Development psychology inspired models for physical and social reasoning in human-robot joint action

Yoan Sallami

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Yoan SALLAMI

Le 29 janvier 2021

Modèles inspirés de la psychologie du développement pour le raisonnement physique et social dans le cadre de l'action jointe humain-robot

---

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par
Rachid ALAMI

Jury

Mme Adriana TAPUS, Rapporteure
M. François CHARPILLET, Rapporteur
M. Simon LACROIX, Examinateur
Mme Aurelie CLODIC, Examinatrice
M. Séverin LEMAIGNAN, Examinateur
M. Rachid ALAMI, Directeur de thèse

Acknowledgments

Many thanks to my mother and stepfather for being supportive during these four years of hard work, uncertainty and moments of joy. A particular thought for Erika that supported me every day. Many thanks to Séverin Lemaignan, who helped to envision a completely novel work during this thesis and to offer me the opportunity to work on an international collaboration. Finally, many thanks to my supervisor Rachid Alami for being always up to intense scientific discussions.

Contents

Introduction

In the last years, robots have started to be present in real-world scenarios, from autonomous cars to automated supply chains. However, human-robot collaboration is still not solved, and despite extensive research in this field, it is hard to make it work outside laboratories. This work is part of a broader effort to build an interactive and cognitive architecture for collaborative robots ([Lemaignan 2017]).

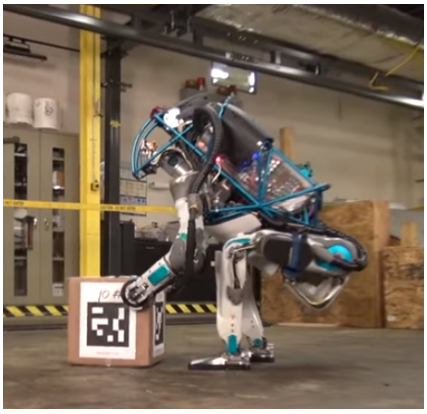


Figure 1: Robot from Boston Dynamics trying to pick up a box in a warehouse.

Figure 1 is from a popular and controversial video from Boston Dynamics, where a robot was trying to pick up a box while a person hit him with a stick to show the robustness of the control algorithm. Ironically they were not collaborating at all. We can easily imagine that this robot could naturally work with humans if provided with the right software.

The current research in robotics has made enormous progress in control or computer vision in the last years but still lacks higher cognitive reasoning. This thesis tried to understand the underlying principles that make collaboration possible at an interpretation level with a developmental psychologist's perspective. After defining the needs and concepts used in human-human collaboration, we present the current view of these challenges in robotics and AI.

This thesis focuses essentially on the *geometric and temporal situation-assessment* layer. The role of this component layer is to gather perception data (identified objects, location, and shape) along with robot proprioception and localization to build up a geometric model of the world representing the robot and its surrounding with symbolic knowledge (first-order predicates) on top of it ([Sisbot 2011a], [Milliez 2014]). This world model needs to be updated in real-time to allow higher-level components (supervision and planning) to reactively execute and plan actions in a human-robot joint action context.

Understanding the robot's surroundings is a long-term goal in the robotic community, and it is one of the hardest ones. Especially when, like in human-robot

collaboration, the robot needs to reason about the objects and the human partner it interacts.

In the context of human-robot joint action, the robot needs not only to understand the environment but also to be able to communicate its understanding to the people it collaborates with, which is more difficult because it needs to build a representation of its environment that makes sense for humans or to have the capability to map its representation to the human one.

Scope of this thesis

Considering the progress of deep learning in computer vision, natural language processing, or intuitive physics, studying how we could benefit from them and where the limits are is natural. This thesis intends to study machine-learning algorithms to understand their limits and advantages to eventually use them to design efficient models inspired by developmental psychology. Numerous authors state that neural networks are the way to go in order to build human-like intelligence. In [Lake 2017] the authors emphasize:

"We also believe that future generations of neural networks will look very different from the current state-of-the-art neural networks. They may be endowed with intuitive physics, theory of mind, causal reasoning, and other capacities"

In their demonstration, they suggest core ingredients to build human-like intelligence by taking inspiration from developmental studies. First, they emphasize that machines should build causal models that support explanation, and then they introduce a set of ingredients needed that are present in infant's early development: *intuitive physics* and *intuitive psychology*. During this thesis, we focused on designing models for these two last domains.

This thesis presents a hybrid architecture that combines machine learning, physical reasoning, and a knowledge-based system for situation assessment. This system integrates a continuous physical reasoning system based on a physics engine with visual perspective-taking capabilities required to interact with people naturally.

In our view, hybrid architectures are the way to go, in the sense that one multipurpose neural network is often impractical in order to cover all spectrum of reasoning processes needed for robots.

Research questions

During the first years of this thesis, we started to study the processes needed to endow robots with the reasoning needed to collaborate. For that matter, we looked at developmental psychology studies. Indeed, being able to perceive the environment is not enough in order to exhibit intelligent behaviour. Based on the findings on human cognition and recent studies on physical intuition, we developed a monitoring algorithm that uses a physics engine to anchor perception into a physically plausible world model. We then conclude on how this system could integrate with

neural-based physics intuition in the future. In complement to this work, we also developed visual perspective-taking capabilities that were enhanced by physics reasoning and allow to compute correct spatial relations between entities. To show the pertinence of such reasonings, we integrated this model with a formal knowledge-based system that uses ontologies to describe the world's semantic. As a final step, we propose a methodology for building a dataset to study and serve as a benchmark for reasoning algorithms in future helping scenarios.

During this process, many questions emerged. Here are presented in a nutshell the research questions that we aim to provide answers to in this work:

- What are the core capabilities we need to endow a robot to perform situation assessment in a human-robot joint action context?
- What are the mechanisms and models needed to perceive the environment and reason on what is not visible?
- Is it possible to use simulation-based physical reasoning at runtime to reason finely about object persistence and physical inconsistencies?
- How can these mechanisms can be integrated with a deep learning approach for intuitive physics?
- How to bind natural language models and knowledge-based systems for questions answering in the context of an embodied agent?
- How to go further in the study of human-robot collaboration, and what are the next steps?

This work has been done in the context of a European project H2020: The MuMMER project. During this project, we also started an international collaboration with the Bristol Robotics Laboratory (BRL). In the following sections, we present more in detail this project and collaboration context.

The MuMMER Project

This thesis was funded by a European project H2020: MuMMER (MultiModal Mall Entertainment Robot). This project was in collaboration with different European partners and SoftBank robotics. The aim was to deploy the robot (see Fig. 2) in a shopping centre in Finland. The robot guided people in the mall by using the mall's geometric and semantic model. By computing what is visible or not, the robot can place itself correctly to share a common perspective and explain the route to the customers (see chapter 4)

Figure 2: The Pepper robot in a shopping mall in Finland surrounded by the people involved in the MuMMER project.

The collaboration with the BRL

After the first year of this thesis, we started, in parallel with the MuMMER project, a collaboration with Séverin Lemaignan about the software that he started to develop (see chapter 3). We refined this software's concepts and principles for four years while always considering the big picture: building a framework for robot situation assessment.

In the last years, we started together to build a dataset that will emphasize the problematics of HRI while providing a source of data in which researchers could train and test their contribution. I spent two months in Bristol (United Kingdom) to start the recordings and prepare the data for that purpose. This effort is, in our view, critical because most of the robotics research is not replicable. By creating robotics benchmarks datasets that fit our purpose like in the computer vision community, we hope to contribute to future progress in the field of human-robot collaboration (see chapter 8).

Contributions

In the first chapters, we present our two main research topics and conclude the current needs regarding investigation and challenges in human-robot joint action.

We then introduce an open-source framework fully compatible with ROS that uses a data structure adapted for physical, geometric, and semantic reasoning. We want to develop a unified representation of the robot's and other people's spatial

and temporal knowledge with this framework. We present a real-world application *in the wild* of this framework in the MuMMER project (see chapter 4).

Then we present the first integration of a physics engine for object permanence that runs parallel with the perception system in a cognitive architecture for human-robot joint action. We conclude this chapter with a discussion about the future integration of neural-based physics engines.

We also show our approach's modularity by binding this perceptual and physical reasoning system with ontology-based reasoners. We show the pertinence of the interaction between low-level and high-level reasoning with preliminary work on integrating neural-based language models.

In a nutshell, the contributions of this thesis can summarize as follow:

- We introduce a design and implementation of an architecture for situation assessment called UNDERWORLDS.
- We provide a reasoning stack for visual, physical, semantic, and belief reasoning in the context of human-robot joint action.
- We introduce the concept of *physical monitoring* that uses a physics engine at runtime to provide the physical understanding and detect tabletop actions. We show the method's effectiveness with challenging examples of tabletop interactions where part of the scene is occluded.
- We provide the binding of this new system with ontology-based reasoners and show the pertinence of such synergy by studying the integration of neural language models with that system.
- We finally introduce an ongoing work on creating a new dataset to study incongruent events and helping behaviour in an everyday scenario called: *Unexpected Daily Situations* (UDS) dataset.

Overview of the thesis

In chapter 1, we present an overview of simulation-based physical reasoning over literature. We will first explore this reasoning in infants through developmental psychology to then move on to robotics and AI applications of such concepts.

In chapter 2, we extend this state of the art with belief reasoning in infants and its interpretation with robots, emphasizing visual perspective-taking and Theory of Mind processes.

After presenting state of the art, we introduce in chapter 3 an implemented open-source framework that intends to help HRI researchers build physical, geometric, and semantic reasoning for human-robot collaboration. We provide some examples and a real-use case with the European project MuMMER in chapter 4.

We then present in detail in chapter 5 the main contribution of this thesis: a working implementation of simulation-based physical reasoning in HRI. This reasoning allows the robot to infer the object's positions beyond the perceptual horizon, correct object poses, and detect object-based actions (pick, place, release).

We then present in chapter 6 the implementation of a perception stack compatible with the overall framework. This work has been done to show our approach's modularity and develop reasoning algorithms that benefit from recent progress in deep learning and, in particular, the deep representations for visual entities.

To show the pertinence of the data structure chosen in the framework developed and its modularity. We present the integration of this work with ontology-based reasoners in chapter 7 and provide an application for grounding verbal expressions with deep language models combined with a knowledge base.

Finally, to help researchers in HRI build more replicable research and gather data for the machine learning algorithms, we present in chapter 8 our progress in building a dataset inspired by Werneken and Tomasello's works ([Warneken 2006]) on altruistic behaviour with naturally collaborative situations (e.g. where collaboration naturally emerges).

We finish in with a general conclusion and future directions for this work.

Chapter 1

State of the art: Physical reasoning

Contents

1.1 Physical reasoning in infants

This section will present developmental psychology (the study of how and why people acquire cognitive abilities over their life) theories related to physical cognition in children. Two major trends in the study of developmental psychology have emerged over the years:

- Cognitive developmental psychology (Piaget, Baillargeon)
- Evolutionary developmental psychology (Spekle)

The first one states that the human mind is a multi-purpose learning machine that learns capabilities by acting and perceiving the world. The second one says that evolution has shaped core systems that allow us to learn and reason by interacting

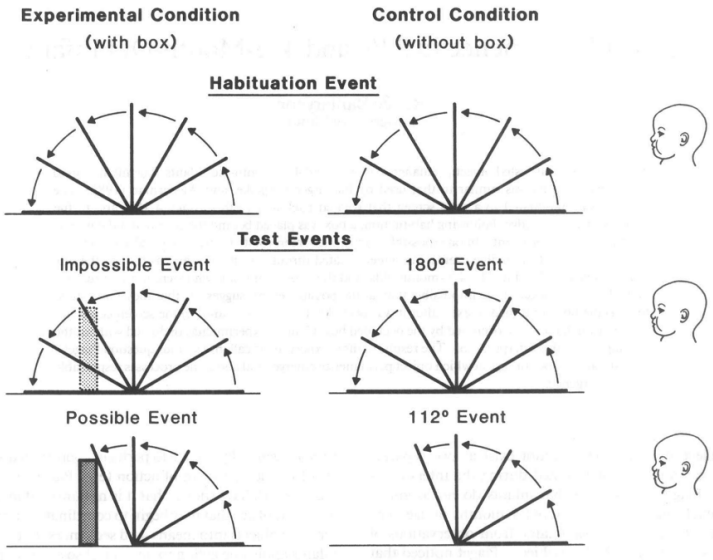


Figure 1.1: The screen experiment from [Baillargeon 1987]. First, they present the habituation event, then one of the two conditions: the possible event or impossible event. The infants looked reliably longer at the impossible event suggesting that they understand that objects can not overlap

.

with the world. The debate here is on the origin of human's cognitive capability, and the study of animals and primitive cultures yielded exciting results in favour of a core knowledge ([Dehaene 2006]). *Here, we do not aim at providing an answer to that question.* However, nowadays, even cognitive developmental psychologists like Baillargeon argue that we probably have innate principles.

1.1.1 Object permanence

Jean Piaget, one of the most influential psychologists of the 20th century, was the first to study object permanence in infants ([Piaget 1952],[Piaget 1954]). Object permanence is the understanding that an object continues to exist even when not perceived. He started to study it by observing how infant reacted when they favourite toy was hidden. If the infant would reach the hidden object, then he stated that they had object permanence. Consequently, Piaget proposed that infants under 8 or 9 months old lack of object permanence.

Since then, Piaget's conclusions revealed to be baised because his work involved actions that require sensory-motor capability that young infants do not have. Using violation-of-expectation (VOE) looking time methodology a method that exploit the tendency that infants naturally look longer at events that violate their expectations (see Figure 1.1 and Figure 1.2), numerous authors have demonstrated that 2.5 to 5 month old infant have already object permanence ([Spelke 1992], [Newcombe 1999], [Wilcox 1996]). They are able to reason about hidden mov-

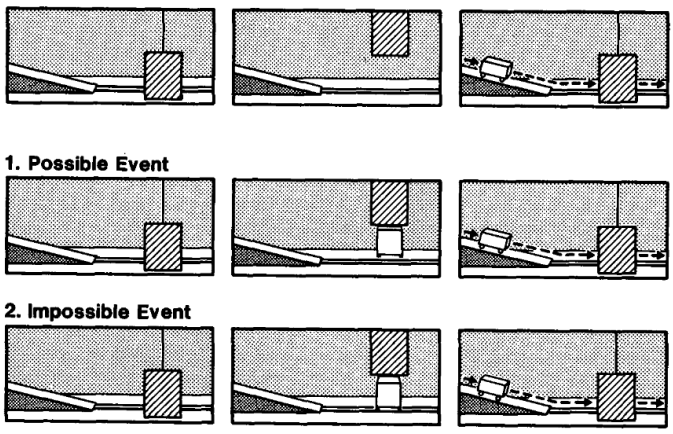


Figure 1.2: The experiment with a toy car and a box from [Baillargeon 1994]. First, they present the habituation event where the car follows a trajectory while a screen hides part of the course. For the test events, a box is placed outside the car's path for the possible event and on the path for the impossible event (the car should have been stopped).

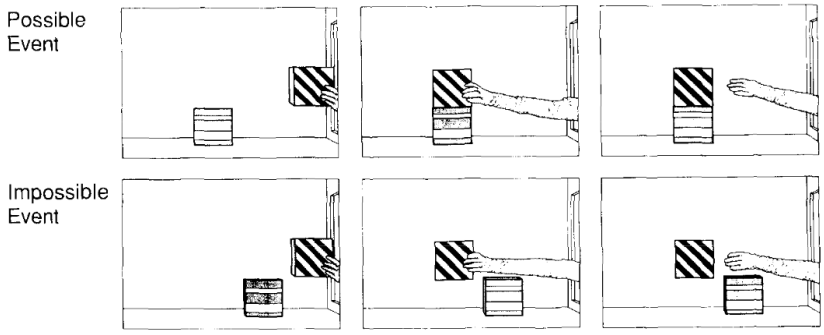


Figure 1.3: The support experiment from [Needham 1993]. This experiment shows that infants have reasoning capabilities about gravity. In this experiment, an object is placed in front of the participant. For the possible event, the object lies on support, and for the impossible one, it stays in place without any support (the object is flying in the air).

ing or stationary objects ([Baillargeon 1994]), containment ([Hespos 2001]) or support ([Needham 1993]).

Some authors ([Bogartz 1997], [Cashon 2000]) suggested that the violation of expectation was due to the habituation trial that introduces a preference for the new event (the impossible one). Recently ([Wang 2004]) empirically demonstrated that even with no habituation trials, infants still looked significantly longer at the impossible events.

These results strongly suggest that very young infants can represent and reason about hidden objects as well as gravity and solidity.

1.1.2 Core knowledge

The results presented above are fundamental in development psychology and emphasise that infants have at a very young age object permanence and thus the basis of physical reasoning (see section 1.1.3 for the definition).

Then arise the question of the origin of such capabilities. An influential theory ([Spelke 1992]) from evolutionary developmental psychology states that infants are born with a framework of core systems that are domain-dependent and allow them to learn and reason about physical events: The core knowledge.

Since then, numerous authors suggested that infant are born with core principles ([Wellman 1992], [Leslie 1993], [Spelke 1995], [Carey 2000],[Baillargeon 2012]), depending on the authors the principles description lightly differ and/or have changed over the last years as few findings come.

The following core principles are based on the infant's physical reasoning as presented in [Lin 2020][1]:

Persistence

This principle states that things persist in time and space, making it impossible for objects to occupy another object (solidity). They cannot disappear (continuity) or break apart without assistance (cohesion).

Inertia

The inertia principle states that objects at rest will stay stationary, and objects in motion will follow a smooth path without abrupt changes unless they interact with another thing.

Gravity

The gravity principle states that objects fall when not supported, consistent with the support experiment, where infants look longer at the inert object suspended in the air (see Figure 1.7).

[1] note that Renée Baillargeon is co-author of this article

1.1.3 Two-model system: Object File and Physical Reasoning

It is commonly accepted among psychologists that infants physical reasoning is built around two different subsystems:

The physical reasoning system is "an abstract, computational system that provides a skeletal causal framework for making sense of the displacements and interactions of objects and other physical entities" ([Baillargeon 2002])

The object file system is "the main end product of perceptual processing of a stationary scene is a set of object files, each containing information about a particular object in the scene" ([Kahneman 1992])

1.1.4 Explaination based learning in infant

Infants also have core concepts about what is unobservable to explain unexpected events. As shown by ([Luo 2005], [Baillargeon 2009]) if we give sufficient evidence that an object is animated, young infants assume that it has its own "internal energy" (thus is animated) and can consequently control its motion or resist to external forces.

This emphasises that infants incrementally refine their physical reasoning to match unexpected situations with their own mental representation.

In [Baillargeon 2017] and [Baillargeon 2010] the authors argue that this refinement is process is made through three main steps that allow infants to learn new rules about the physical world:

Triggering

The first is the triggering part. When a situation does not match the expected outcome (relative to their knowledge), the explanation based learning is triggered.

Explaination construction and generalization

Then infants search to identify the features that support a given outcome. If they succeed in identifying such features, then that rule is added to their knowledge, and they use it in conjunction with the core principles to predict the outcome of future events.

Empirical confirmation

After a rule has been identified, if further evidence are acquired to support that new rule, it is added to their physical knowledge. Otherwise, the rule is rejected.

According to the authors, that process explains why infants need only a few exemplars to learn a new rule by either generalising the rule at a feature level and confirming the acquired rule through empirical confirmation.

1.2 Physical reasoning in AI and robotics

The fact that we now understand better how these processes emerge in infants (and the importance of endowing robots/AI of such capabilities) and the progress of computer vision and physics engine lead to interesting approaches in this area. We voluntarily focus on simulation-based physical reasoning that enables to reason at a geometric and object level.

Before being able to reason on physics, the robot needs first to detect, track, and recognise the scene's different elements: the people and objects it will interact with. For that purpose, computer vision has recently made enormous progress thanks to deep learning. However, many challenges remain or do not have the robustness needed to be used in robotics. Note also that because we are working on robots that need multiple and diverse real-time inferences, we need to perceive the objects, the humans, and reason about them, making the real-time constraint stronger.

