending on the results of a reasoning process. The reasoning process takes into account information about the current situation of the environment, the robot’s goals, and previously available perception data about objects. It is fast, and integrated into the perception-action loop of the agent. The knowledge going into the reasoning process also comes from definitions of image schemas and their relationships. In the next subsections we will provide detail about the image schematic knowledge and the organization of the image schematic reasoning process.

4.1. Frame Semantics and Description & Situation Pattern

To model image schemas as framal structures we adopted Fillmore’s frame semantics [29]. Frame semantics has been most influential as a combination of linguistic descriptions and contextual knowledge to describe cognitive representation of phenomena occurring in the world. Frames are schematic structures representing prototypical and recurrent elements in order for some entity, event or situation to realize. Fillmore describes frame-based structures to notions such as *experiential gestalt* [13], stating that frames can refer to a unified framework of knowledge or

a coherent schematization of recurrent experience. For example a simple action like *walking*, to be represented in its formal structure, would require some *necessary roles* such as: a subject of the action (agent) and some spatial extension covered by the walk (path), but also some external elements such as the duration of the walk (time), and optional elements could be expressed, e.g. the weather conditions during the action, the curvilinear shape of the walk, the unexpected stumbling stone encountered, nesting more complex knowledge in more specific frames and scenarios. Lexical units (words and phrases) are associated with frames in order to fully understand their semantics, based on mental schemes representing these evoked scenes. In FrameNet [30], frames are also explained as *situation types*. Therefore, in our reasoning module, image schemas are represented as frames, while their spatial primitives are represented as roles, and the detection of one of its roles, allows the inference, and gestaltic activation, of the whole image schema, as described in Section 4.2. We furthermore reuse the Description and Situation ontology design pattern [31], which allows the introduction of a constructivist perspective in the ontological module. In our work, as described in section 4.3, the occurrence of e.g. some robot moving towards its GOAL is treated as the occurrence of an image-schematic situation, namely, in the above mentioned example, as a SOURCE_PATH_GOAL situation.

4.2. ISL2OWL

Any formal research aiming to utilize the semantic richness of image schemas has to deal with the complexity of formally representing the full range of abstract conceptualisations associated with these spatio-temporal relationships. One formalization method developed for this purpose is the Image Schema Logic, ISL^{FOL} . Greatly simplified, it combines spatial mereology in the form of Region Connection Calculus (RCC) [32] in Euclidean 3D space, with Qualitative Trajectory Calculus (QTC) to describe relative movement between objects and regions [33] framed with Linear Temporal Logic (LTL) over the Reals to allow for sequential changes (For a complete account of the logic's syntax and semantics see [8, 34]).

In our work, we abstract away from the richer representation in ISL^{FOL} by transposing them into the Web Ontology Language (OWL2) following a frame semantics approach [29], called ISL2OWL¹. While this reduces the semantic richness of the image-schematic concepts to its minimal structural elements, it also allows us to seamlessly integrate them into our cognitive robotics framework. Thus, in ISL2OWL ontological module each Image Schema is modeled as a frame, taking as necessary roles its spatial primitives. E.g. SOURCE_PATH_GOAL takes as necessary roles three elements: SOURCE, PATH, and GOAL. Due to the gestalt, frame-based nature of image-schemas, the activation of one of spatial primitive triggers the activation of the whole image schema; e.g. knowing there is a goal also means knowing there is a path and a source.

¹The current ISL2OWL version used in this work is available at:

https://raw.githubusercontent.com/mpomarlan/robotics2022/main/src/ISL2OWL_4_Robotics.ttl ;

while ISL2OWL full graphs are available at:

<https://github.com/StenDoipanni/ISAAC/tree/main/ISL2OWL>

4.3. Image Schematic Reasoning for Action Selection

In order to answer the competency questions mentioned in Section 3.1, and to do so fast enough to be useful for a robot during its activity, we defined an image schematic reasoning layer (ISRL). It has its own ontological module, is written in a simpler formalism than OWL-DL and in which inference is guaranteed tractable, and contains some simple heuristics about image schematic situations and how these connect to each other. The tractability comes from the low complexity of the inference problem for this formal system – linear time in the size of the knowledge base fragment used to answer a query– as opposed to planning which in the worst case requires time exponential in the size of a planning problem.

The ISRL is queried once every perception-action loop of a robot or agent. It takes as input *prior knowledge* obtained via a process of inference and perception during the previous iteration, and *background knowledge*, i.e. knowledge that is believed true from other sources.

In this paper, we focus on the perception side, so the output of ISRL is *posterior knowledge* about how the situations around an agent are developing and a *perception tasklist*. The perception tasklist provides information about how a robot’s perception system should reconfigure itself to track environmental entities and relationships that are inferred to be relevant. The perception module would then act on this tasklist and produce new knowledge about the environment. The new perception knowledge, together with the inferred posterior knowledge will become the prior knowledge for the next iteration of reasoning with ISRL. In a more general case, the ISRL would also decide, for each of a robot’s actuators (or logical groupings of actuators, such as base, arms, head etc.) a set of actions to perform. Figure 1 shows an overview of how the ISRL is integrated into a robot’s/agent’s perception-action loop.

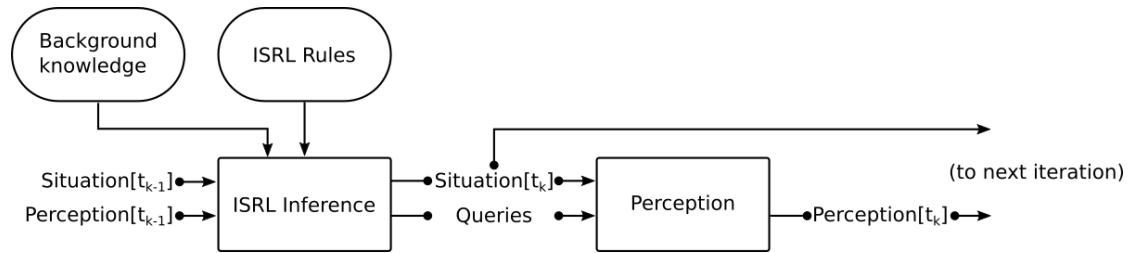


Figure 1: Overview of the information exchanged between ISRL and the perception module of an agent. Indices t_{k-1} , t_k refer to consecutive iterations of the perception-action loop. Situation[t] represents knowledge about a situation at time t, such as image schematic relationships and their participants. Perception[t] is a collection of facts established by perception at time t. Queries describe the information that perception is supposed to produce, and implicitly the processes it must run to do so.

The main benefits of this approach is that it is reactive, adaptable, and situation-aware. Reactivity here means simply that it can operate fast enough to be useful for quick cognitive processes that have to deal with a changing and sometimes surprising environment. It is adaptable in that the complexity of the perception queries scale, in a controlled way, depending on what is deemed relevant at each particular time. Finally, situation-awareness here means that the approach has and makes use of current knowledge of the situation, in other words a top-down understanding of the environment, to filter and make sense of a bottom-up stream of facts about that environment so that it can update its higher level understanding of the situation.

In more detail, the formalism in which ISRL is written is that of defeasible rules, with some limitations. A rule is formed of an antecedent, which is a conjunction of terms, and a consequent, which is one term. A term asserts either that some predicate is true on some collection of arguments (e.g., *inFrontOf(a, b)*), or that the predicate is false on a collection of arguments (e.g. *¬inFrontOf(a, b)*). The meaning of a rule is that, if its antecedent is established, then the consequent is defeasibly provable. We refer the reader to [35] for details about the proof theory of defeasible logic and how conflicts between rules are resolved. We have chosen the combination “ambiguity propagation”, “team defeat”, and “loop detection”.

One limit we place on our formal system implementation, is that predicates can have at most two arguments, which makes them formally similar to the triples commonly used in knowledge representation. Another limitation concerns the use of variables in the expression of a rule: all variables in the consequent must appear in the antecedent as well, i.e. no “existential rules”.