1.2.1 Convolutional neural networks

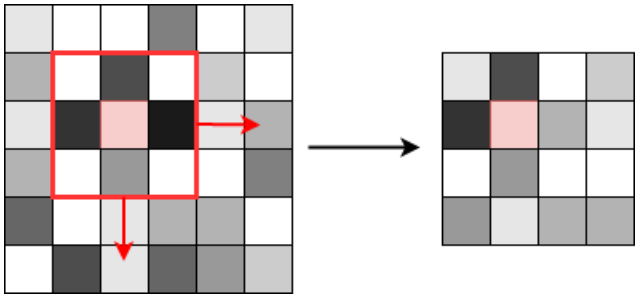


Figure 1.4: By sliding from left to right and top to bottom a kernel along with an image, the convolution operation produce a matrix that represents the activation of the pattern contained in the kernel. The red cell in the left matrix corresponds to the centre of the filter that is used to compute the corresponding red cell on the right.

Convolutional neural networks CNN are special neural networks used in pattern recognition that are maint to learn the kernels (or also called filters) used in a convolution operation (see Figure 1.4). By stacking the convolutional layers, CNN can learn different granularity of features, from low-level to high-level, to accomplish the classification task (see Fig 1.5).

Note that CNN can also be used for 1D data. In natural language processing, it allows to speed up the inference/training by reducing the dimensionality of recurrent neural networks like in [Gehring 2016].

1.2.2 Recurrent neural networks

Recurrent neural networks (RNN) are networks used when modelling sequences of data, their particularity is that they integrate gates and an explicit memory cell.

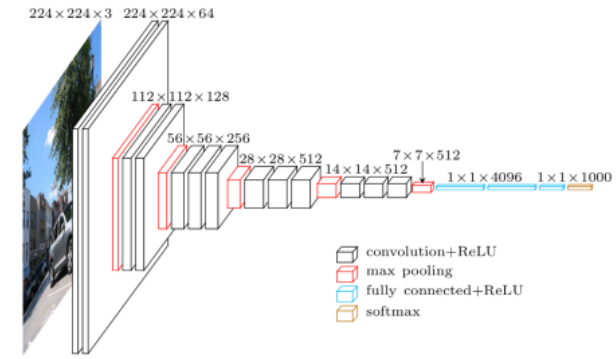


Figure 1.5: VGG16 architecture. By stacking convolutional layers, CNN can represent hierarchical features. Additionally, pooling layers reduce dimensionality by keeping the important characteristic (here the maximum). Then the last layers are responsible for outputting the probability of the classes by using a fully connected and softmax layer. This last part is common to every classification problem and is not specific to the CNN.

Depending on the complexity of the task, the RNN can be a LSTM ([Gers 1999]) or a GRU ([Cho 2014]) which is a simpler version of the first one. Used intensively in natural language processing in conjunction with word embeddings, a technique that allows encoding words into a geometric space of fixed dimension (lower than the number of words in the vocabulary) where distances represent similar meaning ([Mikolov 2013], [Pennington 2014], [Devlin 2018], [Peters 2018]).

1.2.3 One-shot object recognition

One-shot object recognition is a type of task where the aim is to learn new classification labels on the go by just providing one example to the system.

One of the successful method to perform such task is by using Siamese Neural Networks (SNN) ([Koch 2015], [Schroff 2015]). Instead of classifying an image using a softmax prediction layer like in a classical classification problem. SNN learn an embedding vector that maximises a distance metric for pairs of images that do not belong to the same category or minimise it if they belong to the same class using a contrastive loss ([Hadsell 2006]). To do so, two identical networks are used to encode the data into a vector of fixed size using a fully connected layer with sigmoid activation at the end of a CNN (for the case of image classification). The two vectors are then used as input of a metric distance that output the probability of belonging to the same class.

That way, it is possible to learn an embedding vector that can be then used in classical K-nearest neighbours (KNN) or clustering algorithms. At inference time, only one of the two siamese networks is used, and the last layers (the one with the distance metric) are removed. That way, the network infer the representation vector to be clustered or used in a KNN.

By choosing a threshold for new objects, if the observation is too far from what has been memorised in the KNN, the object is considered new, and the robot can start a procedure to discover it. For example, when the robot sees a new item, it can ask what kind of object it is and retrieve the object name/class when it sees it again without any training. However, it would be cleaver for the robot to still record the object appearance under different angles for later training in order to robustify this recognition. This capability is a key aspect of human-robot interaction because the robot can ask the person it interacts with to enhance its knowledge about the world.

This technique's limitation is that it can operate only on problems that do not have high dimensionality. One workaround for big images is to use a pre-trained backbone[2] on ImageNet ([Deng 2009]) that are frozen, the weights are not updated during backpropagation, without the last classification layer to reduce the dimension of the input and help not to overfit. This technique, is part of what is called transfer learning, a set of techniques to transfer knowledge learned from one domain to another. This is possible because only the last layers of a CNN are specialised due to CNN's hierarchical representation.

The most common use of this technique is facial recognition ([Schroff 2015], [Parkhi 2015], [Song 2019]) but SNN works not only on images. If one can encode any input data to a relatively small dimension, it can be used for any data type or sequence. For example, it has been used for text similarity tasks in [Reimers 2019].

The training procedure consists in generating negative and positive pairs of images (or any data) that are difficult to classify and force the network to learn meaningful representation.

This technique is popular among the robotics community because learning a distance metric allows them to apply K-nearest neighbours or clustering algorithms to features that are highly non-linear (like the ones extracted from deep learning).

Note that this capability can be generalised to every object feature based on appearance (colours, affordances, materials, shapes), and the potential applications of that technique are broader than just object recognition. It can be useful in many similarity-based problems, explaining its success.

1.2.4 Object and person detection

Before being able to reason on physics, the robot needs to detect the objects and the people present in its surroundings. For that purpose all state-of-the-art object detection ([Erhan 2014], [Girshick 2015], [Bolya 2019]) use convolutional neural networks (CNN).

Nowaday, two types of detectors are generally used:

- Two stage detector ([Erhan 2014], [He 2017])
- One-stage detector ([Redmon 2016], [Redmon 2018], [Liu 2016])

[2] an architecture for image classification that is known to perform well like VGG16 or ResNet50

The differences between these two types reside in the formulation of the problem. Two stages detectors first find region proposals and then predict the output class and bounding box. In contrast, one-stage detectors aim at predicting the output class and corresponding bounding boxes in one step by predicting anchors probability, which is a set of prior bounding boxes for each class that is computed offline. They are consequently more efficient and can run in real-time on CPU.

These techniques have been widely used in robotics. However, they need to be trained, especially for object detection and recognition, as the datasets available cannot reflect every object's diversity. On the contrary, for persons, face, or human pose detection, the variety of datasets that are available online allow using pre-trained weights.

1.2.5 Object tracking

In classical computer vision task, detection and tracking are two separate processes. However, the term tracking is misleading as it can be two different problems:

- Single object tracking
- Multi object tracking

While single object tracking aims at finding an already detected object over the next frames, multi-object tracking also has to assign the correct ID to the detected objects and eventually use a single object tracker to robustify the detections or speed up the perception pipeline by detecting at a lower frequency.

In our use case, we need multi-object tracking (MOT) to keep the ID of the persons and objects the robot interacts with. Most states of the art multi-object tracker (see MOT challenge[3]) are based on the tracking-by-detection paradigm, where the success is based on the accuracy of the detections(two stages detectors usually). In tracking-by-detection, the tracker is responsible for assigning the ID using Hungarian algorithm ([Kuhn 1955]) to solve assignment problems in polynomial time, with geometric and eventually deep features, predict the motion of the object using Kalman models, and manage the occlusions (re-identify the objects/persons after a short occlusion) using appearance features assignment. This basic methodology is the one adopted by SORT and DeepSORT ([Bewley 2016], [Wojke 2017]) and is the core of almost every MOT algorithm. What makes the multi objects trackers different today is the reliability of the deep feature extracted, allowing them to assign the tracks better. The current best tracker of the 2020 MOT challenge [Karthik 2020] use an unsupervised learning method by first using SORT to build a dataset of noisy tracks to train a network that infers the ID of the track. That tracker is not directly usable in the real world as it is designed only for a defined number of tracks (the output prediction is a vector of the size of the number of tracks in the video), as they say, the best way would have been to use SORT to build a dataset of positive and negative pairs to train a SNN for recognition.

[3] https://motchallenge.net/

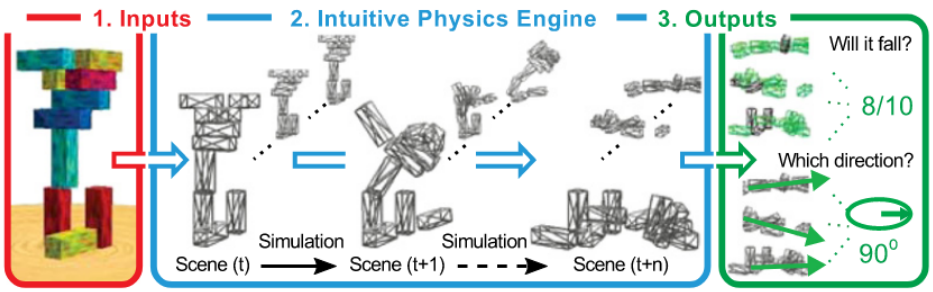


Figure 1.6: Adapted from [Battaglia 2013]. By running multiple times an analytic simulator and compairing with human judgement they conclude that human judgement match closely a probabilist physics engine.

1.2.6 Graph neural networks

Graph Neural Networks (GNN) is a recent technique that combines neural networks with a graph data structure that can be trained with gradient descent in a end-to-end fashion.

The particularity of the GNN is that they are relatively simple and exploit the relational structure of the data to support a wide variety of inferences (edge, node, or graph level). Graph neural networks are based on two small neural networks models called the *propagation model* and the *output model*([Scarselli 2008], [Wu 2020]) which reason respectively an a relation level and object level. Message Passing Neural Networks (MPNN) are a generalisation of GNN that can be run multiple times until converging.

These networks were widely used in a wide variety of tasks over the last years. The most remarkable use for our context of that technique is the neural physics engines ([Battaglia 2016], [Sanchez-Gonzalez 2020]) and semantic graph encoding([Xu 2016]). Semantic graph encoding refers to the operation that transforms a graph representing semantic relations into a fixed size dimension vector that can be integrated into a classification problem.

1.2.7 Physics engines and human judgment

In recent influential studies ([Hamrick 2011], [Battaglia 2013],[Bates 2015], [Ullman 2017]) several authors propose that humans physical scene understanding can be explained with probabilistic simulation engines using generative simulation model to predict the outcome of a variety of scene configurations, which is found to closely match human judgment (see Fig.1.6).

Very recently, researchers successfully explored probabilistic simulators based on graph neural networks (GNN) for rigid bodies or fluids decomposed in particles ([Battaglia 2016],[Ajay 2019],[Sanchez-Gonzalez 2020]). These networks aim at using simple neural networks over a classical graph-structured representation that allows them to generalise well on new configurations.

To simulate physical interactions with neural networks, they use interaction networks [Battaglia 2016], which are composed of two small neural networks and a graph data-structure:

- One neural network that predicts pairwise forces between objects represented as nodes.
- The other that predict the next state of an object based on the state of the objects it interacts with.

These simulators have the benefit of over-analytical simulators that they can learn the parameters of the simulation and be fed with real data to refine the simulator's model.

The limitation is that collision tests are still needed (like in a traditional physics engine) in order to sparsify the interaction graph (objects do not interact at a distance); otherwise, the complexity of the problem makes it not possible to use it in real-time. Collision tests are what consumes most of the classical physics engine resources, making not clear the potential benefit for real-time application.

Despite the promising results, it is unclear if the neural-based simulation's real-time performances are sufficient for continuous runtime inferences where classic analytical models are preferred.

1.2.8 Other use of simulation-based physics reasoning

In [Kunze 2017] they use the simulation to predict the near future of action and reason on the physical plausibility of actions. In [Mösenlechner 2009] they use a physics engine to predict the near future of objects. In [Mösenlechner 2011], they also use physics simulation to sample hypothetical states of the world based on a symbolic plan to parametrise high-level actions.

In [Weitnauer 2010], the authors evaluate simulation-based reasoning to predict the outcome of a robot arm pushing flat pieces on a table. They integrate real-world data by optimising the simulation parameters offline, using an evolutionary algorithm to fit the ground truth trajectory. In [Agrawal 2016], they learn how to poke by integrating the effects of the action into an intuitive physics but lack an object-level representation, which makes it not possible to generalise the poke to others actions.

In the same time, several researchers in AI have explored how to learn the parameters (forces, masses or trajectories) of a the physical world, from videos ([Wu 2015],[Zhu 2016],[Finn 2016]) or static images ([Mottaghi 2016], [Li 2016]).

1.3 Conclusions

In the first months, infants are already able to detect basic spatial and temporal information ([Leslie 1994]). When growing up, infants learn to represent better

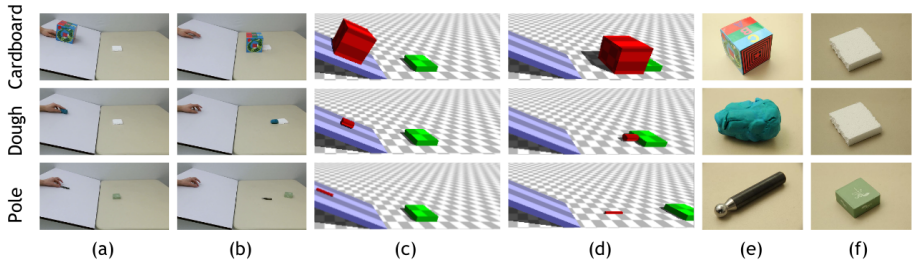


Figure 1.7: Experiment from [Wu 2015]. They train a small neural networks to learn the properties of diverse objects from static images (examples of object in e and f), then used a tracker and the inference from the neural network to parametrize a simulation engine (Bullet) used as generative model to predict the object position.

these events but still, 2.5-month-old infants (the younger possible infants tested with VOE tasks) already possess the continuity and solidity principles.

In general, robots have continuity principles through multi-object tracking of objects but do not analyse nor can detect solidity violations. This fact motivated us to study how we could integrate the solidity and gravity principles in robot's reasoning. Considering the research of [Battaglia 2013] about human physical intuition, it appear to us natural to use a physics engine for that purpose.

Interpreting visual information by relying on physical reasoning is a foundational cognitive ability that allows humans to anchor perception information into a consistent model of the world and reason about what is not visible based on collision and gravity.

Object-level representation is essential to human-robot interaction as it is how humans naturally explain the world. For that reason, we think that physical intuition needs this kind of model, making it not interesting for our context the physical intuition algorithms that do not contain an object-level representation.

Recently physical intuition had major advances in AI because of the formulation of GNN. However, this technology is not mature enough to replace classical physical simulators for rigid body simulation *at real-time*. Nevertheless, we do not doubt that it will be possible in the future to integrate these probabilistic physics engines as they integrate the same object-level representation that classic ones.

Physical simulators have a long history in robotics. However, they are more used in an offline manner to make the robot learn safely or to be used in the planning scheme. Very little work has been done to use a simulation engine for reasoning on what is not visible at runtime in a continuous fashion.

As far as we know, no work has been done to integrate a physics engine that runs in a continuous fashion to infer what is not visible. No work has been done either in human-robot setups where humans can interfere with physics (when picking an object, for example) at runtime. In many cases, the assumptions made that the world is static except when providing known stimuli like a robot action do not fit our context where objects can be also be moved by humans during collaborative

tasks.

Chapter 2

State of the art: Beliefs reasoning

Contents

This chapter presents findings in developmental psychology about belief reasoning and robotics and AI interpretations. In a nutshell, belief reasoning is the ability to interpret other's behaviour according to their attributed mental states (beliefs, goals, desires). It is the foundation of the intuitive psychological-reasoning system of infants.

2.1 Beliefs reasoning in infants

Reasoning about what we think others know (or not) is the core of human intuitive psychology. To study it in the context of human-robot collaboration, we first present an overview of the state of the art in developmental psychology in this area. We then investigate how humans infer what others perceive and later how beliefs are attributed. To do so, we take a closer look at visual perspective taking: the ability to imagine other's points of view.

2.1.1 Visual perspective taking

Like with physical reasoning, it starts with Piaget's work. In [Piaget 1956], Piaget studied at which age infants start to consider others viewpoints using a famous test: the three mountains. In this test, the children take a look at the scene and sit at

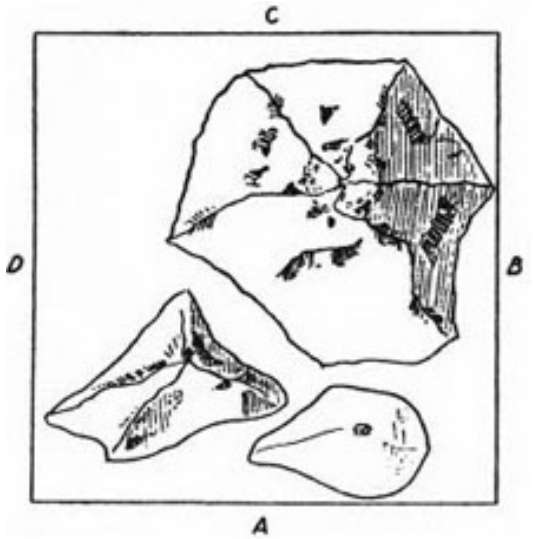


Figure 2.1: The three montains test setup from [Piaget 1956]. Depending on the position of the doll (A, B, C or D) the perspective of the scene is different. In that test, infants were asked to choose the correct perspective that correspond to the doll's view.

a table where three different mountains are presented: one with snow on top, one with a red cross, and another with a house (See Fig. 2.1).

A doll is then placed at different positions, and the experimenter asks the children which photograph is the doll's view among a set of different views. If the children answered correctly, then they stated that they were not egocentric or egocentric otherwise. His results showed that infants start around seven years old to succeed in taking into account the doll's point of view.

Since Piaget's work, this concept of egocentrism is not used anymore because it was not precisely defined, and nowadays, psychologists prefer to talk about visual/perceptual perspective taking (VPT) instead.

At its essence, it is the mechanism that allows people to take into account other viewpoints (used every day when we are talking about spatial references [Schober 1993] and more generally interacting with others).

In [Flavell 1981] and [Flavell 1977] the authors make a distinction between two levels of visual perspective-taking. Level one (VPT1) concerns that children start to understand what is visible from a different perspective, while level two (VPT2) starts when children understand how it is visible from another view.

In [Michelon 2006] the authors give evidence that two different processes are at play when determining the relative position to other viewpoints or when estimating if an agent can see an object, respectively, one that imagines another viewpoint (VPT2) and one that traces lines of sight (VPT1). To establish their conclusion, they asked the participants to tell if the objects were at the left or right of another agent. When the angular distance between the participant and the other

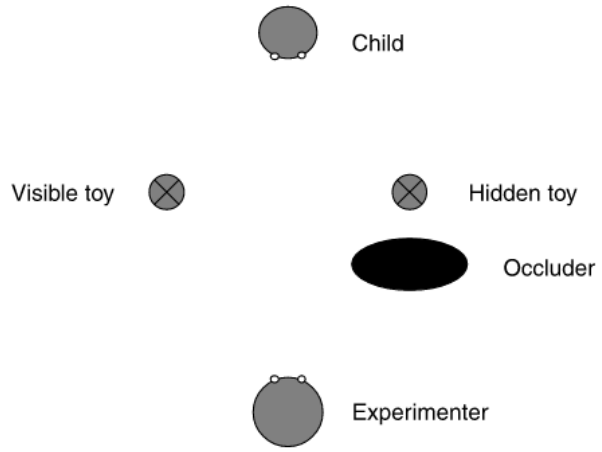


Figure 2.2: The experiment of [Moll 2006] where part of the scene is occluded from the actor view.

agent increased (thus, the perspective is more different and needs more resources to imagine), they noticed that the response time increased. On the contrary, they noticed that the response time was independent of the angle when asked if the agent could see the objects.