The rules in our system include, partially, information coming from ISL2OWL and domain knowledge about our running example for this paper, which is navigation by robots on a planar surface where obstacles may be present. The rules have been written by hand, however see section 6 for a perspective on how they could be at least in part generated by combining several knowledge sources. The knowledge we took from ISL2OWL is as follows:

The state of the environment is considered formally as a `dul:Situation`, which may have other `dul:Situation` participants – in particular, to account for the image schematic relations between objects such as an object being supported by another, or blocked by another. Each modification of relationships among entities is an `dul:Event`. This structure allows inferences about participants in image schematic situations.

Image schematic situations have particular kinds of participants. E.g., `SOURCE_PATH_GOAL` has a `isrl:goal` (among others), a `isrl:BLOCKAGE` has a `isrl:blocked` etc. A situation can have multiple image schematic meanings, and the same object may play several roles in it (e.g., both `isrl:blocked` and `isrl:goal`).

If two objects are not falling and are in `CONTACT` such that one is `isc:above` the other allows inferring a `isrl:SUPPORT` Situation with the two objects as participants.

The domain knowledge for our running example is that robots that move their wheels should move unless they are blocked, and that to block a robot – or other object more generally – then some obstacle should exist in front of the robot/object and be in contact with it. This helps in creating rules with which to infer what questions to ask of perception.

In order to decide what questions to ask, each object currently in the attention of the robot/agent is associated to a corresponding question individual, e.g. `isrl:about(q_box, box)`. Through inference, various information becomes attached to the question individual, such as what kind of question it is – and it can be several kinds of question, because several questions may be relevant about an object. For example, if we know an object is the trajector of a `isrl:SOURCE_PATH_GOAL`, it is relevant to ask whether it is moving, and what is in front of it.

```
isA(?S, SOURCE_PATH_GOAL), trajector(?S, ?R), about(?Q, ?R) => isA(?Q, AskIsMoving)
isA(?S, SOURCE_PATH_GOAL), trajector(?S, ?R), about(?Q, ?R) => isA(?Q, AskIsInSpatialRelation)
isA(?S, SOURCE_PATH_GOAL), trajector(?S, ?R), about(?Q, ?R) => hasMode(?Q, frontOf)
isA(?S, SOURCE_PATH_GOAL), trajector(?S, ?R), about(?Q, ?R) => hasRelatum(?Q, ?R)
```

Once ISRL completes inference, all the inferred symbolic facts are passed on to a perception

module. Primarily, this module is interested in triples asserting more detailed types for question individuals, because these assertions decide what functions will be called to answer the queries. However, the symbolic context provided by the other facts is also important to decide how will these functions answer their queries. For example, to return a list of objects that are in a given spatial relationship with another, the semantics of this spatial relation needs to be described. That is, it is not just interesting for perception to know that an individual q_turtle is of type `AskIsInSpatialRelation`, but also what kind of spatial modality[36] is being queried.

Even in our fairly simple scenarios discussed in section 5, this polysemous nature of spatial relations is important. It is relevant to query what is in front of the robot when it moves; it is also important to query what is in front of an obstacle in the robot's path, because whatever is in front of the obstacle might prevent the robot from pushing the obstacle out of its way. However, these two meanings of the "in front of" relation are not the same. In the first, we have a reference object that provides also the reference axis for what in front means – the robot, and its forward direction. In the second case, the reference object is the obstacle, but the reference axis is the robot's (the blocked object's) forward direction.

Finally, the perception module returns a list of new facts, such as states of (relative) motion of objects, movement of actuators, relative spatial relations. These facts will get propagated to the next iteration of the perception-action loop, where they will be fed into ISRL's inference, and the process repeats. Note that this allows a robot's scope of attention to change as appropriate to the needs of a situation. That is, objects may be added to its attention scope by being mentioned in the facts produced by perception. Objects can also drop from the attention scope when there are no relevant questions to ask about them and they do not get mentioned in the answers to the relevant questions.

5. Evaluation

To evaluate the proposed method for reasoning for situation assessment a series of experiments are performed using mobile robots in a simulated world performing a navigation task. Each robot is supposed to go to a particular goal, but the structure of the environment and the robots' actual actions will differ in the various scenarios.