This classification (visual perspective-taking levels) is fundamental and still holds today among psychologists. In [Moll 2006], the authors argue that VPT1 taking is possible around two years old. The authors did an experiment where part of the scene is occluded (see Fig. 2.2). In this scenario, the actor fake to look for the toy and ask the child to indicate the toy that he is looking for. By studying if infants could help find the occluded toy correctly, they made their conclusions. The problem here is that it involves higher-level cognitive capabilities (verbal and sensory-motor) than just the attribution of false beliefs.

VOE tasks are preferred if one wants to study false-belief tasks in isolation from other cognitive abilities. Recently it has been demonstrated with the VOE task ([Onishi 2005]) that, in fact, 15 month-old infants possess an early ability to perform VPT1 and to attribute false-beliefs. This striking result makes even more important the place of such cognitive abilities in a child psychological intuition.

2.1.2 Theory of Minds

Theory of Minds (ToM) states that human attributes mental states (beliefs, desires, goals, or intentions) by inferring what others see, know, or memorize. In order to study this mechanism, psychologists use what is called a false-belief task. In that kind of task, an actor plays a scenario while perceiving only part of the scene. Questions are then asked, or more recently, only the participant gaze is analyzed to know if the child considers this different perspective.

Sally and Anne is a famous psychological test ([Wimmer 1983,

Baron-Cohen 1985, Wellman 1988]) that was used to test the capacity of children (and in particular children with autism) to understand others false beliefs (See Figure 2.3). It is based on the occlusion of an object (the marble) in a box (emphasizing that object persistence is, by the way, fundamental). This study suggests that only three or more years old infants were able to attribute false beliefs.

However, like Piaget's conclusions, this test's conclusions were inaccurate because it requires that the infant verbalize the correct answer. Recent findings using VOE (children look reliably longer when agents act inconsistently) or spontaneous responses revealed that younger infants (13-15 months old) were able to attribute false beliefs about the location of objects ([Onishi 2005], [Surian 2007],[Träuble 2010]). False perception of an object at 14.5 month old ([Song 2008]). False belief about object's identity at 18 month-old ([Scott 2009]). These new findings suggest that infants can represent other false beliefs even if they cannot efficiently answer the question: "Where Sally will look for her marble?".

2.1.3 Two-model system: SS1 and SS2

During the exploration of the mechanisms that allow infants to attribute to others mental states, psychologists ([Leslie 1994], [**?**]) have been elaborating cognitive models that account for the fact that the cognitive abilities needed to perform ToM rise at different ages, thus suggesting that they lie in different sub-modules.

Psychologists commonly accepted that Theory of Mind related processes could be explained with at least two sub-systems, respectively sub-system 1 (SS1) and sub-system 2 (SS2), here we briefly resume their roles from [**?**]:

SS1 is responsible for attributing the agent's motivations in the scene (using a rationality principle: agents acts according to their goals) and what agents can perceive, memorize, or infer. This information is used to block the information available to the agents using a masking mechanism allowing infants to predict the actor actions in terms of the remaining information.

SS2 enables infants to attribute false beliefs and pretense to others. It allows infants to represent these divergent beliefs. This sub-system is based on at least three distinct processes: *false-belief-representation*: infants must represent other false beliefs. *response-selection*: when asked, the infants need to access their representation of other's false beliefs. *response-inhibition*: when asked about other beliefs, infants inhibit the tendency to answer based on their own knowledge.

While SS1 is present in the first months, SS2 seems to be operational only around two years old, explaining why they fail the Sally and Anne test but can succeed in VOE tasks.

2.2 Beliefs reasoning in AI and robotics

This section defines the beliefs as a mental state that is attributed to others during collaboration. We voluntarily focus on these aspects and do not intend to talk

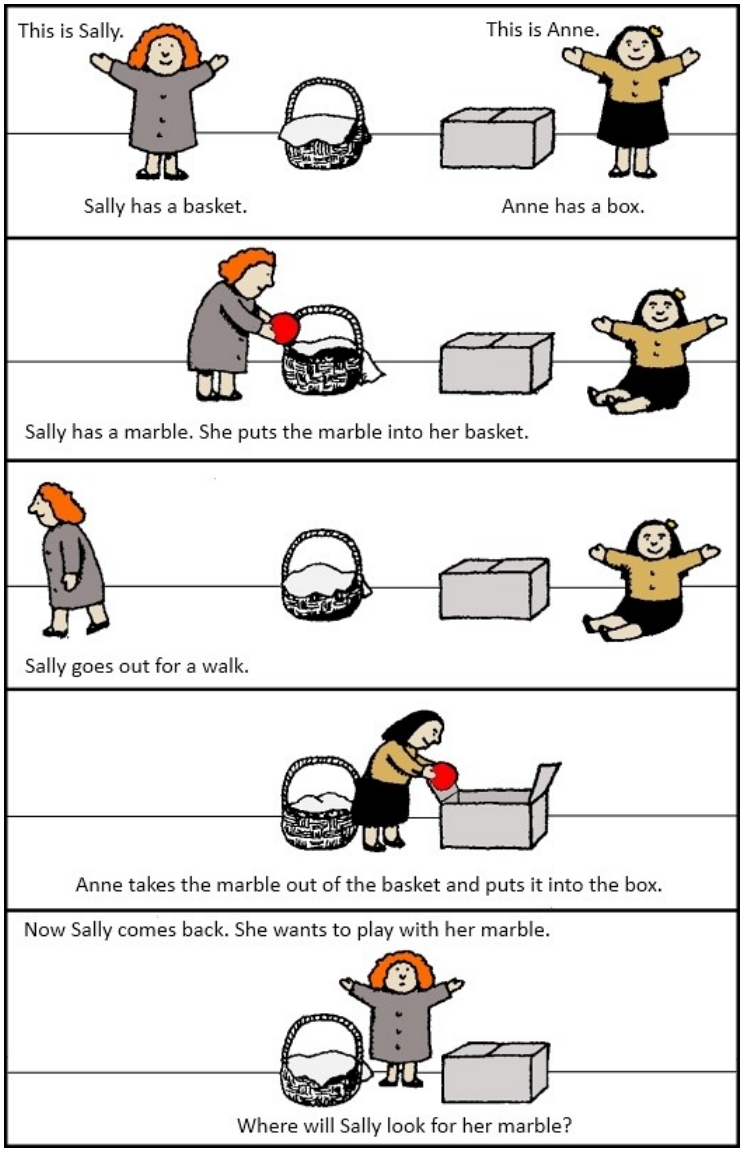


Figure 2.3: The Sally and Anne experiment from [Baron-Cohen 1985]. During this test, a scenario is played in front of an infant. Sally first put the marble in the basket. Then, in her absence, Anne moves the marble in the box. Infants were asked: "where Sally will look for her marble?". Based on the answer (the box or the basket), the psychologists stated the presence or absence of belief reasoning.

about Bayesian inferences, as it is usually used in the robotic community. Also, here we focus on the robot's capability to imagine the human viewpoint.

2.2.1 Visual perspective taking

In [Johnson 2005], the authors first introduce level 2 perspective-taking in a cognitive architecture by using a rendering engine used to generate an arbitrary view of the scene thanks to objects 3D models.

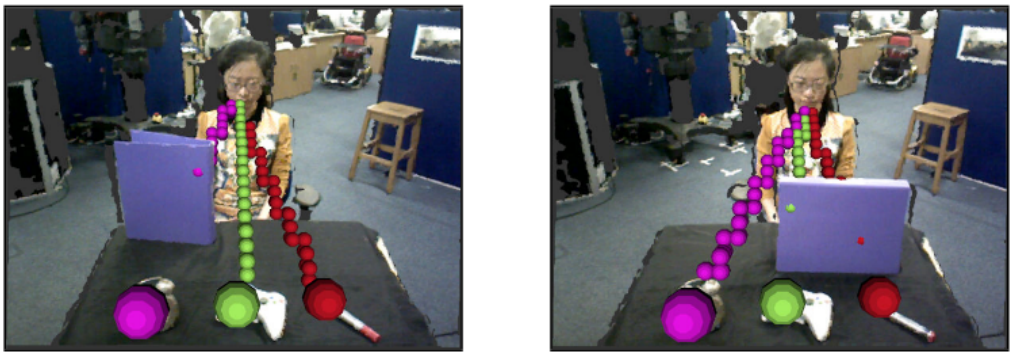


Figure 2.4: Level 1 perspective taking with voxel based representation adapted from [Fischer 2016]. Here a discrete raycasting approach in the voxel based representation is used to compute the visibility of objects.

Figure 2.5: Level 2 perspective taking with point-cloud based representation adapted from [Fischer 2016]. Here a rotation of the point cloud is used to reconstruct the image from the human point of view (assuming that the environment is already known).

In [Fischer 2016], the authors tried to get rid of the need for accurate mesh-based 3D objects by using voxel-based raytracing for level one VPT and point cloud transformation for level two. In their experiment, they assume that the robot has already scanned the whole scene, and this assumption means that the scene needs to be known apriori or scanned around overtime to maintain the 3D model.

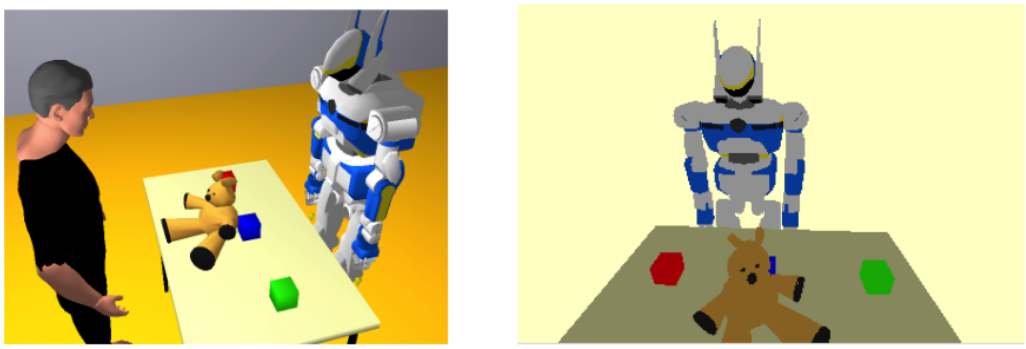


Figure 2.6: Level 2 perspective taking from [Ros 2010a]. On the left is the 3D scene from which the human viewpoint is computed (on the right).

In [Ros 2010b] and [Ros 2010a], the authors endowed the robot with level 2 perspective-taking using a rendering engine and use it to compute the level 1 perspective (whether the agent can see and object or not) using a visibility score computed by counting the pixels of the objects rendered.

In [Milliez 2014], the authors integrated the same computation for perspective-taking and demonstrated its use in the Sally and Anne test.

2.2.2 Cognitive architectures for beliefs reasoning

In [Breazeal 2009] they present an implemented system for embodied cognitive architecture. This system integrates an intention, perceptual, motor, and belief system. This architecture covers all the spectrum of belief reasoning (See Fig. 2.8). Perspective-taking is however, made simple using simple transformations about object positions.

In [Trafton 2013], they developed a framework called ACT-R for cognitive embodied agents in order to study intelligent behavior in a simulated context. This framework is heavily inspired by the research in psychology and explains human's latency or accuracy when performing tasks for studies ([Hiatt 2011]).

In [Lemaignan 2017] the authors present the first implemented architecture for joint manipulation tasks featuring geometric reasoning [Sisbot 2007]), perspective taking for verbal expression grounding ([Ros 2010a]), symbolic knowledge management ([Lemaignan 2010]) and joint symbolic planning ([Lallement 2014]). This architecture directly inspires our work because it is part of the same effort to formalize an architecture for HRI.

2.2.3 Inferring other beliefs

Being able to represent and maintain mutual beliefs with the human partner about the shared context is critical in HRI. What the robot knows is not necessarily the same than his partner.

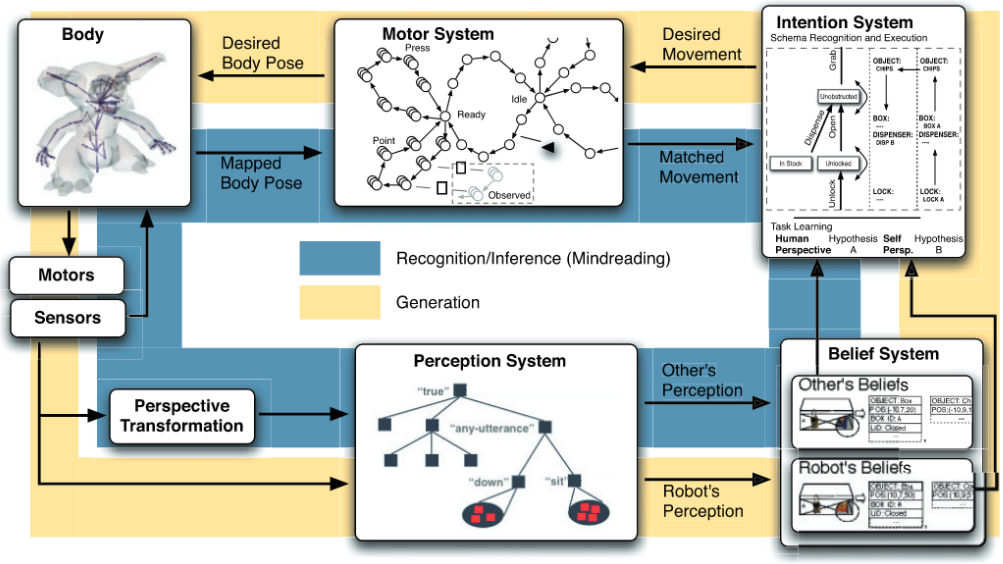


Figure 2.7: Overview of the cognitive architecture in [Breazeal 2009]

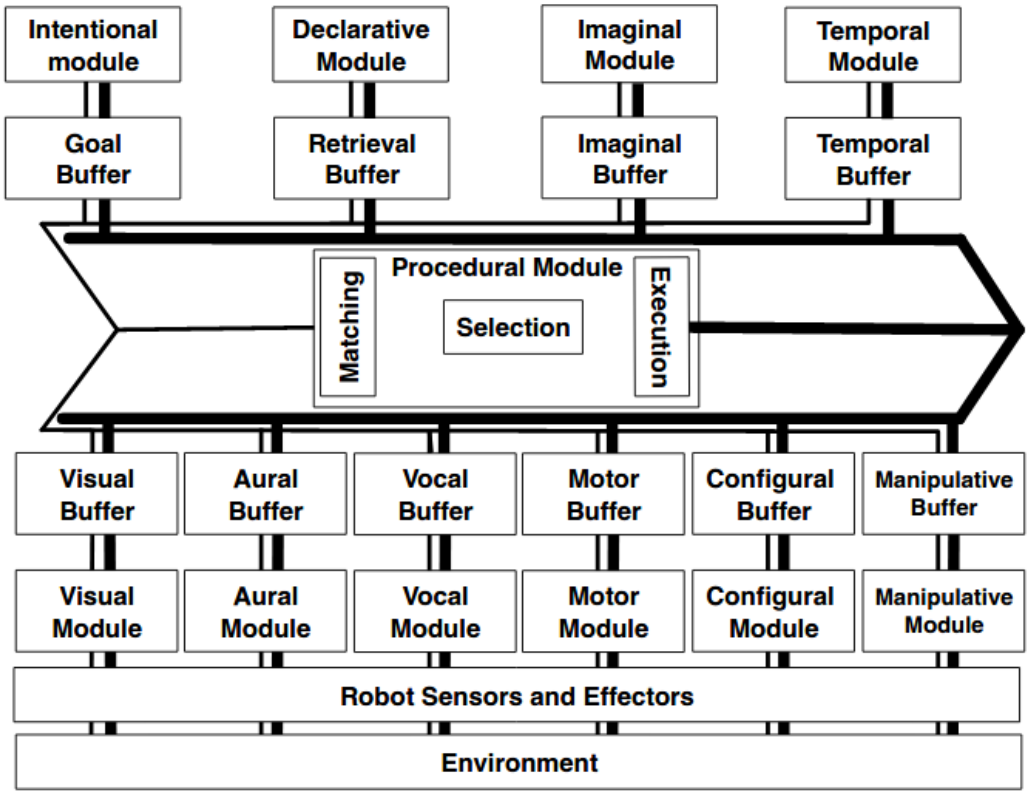


Figure 2.8: Overview of the cognitive architecture in [Trafton 2013]

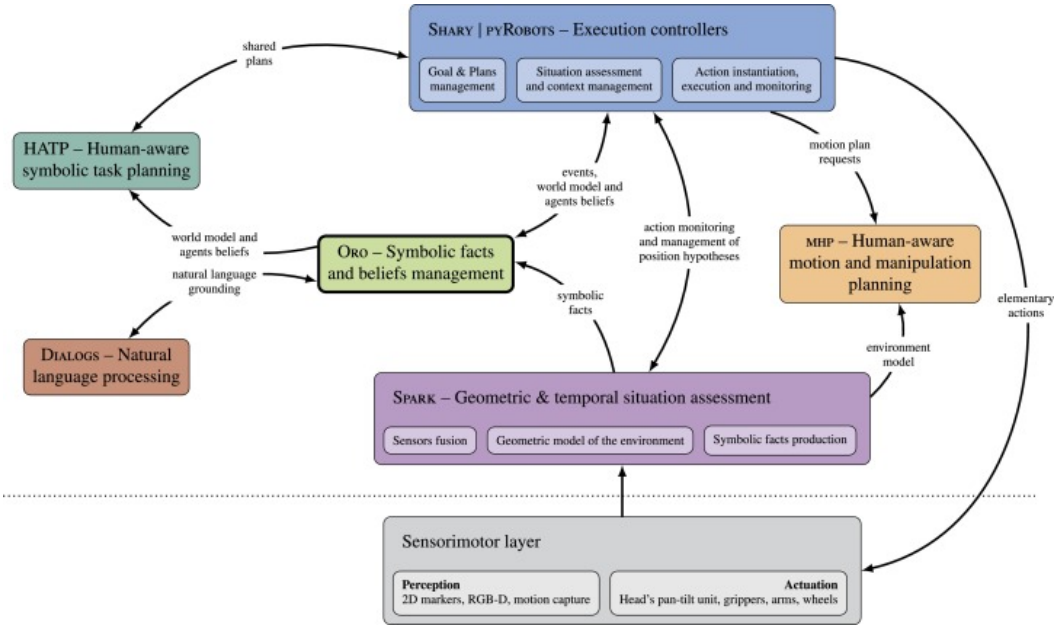


Figure 2.9: The architecture for cognitive and interactive robot presented in [Lemaignan 2017].

In [Milliez 2014], the authors estimate the beliefs of the human by attributing beliefs about objects position: if the agent is present, they attribute the object 3D position to their beliefs. By doing so they were able to run the Sally and Anne test showing management of false beliefs.

Recently in [Yuan 2020], the authors formalized an approach to merge multi-view knowledge and infer the beliefs of the humans and the beliefs of the robot. The novelty of their approach (apart from using recent neural networks for object detection) is that they use the Hungarian assignment algorithm ([Kuhn 1955]) with a cost that use features from color/object recognition to assign (like in a MOT tracker) to people the object detections (object's 2D position relative to the camera) handling this way false beliefs about the object's identity as well as position. They did not, however, account for visual perspective-taking by relying only on the agent presence.

We agree with them about the lack of unified representation for modelizing these problems and also about the lack of dataset. In chapter 3, we also proposed a unified representation based on a 3D scene graph enhanced with symbolic temporal information in [Lemaignan 2018] for representing the robot and human beliefs, which we extend to visual information in our current work (see Chapter 6). In our case, we emphasize the role of 3D mesh-based view independent representation, which allows the robot to move around while being able to compute perspective taking using rendering engines and physical reasoning using physics engines ([Sallami 2019]).

2.3 Conclusions

Reasoning on what belief agents in terms of symbolic information is crucial for human thus robots. With the significant progress of deep learning in the last years, it became evident that deep features were essential to have in any modern reasoning system. Still, the lack of structural representation in deep learning makes that at the end, the more promising approaches aim at mixing graph-based knowledge representation with deep features like in [Yuan 2020], which is the strategy that we also adopted at the end of this thesis.

Most critics about works previously done at LAAS come from the fact that the scenario did not involve state-of-the-art detectors and tracking systems but more straightforward solutions with AR tags or motion capture and was in a controlled environment. By replacing visual inputs with more novel systems that use neural networks, we can easily tackle the critics. From our perspective, this argument is not relevant because the object-based representation is still needed in any case. Integrating 3D reasoning is critical in our architecture but is challenging for sure because the models need to be created online (approximated with more or less granularity) or be known as apriori. It is the main limitation of our work and its strength because we can use that representation for rendering or physics engines that need primitives or mesh-based object models to works. However, we think that combining voxel-based and object-level representation could be the way to go in the future to reason more finely about the occlusions in the environment.

As the reader may have noticed, the cognitive architecture for human-robot interactions presented did not integrate any system similar to infants physical reasoning. It is even more surprising considering the precocity of physical reasoning in infants and its importance. This fact makes our work the first that tries to integrate processes similar to an infant's physical reasoning in a human-robot interaction tabletop setup (see chapter 5) to compute physically consistent beliefs.

One of the significant problems and challenges that we face in robotics is the scientific effort to make experiments reproducible. On the contrary, the computer vision community is used to publish datasets and replicable results, which was needed to make the last year's enormous progress. As stated in [Yuan 2020], there is no publicly available dataset to study these questions.

In order to address better beliefs reasoning in the future, we think that a world-wide scientific collaboration across researchers from computed vision, robotics, and developmental psychologist community is needed to go forward, especially when, like nowadays, enormous efforts need to be made to gather datasets in order to tackle challenging problems.

We need to make more effort in building reproducible experiment, but also in building international collaboration, which is why during this work, we emphasize the collaboration between the BRL and the LAAS and also the reason why we decided to start building together a dataset (see chapter 3) which, we hope, will help HRI community to work on more reproducible research. In [Yuan 2020], they introduce interesting setups in a natural environment where beliefs reasoning is

mandatory; in the future, we want to take inspiration from this kind of setup to enhance our dataset.

Chapter 3

Underworlds: Cascading situation-assessment

Contents

3.1 Introduction

This chapter presents the development and design of a framework for geometric and relational reasoning initiated by [Lemaignan 2017]. During this thesis, the development of this framework was central, and every work was part of an overall effort to build a framework that could be usable in different HRI scenarios, like in MuMMER or other setups.

During this thesis, this framework has been used in a different scenario, and the development was a back and forth between design and concrete applications that sometimes needed intensive work on the implementation and integration level.

That period can be discribed in three phases:

1. Design and proof of concept
2. Communication optimization and deeper integration with ROS
3. Integration of physical and visual reasoning into the core framework

The first stage happened during the start of this thesis when we started a collaboration with Séverin Lemaignan. During that phase, the fundamental principles were already present, but the software was lacking proper integration (only /tf was used to communicate with ROS) into a robotic software, and the communication protocol (HTTP2) was too heavy to cascade easily reasoning in a robotic context where real-time is critical.

In the second stage, we started a new version of Underworlds[1] by using the ROS topics and services to implement the communication protocol. By using ROS nodelets, we achieved zero copy-pointer passing communication allowing to cascade situation-assessment without suffering from delays in the communication protocol.

The last stage was integrating visual features and 2D reasoning along with the physics engine that was designed as a central component.

This decision has been made because of the success of the preliminary work on physics (see chapter 5) and to benefit from recent progress in deep visual features.

In this chapter, we present the current view of the framework[2]. We start with the inspirations and move on to the particularity of our approach by presenting this work's design and implementation aspects. We finally give some directions to improve the reasoning capabilities in the future.

Note for the readers

This chapter introduces the main principles and data structure without being exhaustive about the information actually contained in the data structure, particularly about the different types of relations/features computed in our software. To illustrate in detail the capability of our software, we present concrete examples implemented with UNDERWORLDS and document in chapter 4 and 7 the relations modelized which depend on the task envisioned. In chapter 6 we also present in detail the different types of features extracted and stored by our software.

3.1.1 Inspiration

This framework is directly inspired by geometric and temporal reasoners like SPARK (*SPAtial Reasoning & Knowledge*) [Sisbot 2011a] or TOASTER (*Tracking Of Agents and Spatio-TEmporal Reasoning*) [Milliez 2014]. SPARK acts as a situation assessment reasoner that generates symbolic knowledge from the environment's geometry concerning relations between objects, robots, and humans. It also takes into account the different perspectives that each agent has on the environment. SPARK embeds a modality-independent geometric model of the environment that serves both as the basis for the fusion of the perception modalities and as a bridge with the symbolic layer [Lemaignan 2017]. This geometric model is built from 3D CAD models of the objects, furniture, and robots and full-body rigged models of humans. It is updated at run-time by the robot's sensors. Likewise, Underworlds embeds

[1] https://github.com/underworlds-robot/uwds
[2] https://github.com/LAAS-HRI/uwds3

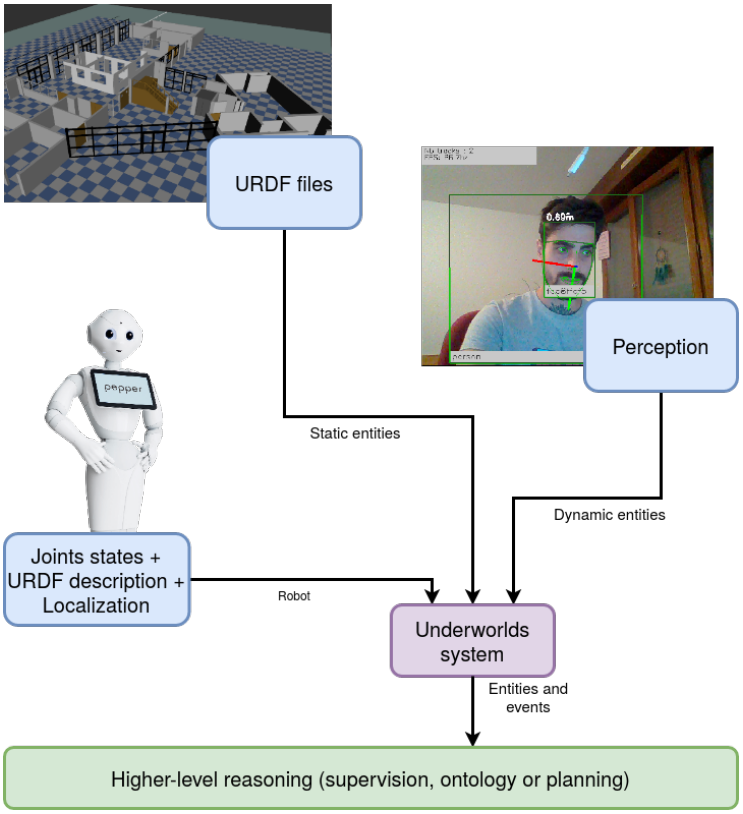


Figure 3.1: The aim of this situation-assessment system is to gather data from different modalities to generate a scene graph representing the robot's surroundings and the robot itself. It generate at runtime the events needed for the supervision system to handle a collaborative task.

a grounded amodal model of the environment, updated online from the robot's sensors.

The main difference with previous geometric and temporal reasoners like SPARK or TOASTER, is that it enables users to represent multiple world states and distribute the reasoning process by cascading world and reasoners. For instance, the world with some objects filtered out; the world viewed from the perspective of another agent; a hypothetical world resulting from the simulated application of a plan, etc. It focuses on maintaining and distributing multiple (and possibly alternative) spatial (based on 2D bounding boxes and 3D geometric primitives) and symbolic models of the physical world (based on relational events). It works as a distributed system where a set of loosely coupled *clients* provide ad-hoc reasoning capabilities (see Fig. 3.2).

Representing alternative states is, however, often highly desirable. For instance, software components manipulating environment models typically perform better if the models are physically consistent. However, low-level perception in-accuracies often introduce hard-to-avoid physical inconsistencies (like detected objects floating

in the air or wrongly inserted into other objects). Therefore, a post-process stage (for instance, using a physics simulation engine like in chapter 5) is needed to move the objects seen by the robot into physically correct positions.

3.1.2 Principles

In a nutshell, UNDERWORLDS is a software to maintain and build multiples (future, past or alternative) dynamic directed graphs called *worlds* that embed semantic (in the sense of semantic relations) and geometric (shape, position, and motion of objects) information about the scene. It can distribute it on demand and allows to combine of situation assessment components (like with a preception pipeline) depending on the task requirements; and to implement quickly new reasoners, as presented in [Lemaignan 2018]. In many ways, Underworlds can be viewed as a set of world states that cascade where geometric and symbolic models are tightly coupled (see Fig. 3.2).

To be able to work on these aspects, we needed a data structure that satisfies our purpose. (1) Be compatible with already existing 3D software (physics engine, renderer). (2) Use the same data structure to represent the robot's beliefs and others beliefs (and maintain multiple versions of it). (3) Be able to represent temporal events to reason about time durations and relations between entities.

3.2 Design

In this section, we present the fundamental design aspects that we emphasized during UNDERWORLDS development. First, we present the world states representation and the cascading principle to give the reader more details about the data structure and core processes.

3.2.1 Cascading worlds representation

UNDERWORLDS is able to store and maintain multiple world states, each one composed of a 3D scene graph and a timeline. What makes different Underworlds from its predecessor is the capability to cascade components in a modular fashion and to maintain multiple versions of the world state. For doing so, we needed a unified representation that integrates all the data that a situation-assessment component may need for run-time reasoning: the *scene graph* that represent the entities along with the geometric information and the *timeline* that store the events and symbolic relations about the entities that change over time.

UNDERWORLDS can be viewed as a set of clients (ROS component) and worlds (ROS topics) that cascade to perform a reasoning process for situation assessment. It can be viewed as an extension of a classical perception pipeline where is added the semantic relations between entities.

The clients can be of different types depending on their role in the pipeline:

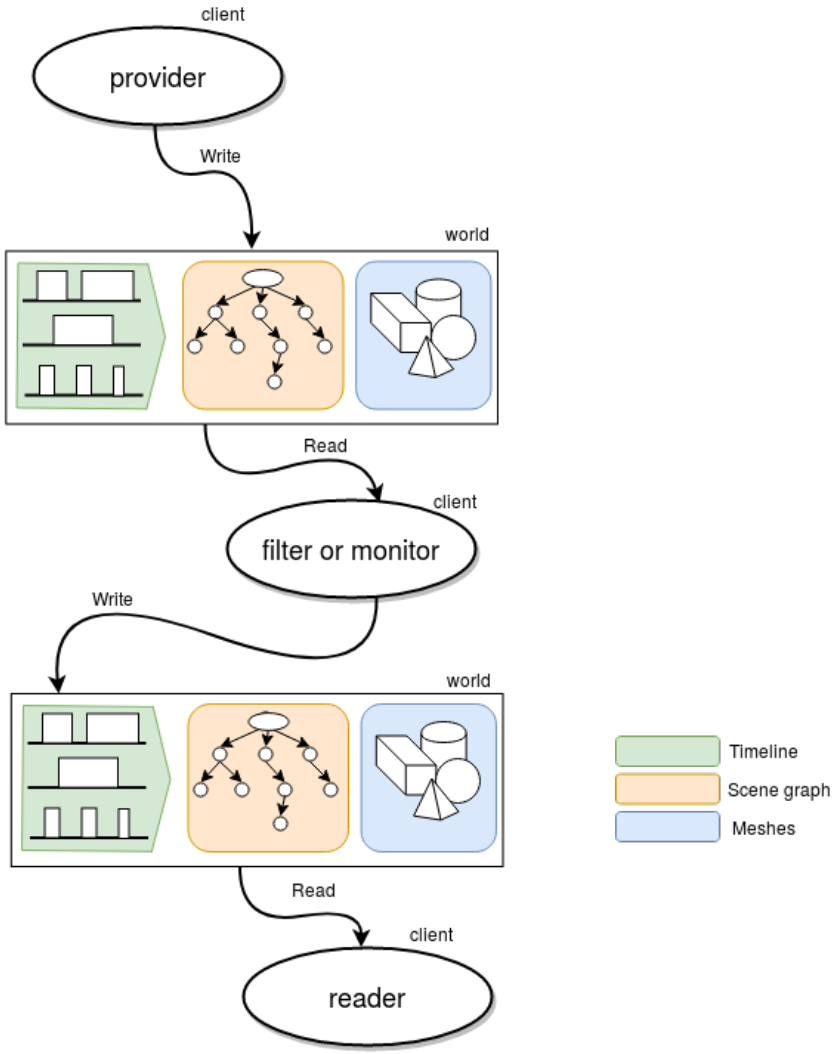


Figure 3.2: Underworlds is from the user-perspective a set of clients and worlds that cascade to generate the world state of the robot in order to be used by the supervision, reasoning and planning levels. Since the world states share the same structure, it is possible to reconfigure the reasoning pipeline to adapt it to a new task. Providing modularity and easy integration with other components. See Fig. 3.5 for more details about the implemented architecture.

- PROVIDER: they provide the input data to the system like perceptual data in the form of nodes and eventually events (when a new person/object appears, for example, or external events).
- MONITOR or FILTER: they compute the events (and relations) among entities and are responsible for correcting, enhancing or generating alternative world states.
- READER: they read and use the data generated for supervision, planning, or

Table 3.1: The world data-structure.

attribute type	attribute name	comment
list of SceneNode	scene	(see Table 3.2)
list of TemporalSituation	timeline	(see Table 3.8)

long-term reasoning.

In this work, we propose a unified data structure to represent 2D/3D geometric, visual and relational data composed by a 3D scene and a timeline of events, in the following table 3.1 is described the world data-structure definition as it is implemented in the ROS messages used for communication. Then we present more in detail each component of this world state.

3.2.1.1 The 3D scene graph

Each world represents the geometry by a set of *scene nodes* that represent the entities of the scene (See Table. 3.2 for the details of the data structure) and compose the 3D scene graph. Each entity can have a camera (represented by a classic pinhole model allowing image rendering), a primitive 3D shape (sphere, cylinder, or box), or a 3D mesh attached to it (See Figure 3.2). Moreover, each node refers to its parent with a position, velocity, and acceleration (plus their respective covariance). The scene nodes also have the particularity that they can represent simultaneously, the 2D (in the image space) and the 3D representation of an object (in the global space).

The scene graph representation is common to every 3D mesh-based software, from rigid body physics simulation to video games engines and rendering engines like OpenGL.

The goal of Underworlds is to build this 3D scene graph using a static environment (a 3D file that contains the prior about the environment), the state of the robot (including the joints states and localization), and its perception, by fusing these pieces of information and reasoning on it, the robot can build a geometric 3D model of the scene (See Fig 3.1).

3.2.1.2 The timeline

Along with the geometric representation composed by the scene nodes, additional edges in the scene graph represent the temporal events (named situations) that give the user additional information about the object interactions at a symbolic and temporal level. It can be used for facts, actions, or captions. The situations are removed when finished after a determined time (few seconds generally) to avoid any bottleneck. Consequently, the timeline does not provide long term data-storage but can be viewed as a working memory for symbolic relations (see chapter 7 for interaction with long term memory). Besides the situation, additional information can be added like a 3D point (to locate events/actions) or additional features data.

Table 3.2: The scene node data-structure. *Every 3D data use ROS header and is expressed relatively to the frame provided

attribute type	attribute name	comment
strig	id	The unique ID of the node
uint8	type	The node type (object, myself, other)
uint8	state	The state of the node (managed by the tracker and monitors)
string	label	The class label (generally provided by the detectors)
string	description	The human friendly description
bool	is_static	True if static
bool	is_perceived	True if 2D bbox is valid
BoundingBox	bbox	(see Table 8.2)
bool	is_located	True if the 3D position is valid
PoseWithCovariance*	pose_stamped	The 3D pose (provided by the 3D Kalman filter)
TwistWithCovariance*	twist_stamped	The 3D velocity (provided by the 3D Kalman filter)
AccelWithCovariance*	accel_stamped	The 3D acceleration (provided by the 3D Kalman filter)
bool	has_shape	True if has a shape
list of PrimitiveShape	shapes	(see Table 3.5)
bool	has_camera	True if has a camera
Camera	camera	(see Table 3.3)
list of Features	features	(see Table 6.1)
list of Property	properties	(see Table 3.7)
time	last_update	The last update of the node
duration	expiration_duration	The expiration duration

Table 3.3: The camera data-structure used for rendering.

attribute type	attribute name	comment
float	clipnear	minimum distance rendered
float	clipfar	maximum distance rendered
CameraInfo	camera_info	standard ROS message for cameras

Table 3.4: The bounding box data-structure used to store 2D data.

attribute type	**attribute name**	**comment**
int32	xmin	the x dimension of the bbox top left corner
int32	ymin	the y dimension of the bbox top left corner
int32	xmax	the x dimension of the bbox bottom right corner
int32	ymax	the y dimension of the bbox bottom right corner
bool	has_depth	True if depth is valid
float64	depth	The depth relative to the sensor view

Table 3.5: The primitive shape data-structure used to store 3D data.

attribute type	**attribute name**	**comment**
uint8	type	type of primitive shape (box, sphere, cylinder, mesh)
string	name	the name of the shape
Pose	pose	the pose of the shape relative to the attached node
list of float	dimensions	the dimensions of the shape
ColorRGBA	color	The color of the shape
Vector3	scale	the scale of the shape (used for meshes)
string	mesh_ressource	the path to the 3D mesh file (used for static mesh)
list of MeshTriangle	triangles	the trangles indices of the mesh (used for dynamic mesh)
list of Point	vertices	the vertices (used for dynamic mesh)
list of ColorRGBA	vertex_color	the vertices color (used for dynamic mesh)

Table 3.6: The features data-structure used for machine learning.

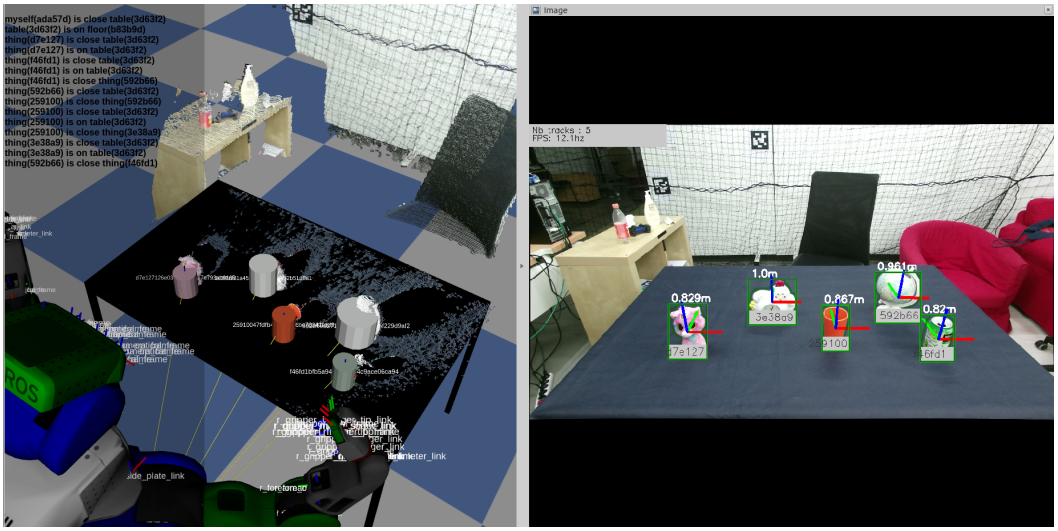
attribute type	**attribute name**	**comment**
string	name	the name of the features
list of float	data	the features data
float	confidence	the confidence score

Table 3.7: The property data-structure used to store additionnal meta-data.