Our scenarios run in PyBullet [37], a physics based simulation environment widely used in robotics research. The simulated robots are Turtlebot 2 robots, equipped with a depth camera sensor. An example setting of an empty environment with two robots trying to reach two goals can be seen in figure 2a. The simulated environments used in the scope of this work, represent situations where robots are required to physically interact with their environment in order to achieve their navigation goals. For example in figure 2b the purple robot needs to understand that its path is blocked by an obstacle and evade it. In the final scenario, seen in figure 2c the path of the purple robot to its goal is blocked by a box trapped between walls such that the robot cannot push it out of the way. In the current setting any classical path planning approach would fail. Instead, the orange robot can understand that and can help by pushing the obstacle out of the way.

Our use case is of a "monitor" agent that, being aware that the two robots have goals they should navigate towards, keeps track of relevant entities and relationships from the environment.

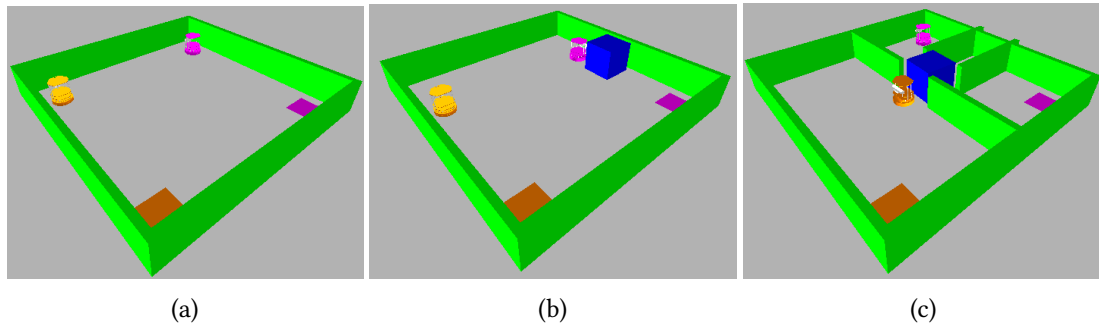


Figure 2: Two Turtlebot2 mobile robots navigating towards their respective goals. The goal for each robot is denoted by their respective colours (a). An obstacle blocking the path of the purple robot reaching its goal directly (b). Given the free space the robot can move around of the obstacle. In (c) the path for the purple robot is blocked by a movable obstacle. The orange robot can help by pushing the obstacle out of the way.

It does so by performing inferences based on spatial relations as reported by perception, the current knowledge of image schematic situations, the action-selection ontological query system and image schematic reasoning. That is, at every iteration, perception is instructed to monitor whether the robots are trying to move, and actually moving towards their goal. However, more complex inferences and queries to the perception system become available when needed.

A query that the monitor agent will make of perception is whether there are objects in front of a robot. If an object is in front of the robot, a further query is asked – is this object in contact? A situation of a robot that attempts to move, but does not, with a contacting object in front of it is one of blockage, and in such a situation the neighborhood of the blocking object also becomes relevant – in particular, what’s “in front” of the blocking object and might prevent it from being pushed aside. In short, the monitoring agent expands its attention as the situation becomes more complex by adding more objects to the list of objects it tracks, in a controlled and motivated manner.

The opposite – removing objects from attention – is also done once the reasons for which an object was deemed relevant cease to hold. For example, once the box in figure 2b is no longer in front of the robot, it is no longer an object about which perception queries are asked.

6. Conclusion

This work presented a method for robots to understand and reason about the current situation evolving around them. Such an understanding is important, as it allows focusing attention and reacting to situations correctly, as well as coping with cases in the environment that would cause classical task, path, and trajectory planning approaches to fail. The robots use concepts from situation awareness theory, along with image schemas and the logic developed around them. To evaluate the outcomes of the proposed approach, a set of simulated scenarios were used, which showed the robots were successful in perceiving the correct information from their environment, and using it to reason about the problem in each situation. Furthermore, the realized system and the heuristics applied in the action-selection loop constitute a robotic image-schema detector,

which performs cognitive reasoning based on spatial information, detecting image-schematic situation from knowledge about the physical static/dynamic *status quo* of the environment.

An immediate future expansion will be focused on increasing the robot autonomy by integrating the current approach with a combination of TAMP in more complex constrained-based manipulation problems that require a sophisticated reasoning mechanism to allow the robot to adapt its plan in both symbolic and geometric levels to the current situation. Moreover, another direction of investigation could be automatic generation of the ontology used by the image schematic reasoning layer (ISRL), i.e. the automatic creation of situation update and perception/action selection rules by combining knowledge from higher-level, more expressive formalisms. A concrete application of this is the analysis of *unsafe actions*, where “unsafe” is understood through a simple heuristic: an irreversible action is potentially unsafe. Deciding about safety then requires accounting for the laws of the environment (physical constraints) and the structure of the robotic agent (agent’s limited capabilities), and may be formalized as a series of queries to a PDDL planner to detect irreversible actions. These queries would be offline, and their results used to create the ontology used by ISRL for situated, fast reasoning. As an example of where this could be useful, the awareness of not being able to restore a SUPPORT situation to its original configuration could set a threshold to the amount of “safety” an agent would pursue, and make it avoid toppling over items. Finally, given the above directions, one could start investigating the possibilities of automated reasoning regarding the causality of phenomena, especially related to the actions of robots, as well as, the explainability of the actions chosen by the robot.

References

- [1] P. E. Hart, N. J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE transactions on Systems Science and Cybernetics* 4 (1968) 100–107.
- [2] A. Stentz, Optimal and efficient path planning for unknown and dynamic environments, *INTERNATIONAL JOURNAL OF ROBOTICS AND AUTOMATION* 10 (1993) 89–100.
- [3] J. J. Kuffner, S. M. LaValle, Rrt-connect: An efficient approach to single-query path planning, in: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, IEEE, 2000, pp. 995–1001.
- [4] C. Rösmann, F. Hoffmann, T. Bertram, Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control, in: *2015 european control conference (ECC)*, IEEE, 2015, pp. 3352–3357.
- [5] M. Ghallab, D. Nau, P. Traverso, *Automated Planning: theory and practice*, Elsevier, 2004.
- [6] Z. Kootbally, C. Schlenoff, C. Lawler, T. Kramer, S. Gupta, Towards robust assembly with knowledge representation for the planning domain definition language (PDDL), *Robot. Comput.-Integr. Manuf.* 33 (2015) 42–55.
- [7] R. A. Brooks, Elephants don’t play chess, *Robotics and Autonomous Systems* 6 (1990) 3–15.