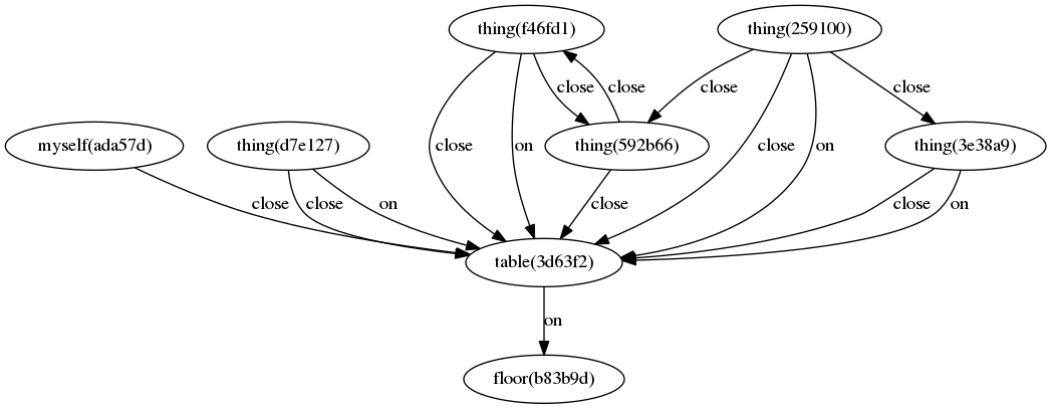
attribute type	attribute name	comment
string	name	the name of the property
string	data	the property data
float	confidence	the confidence score

Table 3.8: The situation data-structure. *Every 3D data use ROS header and is expressed relatively to the frame provided

attribute type	attribute name	comment
string	id	The unique ID of the situation
uint8	type	The type of situation (fact, action or caption)
string	description	The human friendly description
string	predicate	The predicate of the relation
string	subject_id	The subject node ID
string	object_id	The object node ID
bool	is_located	True if the 3D point is valid
Point*	point	The 3D point where the situation happened
list of Features	features	The features to be used by machine learning (list of float)
list of Property	properties	The properties (additionnal data in string format)
time	start	The start time
time	end	The end time
duration	expiration_duration	The expiration duration



(a) Unknown objects are first detected and tracked over time. Then we analyse the physical plausibility([Sallami 2019]) in order to correct the object poses. Finally allocentric spatial relations are computed in a robust and efficient way using aligned-axis bounding boxes test.



(b) The resulting graph data structure generated by Underworlds. This graph is composed by the nodes of the 3D scene graph and the situations (the edges) that compose the timeline.

Figure 3.3: A tabletop example of use of Underworlds for situation assessment.

3.2.2 Visual perspective taking

Like SPARK, Underworlds allow rendering of the scene from any point of view thanks to geometric rendering provided by the OpenGL renderer of the physics engine (See Figure 3.4). This capability is critical in our context because modelling what can see other agents is the basis of belief reasoning. A visibility score is computed from that rendered view by counting the pixels for each entity in the 2D image rendered.

Additionally to the visibility computation, we extract the objects with the same formalism as perceptual data (2D bounding boxes + depth + 2D mask) thanks to

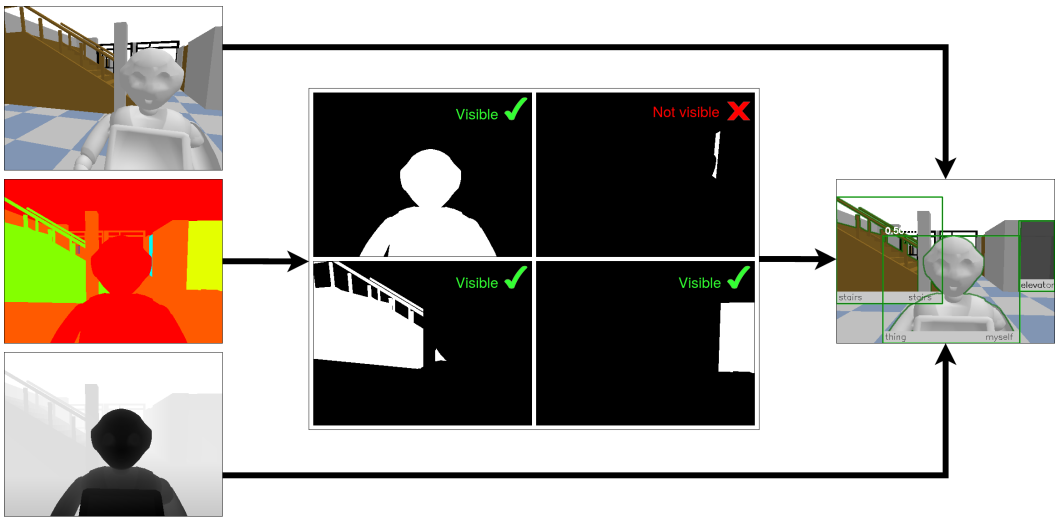


Figure 3.4: The Perspective-taking computation. Based on the mask image generated from rendering, we compute a visibility score and generate alternative scene nodes that represent the objects from the other perspective, allowing us to use reasoning algorithms in the same way as the robot perception perceived them. The depth map is used to compute the depth of the objects relative to the camera.

the depth buffer and mask image generated by the renderer. This process allows the robot to apply then the same reasoning as if the robot perceived the objects. Such visual perspective is better if we have the object 3D model, but it can works also with primitive approximations for small unknown objects.

3.2.3 Geometric spatial relations

Two different types of spatial relations computation are made available with the core software: allocentric (view independent) and egocentric (view-dependent). The allocentric relations (in, on, close) are computed in 3D using the 3D aligned axis bounding boxes (AABB), while the egocentric relations (left_of, right_of, behind) are computed in 2D in the image plane.

This choice has been made because we typically use perspective-taking to compute the view of the human partner with visual perspective-taking to infer view-dependent relations.

3.3 Implementation

3.3.1 Base types

In order to be easier to use, we implemented the base types corresponding to the ROS message for the user to access helpers methods. In particular, we implemented with *opencv* various linear Kalman filters/trackers for scalars, 2D bounding boxes, 2D vectors, 3D vectors (with optional acceleration) and 6D vectors (composed of two

3D vectors filters) to accommodate better with noisy data input. These Kalman filters efficiently allow computing the velocities of the entities in the scene while smoothing the trajectories by integrating noisy observations over time.

3.3.2 Physics engine

One of the strengths of Underworlds is the use of a simulator at run-time, which gives consistency to the perceived world state and allows to detect of object-related actions as well as infer not visible entities based on their interaction with others objects.

We decided to follow [Weitnauer 2010], and we have chosen Bullet RT physics engine ([**?**]), because (1) game-oriented physics engine is optimized towards large scale simulations (hundreds of bodies), and (2) contrary to usual simulation in robotics, speed and stability are preferable to accuracy in our context. Besides, Bullet is already integrated into ROS (the TF library uses Bullet datatypes, for instance), which facilitates future reuse of this work. Moreover, deformable objects have been recently added to Bullet.

This work has been presented in[Sallami 2019](See chapter5 for details). Since then, we made available with the base software the bridge to interface with pybullet([Coumans 2020]), a Python binding of Bullet RT simulation.

3.3.3 Communication protocol

Underworlds main purpose is to be share and maintain world states with a communication protocol in order to store many parallel worlds and distributes them among clients(reasoners). In the first versions[3], for bandwidth efficiency, only the changes were broadcasted, and a central server was used to fetch the full state of the worlds at the initialization of the clients. Then a lazy update mechanism updates the local data structure of the clients and calls the inference process. In its last version, the protocol has been simplified by completely distribute the data and rely only on the inter-clients communication (ROS topics and queues) by broadcasting the full state of the system, getting rid of the central server.

The implementation of the communication protocol of Underworlds was made easy to implement using ROS communication entirely. In a nutshell, it mainly consists of a message package[4] that can be used by any ROS software.

The different ROS messages used were designed to be integrated seamlessly into ROS by using standard messages as far as possible. By using ROS topics, we also allow zero-copy pointer passing between nodelets. Making the communication time negligible thus allow cascade worlds without worrying about inter-client communication.

[3] https://github.com/underworlds-robot/underworlds
[4] https://github.com/LAAS-HRI/uwds3_msgs

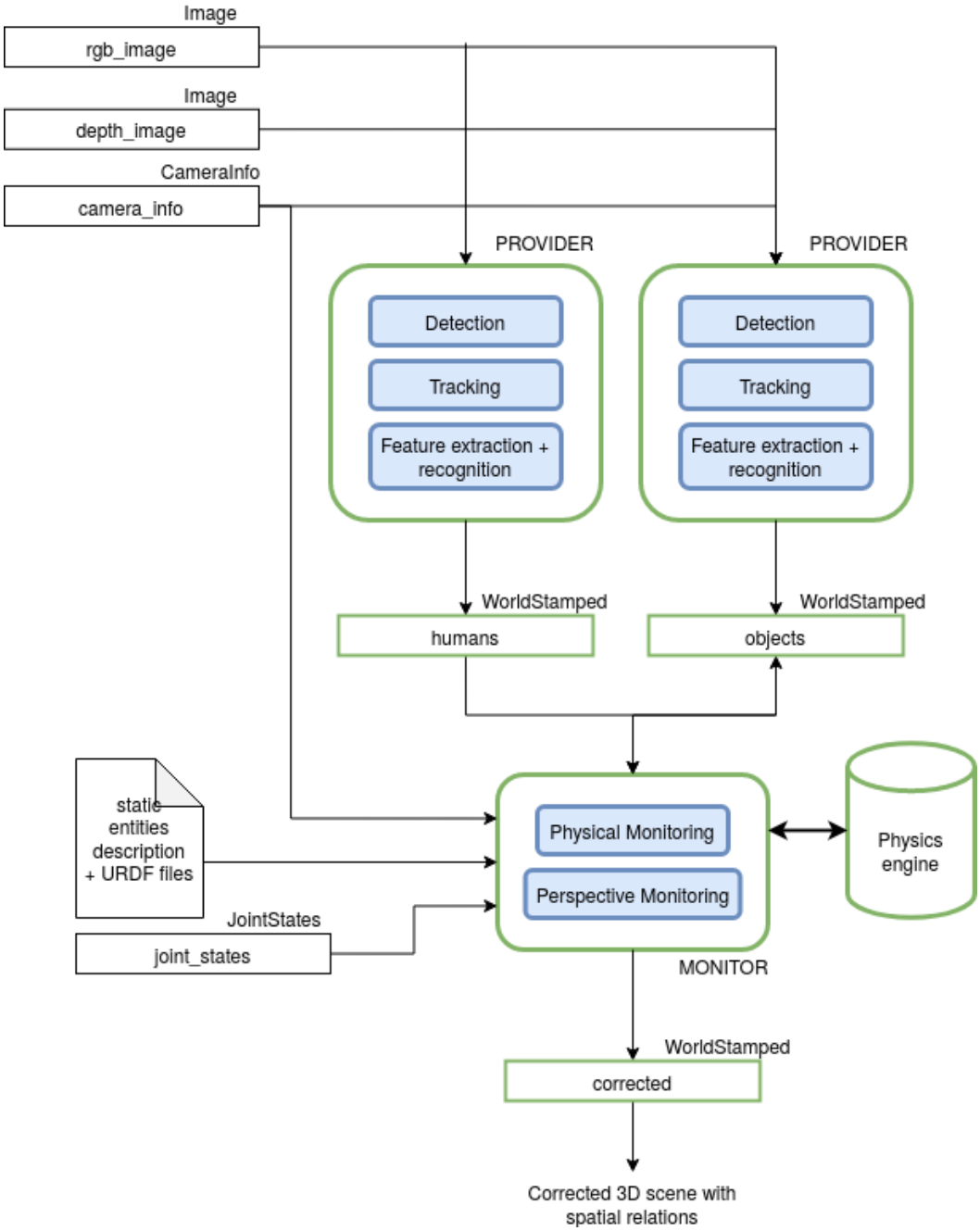


Figure 3.5: The architecture details for the tabletop example. The green lines represent what compose the core principles of Underworlds a framework to share and distribute world states. In blue are the reasoners made available in the *pyuwds3* library (See next chapter).

3.4 Conclusion

In this chapter, we presented the design and implementation of a modular and functional architecture for situation assessment. This architecture fully integrated into ROS allow the robot to reason on geometric and physical aspects of the world states while providing symbolic knowledge about the relations between objects or actions. This framework also provides perspective-taking capabilities by exploiting the depth-buffer along with the mask image.

The modular aspects of this framework make it easily usable in a research context where prototyping reasoning pipelines can be time-consuming.

3.4.1 Future work

In the following section, we briefly present directions that could be taken for future work. In particular, we explore the advantages that could benefit this software by combining voxel-based representation.

3.4.1.1 3D occupacy grid

In the first design of the framework, we wanted to integrate octree representation to provide a modality to complement the reasoning at the object level. Indeed volumetric data do not need known 3D models (or primitive approximation) for objects and therefore are best suited for unknown environments. One interesting approach is combining object-based representation with an octree that can model the volume of what is not detected by the objects or human detectors.

Unfortunately, we did not have time to explore this aspect, but we think that in the future, adding octree representation could enhance significantly the geometric capabilities, as well as perspective-taking, by allowing discrete raycasting to compute visibilities. In the architecture, the octree representation could be integrated with the same fashion as the physics engine and be accessible to monitoring processes to perform raycasting, occupation tests, or know what part of the scene has not been explored.

Chapter 4

MuMMER: MultiModal Mall Entertainment Robot

Contents

4.1 Introduction

The European H2020 MuMMER project has funded this thesis ([Foster 2016, Foster 2019]), where the aim was to develop a humanoid robotic platform (based on SoftBank's Pepper) that could interact with people naturally and dynamically. The robot had to be able to run autonomously during days. The robot performed guidance to various locations within the mall, small-talk, and playing quiz games with customers.

The robot was deployed in IdeaPark[1], a finish shopping mall (see Fig.4.1). This chapter introduces the overall context and integration of the situation assessment software used during the project.

[1] https://lempaala.ideapark.fi/

Figure 4.1: The pepper robot interacting with a customer in a shopping mall in Finland.

To develop the situation assessment, we chose to use UNDERWORLDS as we started a collaboration with Séverin Lemaignan at that time.

In the following sections, we present the context of this work and briefly present the overall architecture. We then present more in detail our actual contribution to the project.

4.1.1 European partners

The MuMMER consortium included seven universities, research institutes, and private companies from Scotland, France, Switzerland, and Finland:

- University of Glasgow (social signals processing)
- Heriot-Watt University (dialogue system)
- Idiap Research Institute (perception system)
- LAAS-CNRS (guiding task)
- Softbank Robotics Europe (The creators of the hardware platform)
- VTT Technical Research Centre of Finland (our main collaborator on site)
- Ideapark (where the robot was deployed)

4.1.2 Challenges

The challenges in this project were multiple. First, we had to develop with our partners a robotic platform that integrates: speech-based conversational interaction with non-verbal communication, motion planning, audiovisual scene processing,

ontology-based reasoning, task supervision, shared perspective planning, conversational AI, perspective-taking, and geometric reasoning.

In this project, the integration with other components was key and involved many challenges in terms of coordination between partners. Moreover, the project aimed to envision a complete demonstration of a robotic system in a real-world setting with long-term interactions. The LAAS-CNRS was mainly involved in the development of the guiding task and reactive navigation.

4.2 Guiding as a joint activity

One of the needs formulated by IdeaPark was that the robot should be able to guide customers within the mall. This challenging task has been designed as a joint activity ([Khambhaita 2020]), where the robot and the customers have to collaborate (share space and perspectives) in order for the robot to guide naturally and efficiently. For that task, we developed a geometric model that describes the shop's landmarks.

4.2.1 Human-Human study

To first acquire knowledge regarding this route description task conducted in collaboration with VTT, a first study of the human-human guiding task at Ideapark ([Belhassein 2017]). The study reproduces the same setting to get information about how to handle this task. Indeed, this first exploratory study aimed to understand the key elements of a route description situation and explore the stereotypic patterns occurring in gazes, hand gestures or relative positions through observational analysis of recorded videos (see Fig 4.2).

During that preliminary study, two persons working at the information desk of Ideapark were asked by VTT researchers for several locations and shops within the mall, referring to typical questions from real visitors. This information came from preliminary interviews. The questions covered all the directions of the shopping centre, more or less far from the assumed future location of the robot. Some questions have also been asked to create more complicated situations, like asking two places in the same question or questions covering several possible locations (e.g. shoe shops). This study involved 10 participants and was conducted in September 2017.

4.2.2 Architecture

Based on previously developed architecture at LAAS for cognitive and interactive robots ([Lemaignan 2012]), we developed an architecture to perform guiding as a joint activity. We studied how the robot could share his space with a human partner to perform joint attention about landmarks in the shopping mall. In order to give the reader a global overview of the project, we quickly present in the following sections, one-by-one the components developed at LAAS presented in Fig. 4.3.

Figure 4.2: The placement when guiding people in the mall is crutial [Belhassein 2017]. Placing the robot in order to re-create this triangle (called F-formation) is the main goal of the joint task.

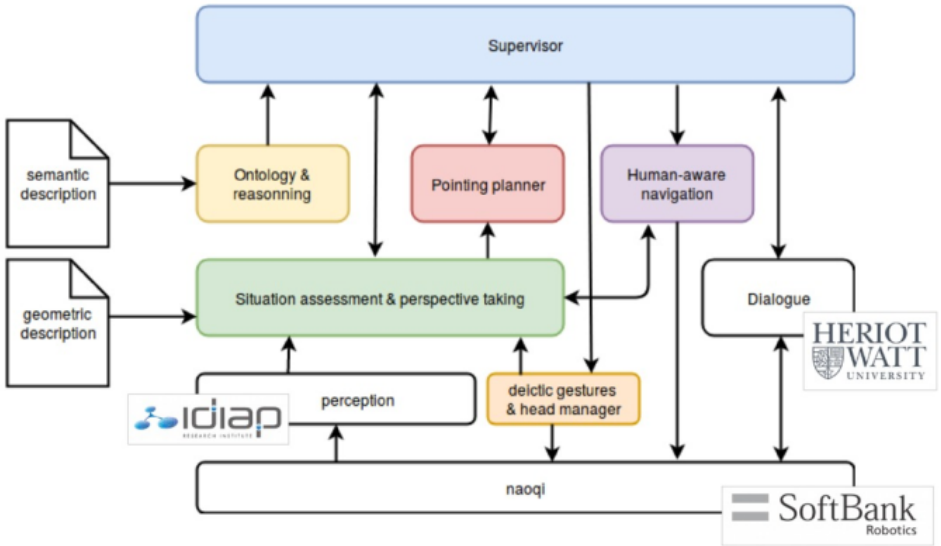


Figure 4.3: The architecture used in the MuMMER project for the guiding task. The architecture involve many component interactions from multiple partners. The colored component are developed by the LAAS to handle the guiding scenario as a joint task.

4.2.2.1 Pointing Planner

The placement of the robot relative to the landmarks designed and the customer is key in this application, as such, we developed a geometric planner that takes into account different constraints(robot and human visibilities) for the robot and human, which were modelized as a cost over a discretized 3D space ([Waldhart 2018, Waldhart 2019]) (see Fig 4.4).