- [8] M. M. Hedblom, *Image Schemas and Concept Invention: Cognitive, Logical, and Linguistic Investigations*, Cognitive Technologies, Springer Computer Science, 2020.
- [9] M. Johnson, *The Body in the Mind Metaphors*, University of Chicago Press, 1987.
- [10] G. Metta, G. Sandini, D. Vernon, L. Natale, F. Nori, The icub humanoid robot: an open platform for research in embodied cognition, in: *Proceedings of the 8th workshop on performance metrics for intelligent systems*, 2008, pp. 50–56.
- [11] L. Shapiro, *Embodied Cognition*, New problems of philosophy, Routledge, London and New York, 2011.
- [12] J. M. Mandler, How to build a baby: II. Conceptual primitives, *Psychological review* 99 (1992) 587–604.
- [13] G. Lakoff, M. Johnson, *Metaphors we live by*, University of Chicago press, 1980.
- [14] M. M. Hedblom, O. Kutz, R. Peñaloza, G. Guizzardi, Image schema combinations and complex events, *KI-Künstliche Intelligenz* 33 (2019) 279–291.
- [15] M. Pomarlan, J. A. Bateman, Embodied functional relations: A formal account combining abstract logical theory with grounding in simulation, in: *Formal Ontology in Information Systems: Proceedings of the 11th International Conference (FOIS 2020)*, volume 330, IOS Press, 2020, p. 155.
- [16] M. Pomarlan, M. M. Hedblom, R. Porzel, Panta rhei: Curiosity-driven exploration to learn the image-schematic affordances of pouring liquids, in: *Proceedings of the 29th Irish Conference on Artificial Intelligence and Cognitive Science*, Dublin, Ireland, 2021.
- [17] K. Dhanabalachandran, V. Hassouna, M. M. Hedblom, M. Küempel, N. Leusmann, M. Beetz, Cutting events: Towards autonomous plan adaption by robotic agents through image-schematic event segmentation, in: *Proceedings of the 11th on Knowledge Capture Conference*, 2021, pp. 25–32.
- [18] W. Kuhn, A. U. Frank, A formalization of metaphors and image-schemas in user interfaces, in: *Cognitive and linguistic aspects of geographic space*, Springer, 1991, pp. 419–434.
- [19] S. De Giorgis, A. Gangemi, D. Gromann, Imageschemanet: Formalizing embodied commonsense knowledge providing an image-schematic layer to framester, *Semantic Web Journal* forthcoming (2022).
- [20] M. M. Hedblom, D. Gromann, O. Kutz, In, Out and Through: Formalising some dynamic aspects of the image schema Containment, in: *SAC '18: Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, Pau, France, 2018, pp. 918–925.
- [21] J. M. Mandler, How to build a baby: Ii. conceptual primitives., *Psychological review* 99 (1992) 587.
- [22] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, D. Wilkins, *Pddl-the planning domain definition language* (1998).
- [23] T. Pan, A. M. Wells, R. Shome, L. E. Kavraki, A general task and motion planning framework for multiple manipulators, in: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 3168–3174. doi:PATH10.1109/IROS51168.2021.9636119.
- [24] M. Diab, A. Akbari, M. Ud Din, J. Rosell, Pmk—a knowledge processing framework for autonomous robotics perception and manipulation, *Sensors* 19 (2019). URL: <https://www.mdpi.com/1424-8220/19/5/1166>. doi:PATH10.3390/s19051166.
- [25] T. Migimatsu, J. Bohg, Object-centric task and motion planning in dynamic environments, *IEEE Robotics and Automation Letters* 5 (2020) 844–851.

- [26] M. Diab, M. Pomarlan, D. Beßler, A. Akbari, J. Rosell, J. Bateman, M. Beetz, Skillman – a skill-based robotic manipulation framework based on perception and reasoning, *Robotics and Autonomous Systems* 134 (2020) 103653.
- [27] P. Englert, M. Toussaint, Learning manipulation skills from a single demonstration, *The International Journal of Robotics Research* 37 (2018) 137–154. URL: <https://doi.org/10.1177/0278364917743795>. doi:PATH10.1177/0278364917743795.
- [28] M. R. Endsley, Toward a theory of situation awareness in dynamic systems, in: *Situational awareness*, Routledge, 2017, pp. 9–42.
- [29] C. J. Fillmore, Frame semantics, in: *Linguistics in the Morning Calm*, Seoul: Hanshin, 1982, pp. 111–138.
- [30] C. F. Baker, C. J. Fillmore, J. B. Lowe, The berkeley framenet project, in: *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, Association for Computational Linguistics, 1998, pp. 86–90.
- [31] A. Gangemi, P. Mika, Understanding the semantic web through descriptions and situations, in: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, Springer, 2003, pp. 689–706.
- [32] D. A. Randell, Z. Cui, A. G. Cohn, A spatial logic based on regions and connection, in: *Proc. 3rd Int. Conf. on knowledge representation and reasoning*, 1992.
- [33] N. V. D. Weghe, A. G. Cohn, G. D. Tré, P. D. Maeyer, A qualitative trajectory calculus as a basis for representing moving objects in geographical information systems, *Control and cybernetics* 35 (2006) 97–119.
- [34] M. M. Hedblom, O. Kutz, T. Mossakowski, F. Neuhaus, Between contact and support: Introducing a logic for image schemas and directed movement, in: F. Esposito, R. Basili, S. Ferilli, F. A. Lisi (Eds.), *AI*IA 2017: Advances in Artificial Intelligence*, 2017, pp. 256–268.
- [35] H. P. Lam, On the derivability of defeasible logic, 2012.
- [36] J. A. Bateman, Gum: The generalized upper model, *Applied Ontology* (2021). doi:PATH10.3233/AO-210258.
- [37] E. Coumans, Y. Bai, Pybullet, a python module for physics simulation for games, robotics and machine learning, <http://pybullet.org>, 2016–2022.