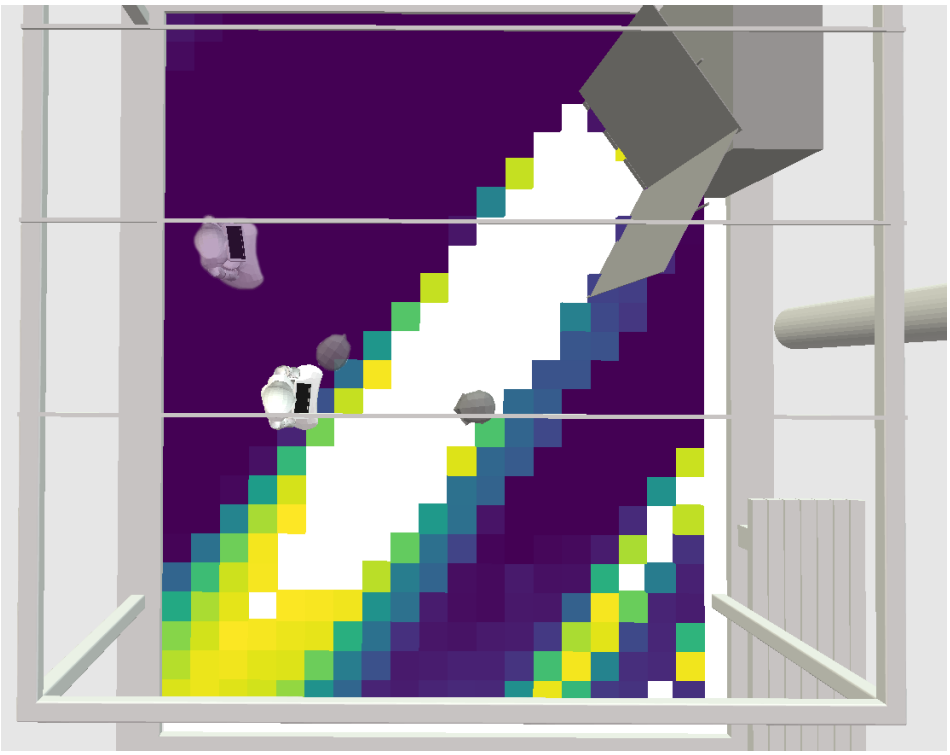


Figure 4.4: Costmap computed by the pointing planner, which allows computing the desired position for the human and the robot. In blue are the area where the landmark is visible, In yellow is the area where the landmark is barely seen, and in white where it is not visible.

Due to the limitations of the hardware platform, the robot could only move around a limited space. Therefore the robot optimizes this limited space to place itself correctly to point landmarks used in the route description.

4.2.2.2 Human-aware navigation

When the supervision decided (based on the pointing planner output) that there was a better place to point, a navigation goal was to send the robot. Because space was limited, and the safety aspects of moving a robot such as close, HATEB, a human-aware reactive planning scheme that used elastic bands ([Teja 2020]).

4.2.2.3 Ontology and knowledge processing

The ontology-based system is used here as a database for the items in the shop and computes the symbolic path towards the different shops of the mall used to verbalize the route direction to people ([Sarthou 2019]).

4.2.2.4 Deictic gesture

This component generates and executes deictics gestures (pointing and head cues) using the robot's low-level motor system. It controls the motion of the arms for pointing and the head of the robot.

4.2.2.5 Supervision

The supervision system ([Mayima]) is in charge of performing the guiding task by calling the different components and monitoring the possible incongruity that could occur, for example, a person that we started to interact with go out of the view or is not engaged anymore in the task. Another case is when the robot moves to show specific landmarks and expects the person to follow him, but the person stays in the same place.

4.2.2.6 Situation-assessment and pespective taking

This component is in charge of fusing the perceptual information with the robot localization and the mall's 3D geometric model. This component's role is to generate first-order predicates about the surrounding of the robot that is then used by the supervision system to handle the interactive task. In the following section, we present its integration with the others component of the architecture.

4.3 Situation-assessment inputs

4.3.1 Human perception

During this project, the perception was done by IDIAP. They were responsible for providing the detection and tracking of the faces, recognizing people using facial recognition, and localizing who was speaking in order to handle the interaction and dialogue task.

4.3.2 Geometric model

To compute the perspective of the human the robot interact with, we needed a 3D geometric description of the mall to be rendered. That model (see Figurure 4.5) is known as apriori and was loaded at the start. For the construction of those models, we chose to represent only the entities visible from the place where the robot was. Indeed, the robot had to interact with people in the mall's main area

but could guide them to shops that were not visible. For shops or places that were not visible, only a 3D position associated with the shop were used to describe them.

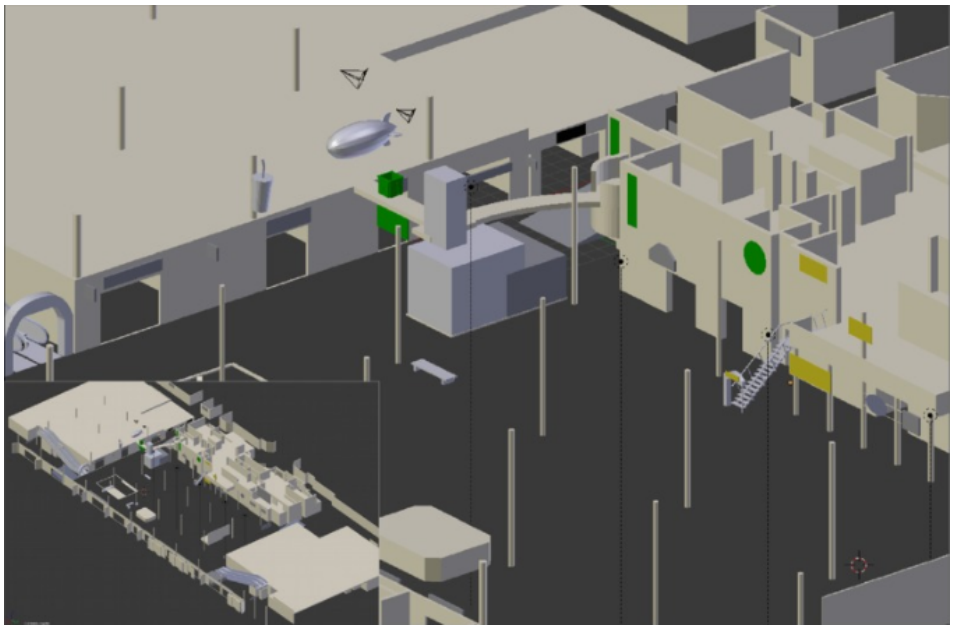


Figure 4.5: The 3D model of the mall used in the MuMMER project. The green and yellow objects represent the shop logos used as landmarks during the interaction.

4.3.3 Feedback from others component

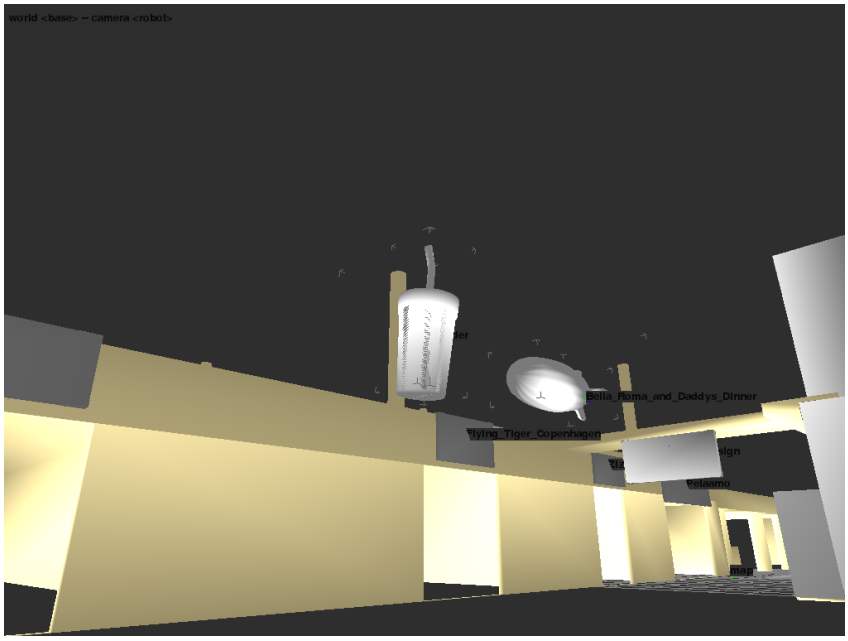
4.4 Contributions

The situation assessment in this architecture aims to build a 3D geometric model of the world with associated semantic relations to be then used by the supervision system. For the implementation of the system, we decided to use UNDERWORLDS([Lemaignan 2018]), which is a framework to build situation-assessment pipelines by cascading worlds states. The reasoning can be seen as a data flow between modular components that provide had-oc capabilities. The technical contribution to the project was components specially designed to interface with the perception provided by our partners at IDIAP in order to encapsulate the perception information into a data structure exploitable by the supervision system. Also, this component was in charge of maintaining the 3D model of the environment along with the robot.

4.4.1 Predicates for joint task handling

In order to handle the guiding task, the supervision system used the predicates generated by the situation assessment. Several predicates were a combination of information already made available by the perception modules, while others were

(a) The view of the robot within the mall



(b) The corresponding view of the robot rendered from the 3D model.

Figure 4.6: 3D model of the mall and corresponding real view.

provided as feedback from navigation or supervision ($isNavigating$, $isApproaching$, $isMonitoring$) or from the visual perspective-taking ($isVisibleBy$). See Table 4.1 for the exhaustive list of the predicates generated. The supervision system used the predicate $isEngagedWith$ to start/stop the interaction with someone during guidance while using the predicate $isClose$ to tell people to come closer when needed during the interaction.

The situation-assessment provided also 3D points of interests that was indicating where to look at by using the following rules:

- When interacting with people, the robot looks to the closest person who speaks. Otherwise, the closest person perceived.
- When navigating the robot, look in front of him
- When approaching someone, the robot look at the person's face

However, the points of interest were not necessarily used. The supervision was deciding through the head manager to use it or not by preempting the robot head gaze to send head cues during the guiding task.

4.4.2 Perspective taking

The situation assessment module's main contribution was the perspective-taking capability to monitor what people could see and check that the pointed landmarks had been seen. The Fig. 4.7 illustrate it.

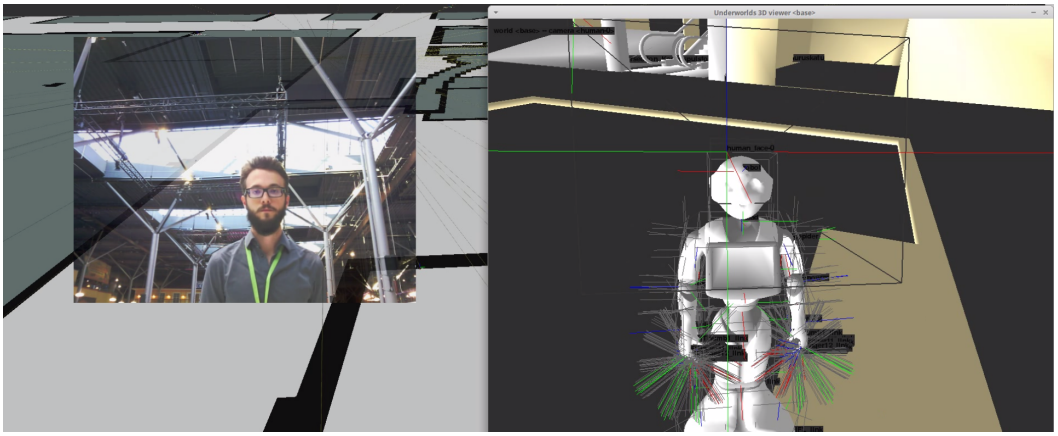
4.5 Conclusions

In this section, we conclude this project by discussing the limitations of the system and the lessons learned during this project.

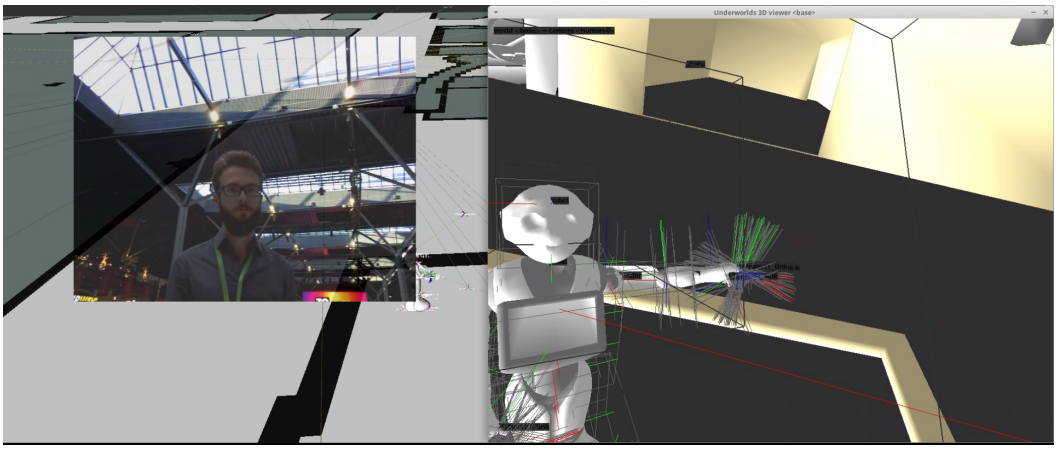
4.5.1 Limitations

The main limitation of this work was because people were tracked only using head detection. During the collaboration, people could easily go out of view: looking away to the landmarks or going out of view when the robot sends head cues. Because the perception system was used in long term interactions, facial recognition was less effective as more people were memorized in the system. Possible enhancement could also be to reason on the people's body and recognize them using the whole body's appearance. During the project, the recognition was shunt by relying on the fact that the closest human was most probably the good one.

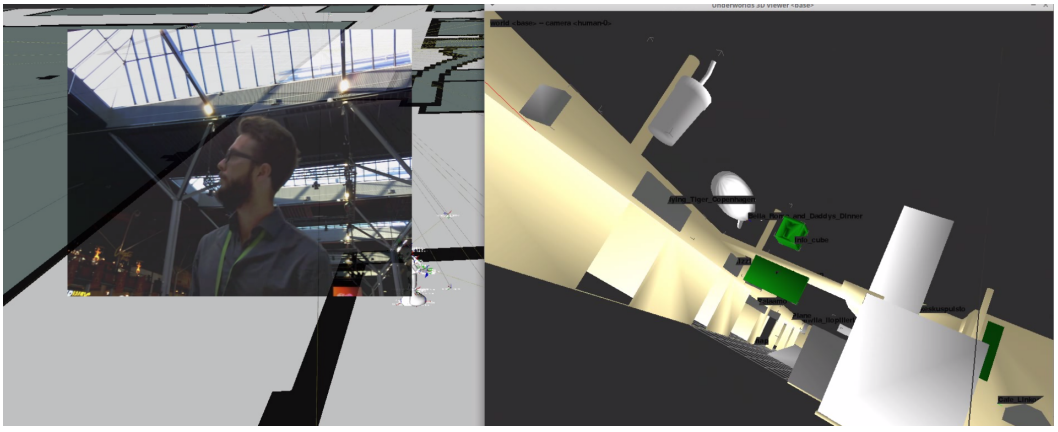
The other limitation of the system was due to the configuration of the robot during the guiding scenario. Our architecture is best suited for clustered environments, and in the final deployment, the robot was deployed in a big open environment, which limited the usefulness of visual perspective-taking. For the final deployment, the visual perspective-taking was disabled, and the robot relied only



(a) First, the interaction start and the actor ask for a shop.



(b) The robot move to the best place to point and point toward the first landmark of the route.



(c) Then it check that the actor is looking at the landmark.

Figure 4.7: The pilot scenario deployed at IdeaPark. Here we chose the best landmark to point (which is difficult to see) in order to show the pertinence of the perspective-taking.

Table 4.1: The predicates computed by the situation assessment.

predicates	**input data**	**rule**
isPerceiving(Robot,Agent)	face tracking	True when perceived by the robot perception, False otherwise.
isSpeaking(Agent)	sound localization	Tue when someone speaking, False otherwise. (raw data from IDIAP)
isLookingAt(Agent,Agent)	head pose	True when looking at someone, False otherwise. (raw data from IDIAP)
isSpeakingTo(Agent,Agent)	head pose and sound localization	True when looking at someone while speaking, False otherwise.
isCloseTo(Agent,Agent)	head position	True when the distance between people is less than one meter, False otherwise.
isEngagingWith(Agent,Agent)	head pose	True if looking at someone and close to him/her, False otherwise.
isEngagedWith(Agent,Agent)	head pose	True if engaging for more than 3 secs and False when looking away for more than 5 secs
isNavigating(Robot)	navigation	True when the robot is navigating, False otherwise.
isApproaching(Robot,Human)	navigation	True when the robot is approaching someone, False otherwise.
isMonitoring(Robot,Agent)	supervision	True when interacting with the robot, False otherwise.
isPointingAt(Robot,Object)	deictic gesture	True when the robot point towars an object, False otherwise.
isVisibleBy(Object, Agent)	perspective taking	True when the object is visible from an agent rendered view, False otherwise.

on verbal communication to check if the landmark was visible by asking the person it was interacting with.

4.5.2 General conclusions about the project

During this project, the contribution of the situation assessment was limited in terms of novelty. Besides that, this project was the occasion to face many integration challenges that roboticists need to handle due to the diversity of reasoning a robot needs to perform a joint task. In that aspect, the previous development of UNDERWORLDS framework made that task more straightforward, and we had the opportunity to show the modularity of this situation-assessment component in a real-world scenario.

Chapter 5

Simulation-based physics reasoning

Contents

5.1 Introduction

In this chapter, we present a situation assessment component that builds a consistent estimation of the scene observed by the robot with the help of a physics engine. This work has been published in [Sallami 2019], and we present that publication in this chapter. In conclusion, we discuss the novel work that we have done in the directions suggested in [Sallami 2019] and give future directions.

In this work, the physics engine is used in real-time to reason about objects occlusions, gravity, collisions, and other associated geometric features, such as the surfaces on which objects are laid or the contents of boxes or containers. This

high-level reasoning pipeline analyzes geometric violations and corrects the object's poses, even when out of sight.

It has been designed as an extension of a pre-existing perception pipeline to provide corrected object poses, interpretation of the scene spatial configuration, and recognition of human actions by analyzing the transition of objects from physically plausible to not physically plausible states. Examples of such capabilities include: inferring that a hidden object has likely fallen onto the floor (see Fig. 5.6a); inferring that an object is being transferred from one container to another (see Fig. 5.6c). Based on the output of the reasoner, we can also compute estimations of other key information such as visibility or reachability for non-observable objects: the physics-based reasoner makes it possible for the robot to continue to estimate the visibility or reachability of an object by its human partner even if it is no more visible to the robot.

5.1.1 Motivation

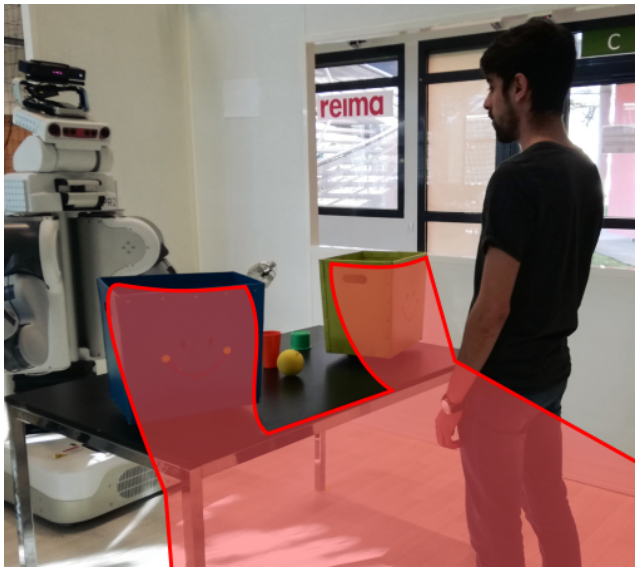


Figure 5.1: Our experimental setup setup with a PR2 robot. From the point of view of the robot, a part of the scene is occluded (colored in red in the picture).

This component's motivation came from the fact that objects extracted from perception are not consistent due to errors in the scene's perception or occlusions. Also, the importance of physical reasoning in infants and the recent studies suggest that human minds use processes similar to physics engines to predict the near future ([Battaglia 2013]) motivated us to study this topic. In Figure 5.1, we see that part of the scene is occluded to the robot that only has a Kinect2 mounted on his head. Besides, the containers are opaque, and the robot cannot see what is inside.

Figure 5.2: The red circle indicates where the Kinect2 RGBD sensor is positioned, and the red rectangle indicates the camera view of the sensor. The orange circle indicates where interaction takes place, and the blue circle where the human is placed.

5.1.2 Inspiration

Regarding the state-of-the-art in developmental psychology and the experiences of [Battaglia 2013] it was clear that physics engines were a key component to consider in cognitive and interactive architectures.

For the development of this component, the main inspiration is the two-system model. As such, the aim is to build a physically anchored perception pipeline with two sub-systems:

- The perception system (similar to the Object File system)
- The physics engine (similar to the Physical Reasoning system)

These two systems run and maintain in parallel two different versions of the world state. The perception system extracts the objects and features without being physically consistent, while the physics engine uses the features of the object (shape, location) from the perception to create and maintain a world state where the following principles are applied: *gravity and solidity, and contact.*

This component's particularity is that instead of merely applying the gravity with the physics engine to the perception data. It continuously monitors the divergence between the perceived world state and the simulated world state maintained in parallel to *trigger inconsistencies.*

At each time, a monitoring algorithm chooses between the perceptual data or the simulated data, providing an output world state that (1) is consistent regarding the physical laws.

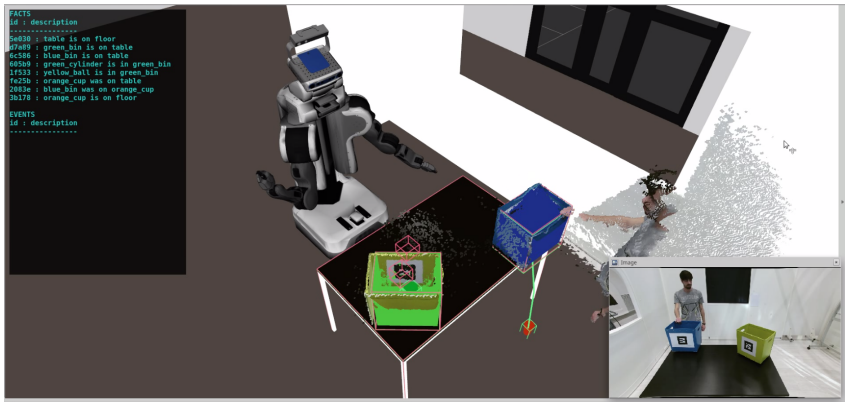


Figure 5.3: Example on inference provided by the system, here the bin is being pushed forward, the cup was not visible from the robot, but still solidity principle remain (objects cannot overlap). On the right bottom is the view of the robot. The physical reasoner infers the movement of the hidden cup that falls on the ground, thanks to the collision models of the box and the physics engine. Then allocentric relations are computed based on the corrected state (on the left). For the complete sequence see Fig 5.6a.

5.2 Implementation

5.2.1 Predicates used

In order to build a physically plausible estimation of the scene and infer actions, this component computes at each reasoning step two predicates for each object:

- $isPerceived(object)$ true when the object was recently seen by the robot.
- $isPhysicallyPlausible(object)$ true when the object is in a stable configuration with respect to the scene.

Based on these predicates, the system is able to infer the following actions (to explain physical inconsistencies) :

- $Pick(object)$ when an object is picked up
- $Place(object)$ when an object is placed on a surface
- $Release(object)$ when an object is not held anymore

Then it computes allocentric spatial relations, based on a physically plausible estimation of the scene (see section 5.2.6):

- $isIn(object, object)$ true when an object contains another object
- $isOnTop(object, object)$ true when an object lies on a surface

5.2.2 Stability reasoning

To estimate whether an object is in a stable configuration, we execute several simulation steps in the future as fast as possible. The number of steps to execute is given by:

$$n_{steps} = P_{horizon}/S_{step} \tag{1}$$

Where $P_{horizon}$ is the prediction horizon and S_{step} the duration that one step simulates. The computation time of the simulation relies heavily on a trade-off between the simulation step (which needs to be small to prevent missed collisions) and the prediction time (which needs to be long enough to e.g. give time to objects to fall).

To know if an object is in a *physically plausible state* (the value of the predicate $isPhysicallyPlausible$), we monitor the divergence in position between the position of the item at the end of the simulated steps and the perceived position ($\overrightarrow{d_{sim/perc}}$) (see Alg. 1).

This approach is similar to the one used by [Mösenlechner 2013] to know if an object will be at a stable state. In our case, we evaluate the stability (or physical plausibility) for each object at each simulation step because we need to avoid as much as possible disturbances in the simulation scene caused by the objects falling during the first steps of the process (before being considered as not stable).

As soon as we consider an object at a not plausible state, we override the simulated object's position and velocity with the perception data and trigger an inconsistency detection, which is then resolved by inferring actions.

5.2.3 Physical Monitoring

To generate the output scene (see Alg. 2), we apply the following reasoning: if the object is in a physically plausible configuration at the end of the simulation steps, we use the resulting computed pose; otherwise, use the perceived pose. In that case, and if the object is left in a physically implausible state, we seek to explain the inconsistency by looking for a human action that would explain the state.

Besides, when an object's position jumps out (i.e., its simulated displacement $\overrightarrow{d_{perc/prev}}$ is more significant than a threshold $D^{max}_{perc/prev}$), we move *contained* objects as well if any.

5.2.4 Action detection

The inference of human actions builds on the assumption that human manipulating objects cause physical inconsistencies. Specifically, when an item is perceived as being in a non-plausible state, we apply the heuristics described in Fig. 5.4. This leads to a robust algorithm for object-oriented actions like pick, place, or release as soon as we can efficiently detect the objects.

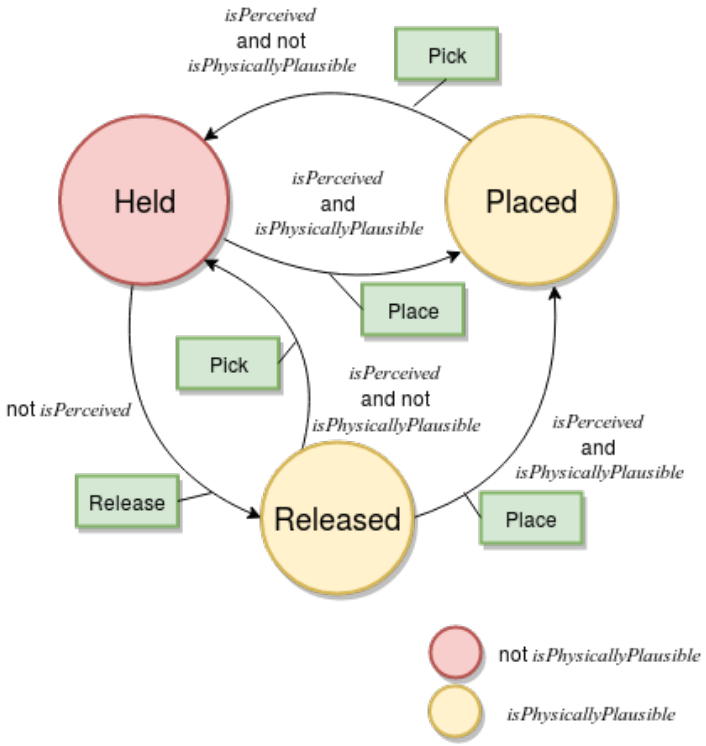


Figure 5.4: The state-machine used to infer human actions based on the object state. The consistency is checked by monitoring the distance between what is actually perceived and the internal simulation of the robot (see Alg. 1)

5.2.5 Output computation

To generate the output scene (see Alg. 2), we apply the following reasoning: if the object is in a physically plausible configuration at the end of the simulation steps, we use the resulting computed pose; otherwise, use the perceived pose. In that case, and if the object is left in a physically implausible state, we seek to explain the inconsistency by looking for a human action that would explain the state.

In addition, when an object's position jumps out (i.e. its simulated displacement $\overrightarrow{d_{perc/prev}}$ is greater than a threshold $D^{max}_{perc/prev}$), we also move *contained* objects as well, if any.

The inference of human actions builds on the assumption that physical inconsistencies are caused by humans manipulating objects. Specifically, when an object is perceived as being in a non-plausible state, we apply the heuristics described in Fig. 5.4.

5.2.6 Support and contents computation

Since the simulation engine corrects the bounding object boxes: the meshes that overlap are popped up, and the floating objects are placed on their support. We can compute the contents and placement support relations with an efficient classic

Algorithm 1 Physical plausibility computation

for n_{steps} (see Eq. 1) **do**
  step simulation for S_{step} seconds
  for all *object* in the input scene graph **do**
    if $\overrightarrow{d_{sim/perc}}(object) > D^{max}_{sim/perc}$ **then**
      $isPhysicallyPlausible(object) = false$
    end if
    if not $isPhysicallyPlausible(object)$ **then**
      override *object* simulation with perceived data
      for all $object_{contained}$ in *object* contents **do**
        move $object_{contained}$
        simulation of $\overrightarrow{d_{sim/perc}}(object)$
      end for
    end if
  end for
end for

Algorithm 2 Output scene computation

for all *object* in the input scene graph **do**
  if object last observation $< T^{max}_{perceived}$ **then**
    $isPerceived(object) = true$
  else
    $isPerceived(object) = false$
  end if
  if $isPerceived(object) = true$ **then**
    Place object where perceived
    for all $object_{contained}$ in *object* contents **do**
      if $\overrightarrow{d_{perc/prev}}(object) > D^{max}_{perc/prev}$ **then**
        move $object_{contained}$ by $\overrightarrow{d_{perc/prev}}(object)$
      end if
    end for
  end if
end for
Update $isPhysicallyPlausible$ (Alg. 1)
for all *objects* in input scene graph **do**
  if $isPerceived(object)$ and not $isPhysicallyPlausible(object)$ **then**
    Set *object* to perceived pose
  else
    Set *object* to simulated pose
  end if
end for
Compute $isOnTop$ and $isIn$ on a physically plausible world (see Section 5.2.6)

approach based on 3D world bounding boxes tests as used in [Sisbot 2011b].

If an inconsistency is generated by the perception (one mesh perceived inside another), the simulation engine will correct it, thanks to the penetration and collision tests performed by Bullet. Because of that, we can use simple and efficient bounding boxes test to compute allocentric predicates (see Fig 5.3).

5.2.7 Parameters

Table 5.1 lists the reasoner parameters used in our experiment. In order to have correct behaviour, the prediction horizon needs to be long enough to make the objects fall while been short enough to speed up the reasoning process. These values depend on the CPU/GPU combination used, and, combined with the *de facto* non-deterministic behaviour of Bullet's collision detection, our results might not be precisely reproducible.

Table 5.1: Reasoner parameters used in the experiment

Parameter	Value	Description
$T^{max}_{perceived}$	$0.7[s]$	Perceived max duration
$D^{max}_{sim/perc}$	$0.045[m]$	Simulation tolerance
$D^{max}_{perc/prev}$	$0.032[m]$	Perception tolerance
$P_{horizon}$	$0.08[s]$	Prediction horizon
S_{step}	$0.00416[s]$	Simulation step

5.2.8 Reasoning pipeline

The input of our *physics_reasoner* component is a ROS URDF file and a 6D pose tracker for objects of interest. To generalize to any perception algorithm, Underworlds uses a special kind of client called *providers* which bring the scene data into the system (e.g. convert the 6D object pose into a node of the scene graph, convert the object model to a 3D bounding box or a 3D mesh, or bind an event from an external reasoner into the timeline).

In [Sallami 2019], we rely on simple perception algorithms to put the focus on the underlying concepts. As such, perception is simplified by using either object with AR tags or items whose unique colour can be used to cluster and segment an RGBD point cloud (in that latter case, the detected objects have a fixed orientation).

Have one provider per modality (allowing for asynchronous updates between modalities), which outputs a simple world (containing only a few objects). The *world_merger* node asynchronously merge these worlds into a single *merged* world.

Finally the *physics_reasoner* is triggered after each update of the *merged* world, to correct object poses, infer out-of-sight objects poses and human's actions. This results in a final, stabilized, world called *merged_stable* (see Fig. 5.5).

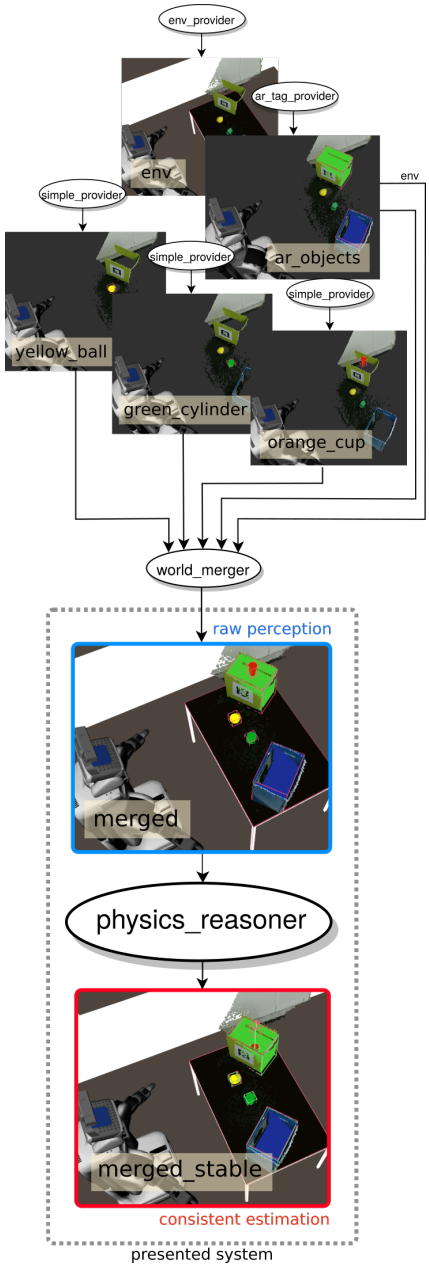


Figure 5.5: The reasoning pipeline build with UNDERWORLDS

5.3 Results

5.3.1 Experimental setup

For this study, we use a PR2 robot with a Kinect 2 placed on its head as the only camera input. The head is static and directed towards the table, where the interaction takes place. From the point of view of the robot, the boxes and the table occlude part of the scene (see Fig. 5.2). We have used this configuration as it is a

classical setup for tabletop human-robot collaboration. The perception algorithms run on a laptop computer to which the RGBD sensor was directly plugged, and the reasoning pipeline runs on a desktop computer with a NVIDIA Quadro K1200 as a graphic card. Here the aim is to have the perception pipeline and the reasoning pipeline running on their own nodelet manager.

5.3.2 Challenging inferences

Fig. 5.6 presents different qualitative results for challenging interactions. In these use cases, reasoning about physics and gravity is crucial to correctly infer overtime the positions and velocities of the objects. These situations have been chosen because they could occur in a classic tabletop human-robot interaction setting. In such dynamic situations, the physics reasoner demonstrates that it can correct the scene geometry and maintain a symbolic state of the environment robustly and estimate the object poses even if entirely hidden. A presentation video of the results featuring a complete sequence is available online[1].

5.4 Conclusion

We have presented work on a simulation-based physics reasoner integrated into a situation-assessment framework called UNDERWORLDS and illustrated how it could be already used to provide a more physically plausible world state in human-robot interaction context, as it can deal with sensory data inaccuracies and potential inconsistencies between different sensor sources by correcting objects poses. It can also estimate the effects of the perceived object motions on completely hidden objects while inferring the human partner's actions.

This component's inspiration comes from physical reasoning in infants, and therefore, we created a physics-aware object permanence for the robot.

As far as we know, we were the first to integrate such simulation-based reasoning in an online manner. This can be explained by the recent progress made by real-time physics engines that allow us to use it online without suffering too much latency in the collision checking process.

We discuss some limitations of the system in its current state as well as future work. First, we introduce how we could enhance the simulation engine with stochastic reasoning, and secondly, we discuss the future steps of the reasoning pipeline.

In [Sallami 2019], we presented different aspects to continue to work. Since that publication, several works have been done in this direction and will be described in the following sections.

[1] https://www.youtube.com/watch?v=f0uqYQzNLYc

5.4.1 Novel work

Enhancing tracking algorithms The reasoning pipeline presented above takes as input correctly identified and localized objects: we assume that the low-level perception pipeline takes care of performing the appropriate tracking and filtering stages, and the physics reasoner does not concern itself with dealing with e.g. noisy perception, as this would be the role of the filters.

Combined tracking and filtering are typically performed with a Kalman filter, which considers the motion of the objects and noisy observations to predict a more accurate object position and velocity. However, the filter has no information about the physics of the scene, and more importantly, whenever the object is occluded, the filter cannot update its motion model. As our pipeline can infer the pose of out-of-sight objects, it would seem that a natural extension of the existing low-level tracking algorithm could benefit from a simulation 'feedback' from the physics reasoner to update the motion models of all the objects even the occluded ones. Such a physics-aware Kalman filter would lead to enhanced object tracking while smoothing the physical reasoner's object motion. This approach relies on a close interaction loop between physics reasoning and low-level perception.

Since the development of a complete perception stack adapted to our needs (presented in the chapter 6), we were able to implement a physics-aware Kalman filter by updating the 3D Kalman filter with the pose of the objects from the physical simulation.

Pipeline for unknown objects One of the critics of this work is the fact that we used simple visual primitives for objects. With the recent implementation of a more complex perception system involving neural networks for object detection and tracking-by-detection MOT, we have extended this work to unknown objects. We approximate the object's shapes by aligned bounding cylinders or spheres using the object class and intrinsic parameters of the robot camera and the depth map provided by the RGBD sensor. The object's mass is then computed by taking into account the volume of the shape and giving it the water density. This approximation is sufficient in our context with small manipulable objects and benefits from physical reasoning for unknown objects. More details about the perception system are given in the chapter 6, and we use the physics simulation and CNN based detectors in chapter 7. However, because we reason on rigid objects, we still use AR tags for containers as we need a more precise 3D model for them.

Exploiting human model for action detection In classical action detection tasks, the aim is not only to know, for example, that an object has been picked but also who picked it and to generate a triplet $< subject, action, object >$. Classically, these two tasks are handled together by jointly classifying human temporal motion with respect to objects.

In the current implementation, the human is now detected by SSD detectors and being tracked over time. When an inconsistency is detected, we assign the action

to the object's closest person using a classic Hungarian assignment algorithm with 3D distance. By doing so, we now generate the events $< agent, action, object >$, and we make available that information as an UNDERWORLDS situation to higher-level reasoning. In our current implementation, we can assign actions and report an inconsistency if no human is in the vicinity.

However, because of the nature of the algorithm that analyses physical inconsistency and detects action, we were unable in its current form to solve the limitations presented in the qualitative results. Indeed, because we analyze the human actions as a perturbation in the scene's physics, if we constraint the object's position when picked, it would not be possible to detect when the objects are released. Unless reasoning on containers (if the objects are released above a container, we disconnect them), but in that case, it would be impossible to infer that the person released the objects on the floor, for example. More work could be done in this direction in particular. Combining our approach with a more traditional classification of human motion may be the way to go.

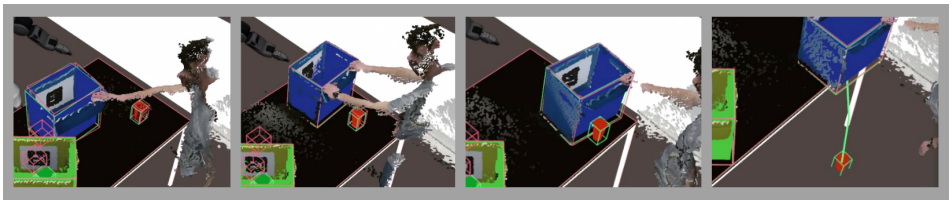
5.4.2 Future work

This section presents future work that we did not explore yet, mainly because the performances did not convince us to use neural simulators. Despite promising results, real-time performances are not clear enough. However, as the technology will be mature, it will be straightforward to change it because they possess object-level representation.

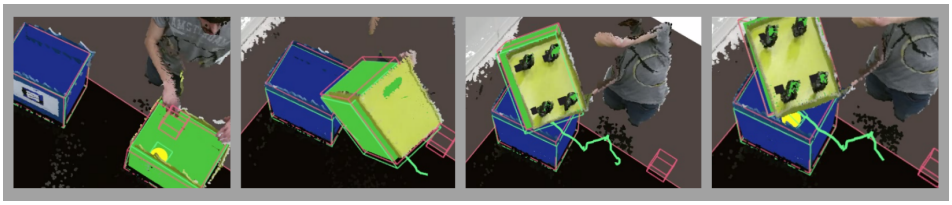
Uncertainty and simulation engine limits

Bullet does not handle uncertainty, and consequently, we cannot output a precise value for the covariances of Underworlds. However, we update the 3D Kalman filters with the simulated pose, which allows us to "mimic" a probabilistic output of the simulator by considering the simulation output as a noisy input that we integrate with perception observations.

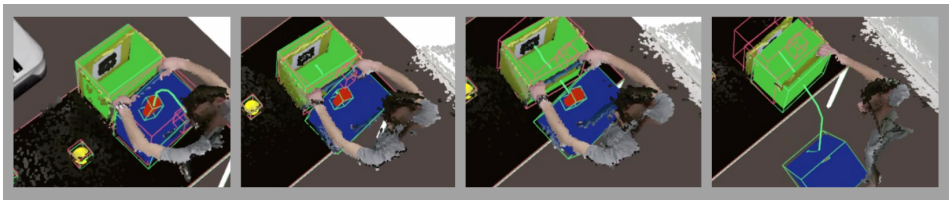
The current state of the art in probabilistic physics engine[Battaglia 2016] is not clear about real-time constraints making the use of an analytical simulator preferable. However, because we use an object-level representation, it would be easy to shift between an analytical simulator and a stochastic one. The algorithm we designed uses a distance metric that can be applied to any form of the object's internal state as long as we can formalize a likelihood metric. In essence, the algorithm we use mimics an infant's physical reasoning system by checking in the background that the internal physical models and the reality (when observed) match in their behaviour and, if not, trigger an inconsistency.



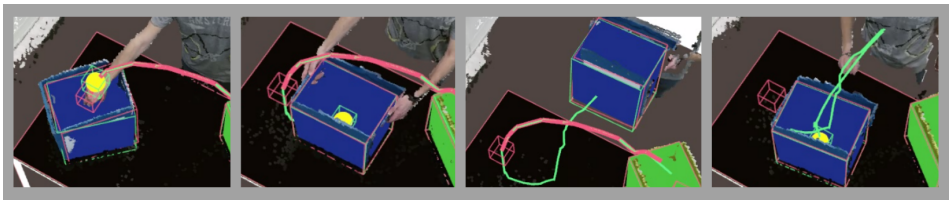
(a) In this example, the robot first perceives the blue box and the orange cup. The human moves the box that hides the cup, and pulls it back until the cup falls. Even though the robot did not see the cup falling, the reasoner infers that it is on the floor.



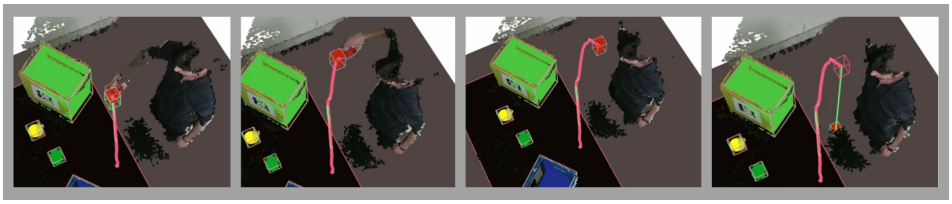
(b) In this example, the yellow ball is known to be in the green box. The human empties the green box into the blue one. The reasoner successfully infers that the ball is now inside the blue container, without ever seeing the ball.



(c) In this example, the yellow ball is known to be in the green box. The human empties the green box into the blue one. The reasoner successfully infers that the ball is now inside the blue container, without ever seeing the ball.



(d) In this example, the reasoner successfully estimates the movement of the yellow ball, while inside the blue box.



(e) This last example illustrates one of the limitations of the system: the human intentionally hides the cup behind himself; the reasoner fails to infer that it is still held, and instead computes that the cup must have fallen to the floor.

Figure 5.6: Examples of challenging inferences only possible with the use of the physical reasoning. Only the RGBD sensor mounted on the PR2 head is used. The pink trajectories represent the observed trajectories, with the corresponding bounding boxes in pink; the green trajectories are those computed by the physics reasoner (and accordingly, the green bounding boxes).

Chapter 6

Uwds3-HRI: A library of reasoner

Contents

In this chapter, we introduce the building of a basic library that was developed to ease the design and implementation of HRI scenarios. That library uses Underworlds data types to be natively compatible with the framework and provide ready to use components that benefit from Underworlds modularity.

6.1 Introduction

Underworlds have been created to connect perception components and higher-level reasoning while providing relations/events detection and consistent scene estimation. And at the beginning of this thesis, we used it as such.

However, we decided to build a library that combines computer vision with geometric reasoning that fit our purpose in the last few years. Because when just binding software (by converting the data into Underworlds format) from people who do not have the same constraints and goal in robotics, we noticed that we were losing in the possibilities offered by nowadays computer vision algorithm.

For example, we may prefer stability over accuracy in some context or link recognition and supervision by triggering an event when a new individual appears.

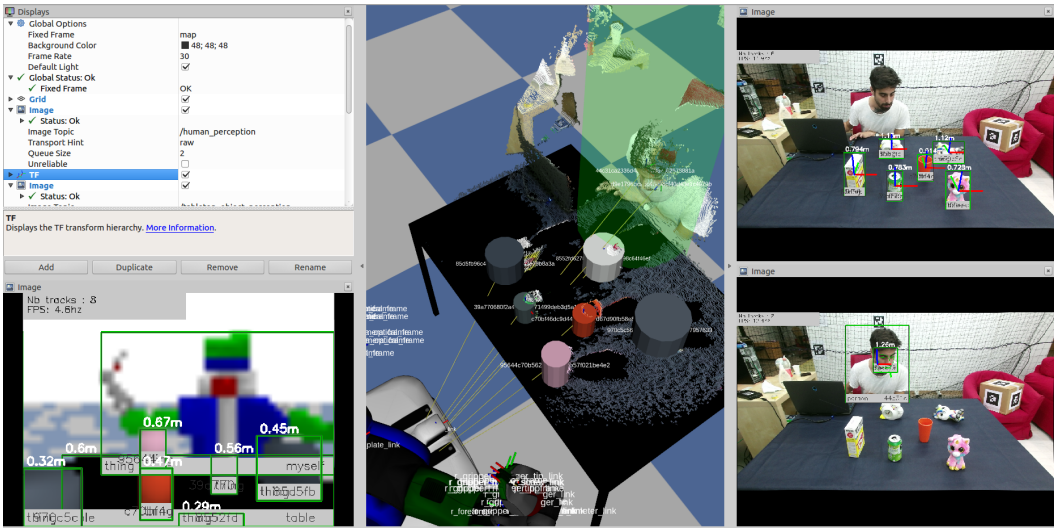


Figure 6.1: Markerless physics-aware level 2 perspective taking for unknown objects. Objects are detected with gaussian mixture models, humans with neural networks. These information are used to build a 3D model that is corrected by the physics engine. The perspective of the human partners is computed from the 3D model generated in a online manner.

6.1.1 Practical challenges for HRI

Many useful neural networks have been developed to fit a challenge or a dataset. They sometimes needed intensive work to make it usable in a robotic context. For example, the batch inference could be challenging to perform or not accessible from the given API.

One other common problem is that computer vision challenges (and software) are often specialized in one task, so the best state-of-the-art techniques are often designed to use the GPU entirely and use the best quality ones to perform best on the benchmarks.

When we deal with humans and objects, we need to reason on two types of entities, making it more difficult as we need twice the power of a good GPU.

One workaround is to use a cloud platform, which explains the recent interest in this area. However, it raises security and privacy issues, and when doing experiments with people, these questions are essential.

ROS middleware has been designed to use multiple computers in a decentralized fashion by using a server that manages a subscribing/publishing communication scheme (which we exploited fully in this work). Here, the local network is the limitation and could be a problem if one tries to send images over many computers.

We also use a physics engine that performs collision tests and rendering on top of that classical issues that explain the care we had to have on real-time performances.

6.1.2 Motivation

During the project MuMMER, we faced many integration issues, mainly because the perception stack developed was not emphasizing the aspects that we needed in our context. A good result in computer vision does not necessarily mean that it will be usable in a robotic context. For example, one error in the face recognition algorithm can easily result in a failing interaction at the supervision level. Another example is the perspective-taking computation that needs a stable head pose estimation, even if less accurate.

This makes us realize that it was worthing to have a dedicated perceptual system specially designed to be integrated with situation assessment. Moreover, the cascading nature of the pipelines that UNDERWORLDS builds makes it similar to how perception algorithms are implemented. As such, our system can be seen as an extension of a classical perception stack.

With that library, we were able to work on physical reasoning for unknown objects and the link between the tracking system and physical reasoning.

6.2 Implementation

The library have been implemented in Python with the help of *numpy*, *scipy* and *opencv* and *dlib*. This choice has been made because of the rapidity in prototyping and could benefit from a C++ API as these libraries have a C++ version.

6.2.1 Detection

To detect people or objects, we decided to mainly use deep learning approaches because of their impressive results in image detection. However, there are still challenges in this area: state-of-the-art detection requires a certain amount of data to be effective. For human face or human body detection, the publicly available data is sufficient to use pre-trained weight.

However, for objects, the data available online is insufficient to cover all the diversity of everyday or specialized objects (working tools or special items for an HRI task). The more commonly used dataset for object detections, MSCOCO ([Lin 2014]), cannot represent all the possibilities as we want to enhance the robot knowledge during the interaction.

These types of scenarios only start to be tackled by the computer vision community and remain challenging. In [Bansal 2018] they introduce only the last years, the problem of zero-shot detection, which consists of detecting objects that have never been seen during training.

To be able to work on a wide variety of objects, we developed a static foreground detector based on two gaussian mixture models that model respectively motion and foreground at pixel-level (see Fig6.3). This detector can accurately detect any object in the scene different from the background (usually a table where the interaction takes place). However, this technique's main limitation is that it needs a fixed sensor

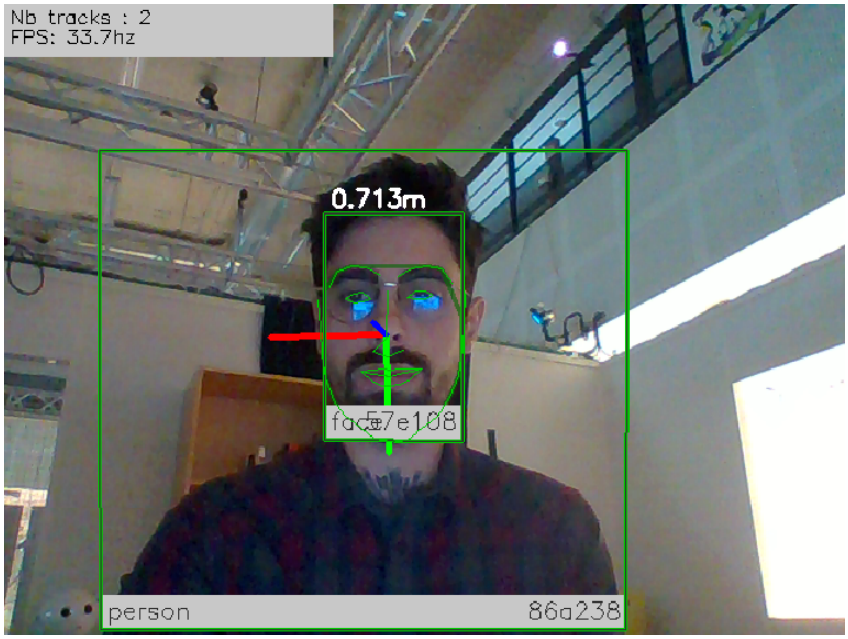


Figure 6.2: The human perception avaible with the framework to perform perspective taking. Note that the axis convention is Z-forward (blue axis), in order to be used by OpenGL rendering. The facial landmarks used to compute the head orientation are made available as an aditionnal feature associated with the face.

camera to work and can be used only in specific scenarios or to record training data about new objects in a weakly supervised fashion.

6.2.1.1 Human detection

Because people are not always facing the camera, we choose to use a face detector and a body detector combined. For body detection, we used an SSD with Mobilenet backbone trained on MSCOCO, and for face detection, we used the HOG detector alternatively from Dlib (because common available facial recognition is based on it) or the OpenCV SSD face detector when needed more robustness in the face detection. These detectors have been chosen because they can perform real-time inferences.

In the near future, we plan to integrate [Zhang 2016] work to jointly infer face detection and facial landmarks for alignment and head pose. Also, we plan to integrate the work of [Fischer 2018] for gaze estimation.

6.2.1.2 Object detection

To be able to extract objects, different perception modalities have been implemented into different Provider to fit multiple HRI scenario:

- Color detection for toy scenario

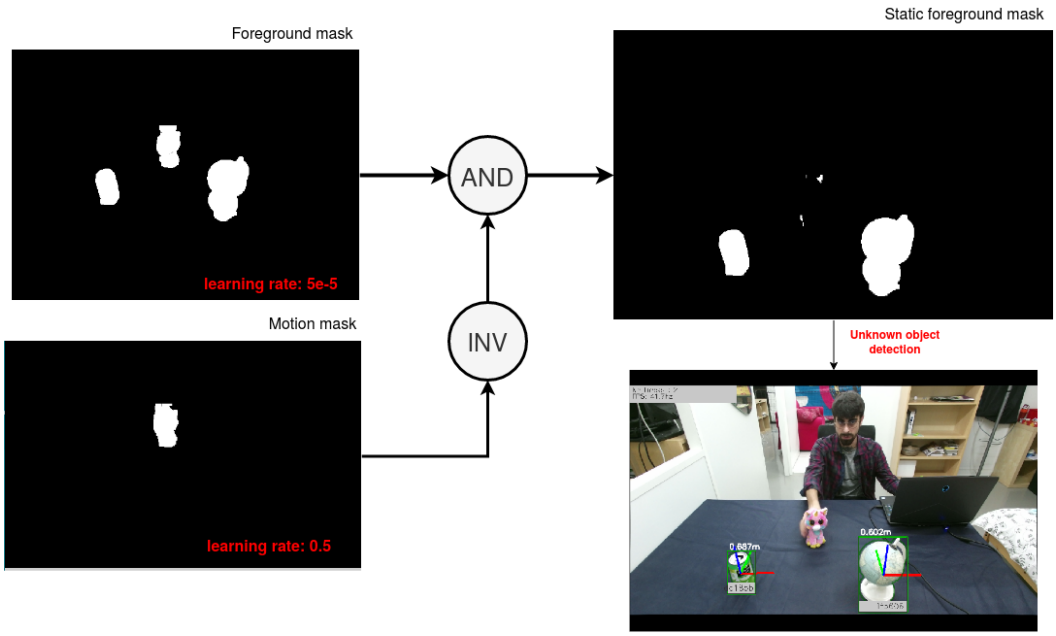


Figure 6.3: By combining Gaussian mixtures models with different learning rate parameters, we can detect completely unknown objects (never been trained on). We chose to remove moving entities because the moving hands interfere with the foreground detector when interacting with the objects. As soon as an object is detected, if it moves, we rely on the single objects trackers (Medianflow and 2D Kalman) that keep track of the objects. This algorithm outputs a mask and the bounding boxes that can be used to generate training data for the Mask-RCNN detector in a weakly supervised fashion.

- SSD detector with mobilenet backbone pre-trained weigths on MSCOCO
- Mask-RCNN detector with inception backbone pre-trained on MSCOCO
- AR tags detectors for specific HRI scenario
- Static foreground detector using gaussian mixtures (See Fig. 6.3)

The foreground detector has been very promising as it can detect objects accurately without any training (but needs a fixed camera). The detected objects can then be used to generate datasets for custom objects by interacting with the robot.

For known objects, we rely on AR tags or colour detection (for objects without texture). Furthermore, for unknown items, if the camera is fixed, we use the foreground detector. Otherwise, we rely on the objects classes in the pre-trained SSD and Mask-RCNN trained on MSCOCO to perform object detection.

6.2.2 Tracking

Tracking is done using the classical approach of tracking-by-detection. We assign the detections to the tracks by first using the Hungarian algorithm with an

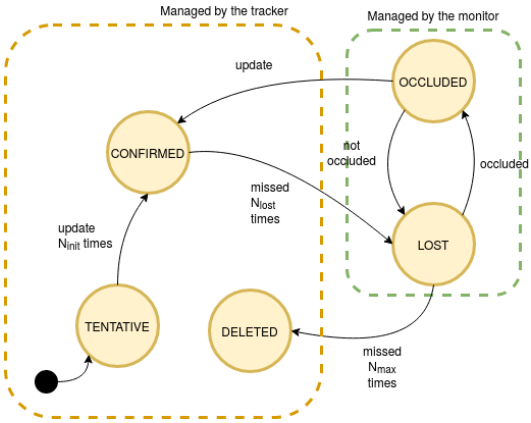


Figure 6.4: The state of the scene nodes are updated by the tracker and the physics monitor. With N_{init} the minimum number of observations before being considered confirmed, N_{lost} the maximum number of missed observations before considered lost, and $Nmax$ the maximum number of missed observations before deleting the node. Both the tracker and the monitor manage the state of the scene nodes.

intersection-over-union (IOU) metric. Then we use a centroid or deep feature based assignment to the remaining detections depending on the cases. This tracker uses a Medianflow and Kalman single object tracker to make it possible to use the detection algorithm, not on every frame, making it a more efficient overall pipeline.

The particularity of the tracker implemented is that it integrates a LOST state when the track is not detected anymore, which is different from the OCCLUDED state (See Fig 6.4). The OCCLUDED state can be set only by the monitors that have access to the whole physical context and explain if the object is lost because another object occludes it or not.

6.2.3 Features extraction

In our context, the objects can be out of view. Therefore, we need to extract and store ourselves the features needed to infer the object's properties because traditional perception pipelines only work on visible objects. For that purpose, we extract the face's facial landmarks and the appearance features using a pre-trained CNN on ImageNet for the objects. In order to provide semantic features (a vector that encapsulates semantic meaning into a geometric space) to the nodes, we compute the word embeddings associated with the node's description using *fasttext*[1] the fastest way to compute word embeddings for unknown words by using n-gram embeddings. For facial features, we use *openface*[2]([Amos 2016]) embeddings that are used by the facial recognition combined with a KNN to identify people known and unknown people. A summary of the visual features extracted is presented in Table 6.1.

[1]https://fasttext.cc/
[2]http://cmusatyalab.github.io/openface/

The choice for the features has been made to fulfil different constraints: (1) extendability of the labels because the robot needs to learn on the go (2) being able to understand social signals (3) offer the possibility to learn relevant visual knowledge that can be coupled with the ontology (material, colour, shape).

We choose to use static word embedding for the labels always to have a fixed size vector (300 in this case) that is more compact than using the one-hot encoding (one bit per label), which means that the robot can learn more than 300 objects labels without augmenting the feature's size and allowing to use this input into other processes. Moreover, this enables us to reason on the labels semantic by integrating knowledge from unsupervised learning.

We choose to have the facial landmarks to reason on emotions and head pose to interpret social cues. We decided not to use an emotion classifier to benefit from the complete information from the facial landmarks to interpret emotions not as happy, sad or neutral but more to detect social cues related to the joint task.

Finally, we decided to integrate an appearance feature. Currently, it is just the pre-trained (on ImageNet) features extracted with a CNN backbone. In the future, this feature could benefit from siamese networks trained for our task (classifying colour, shape and material) to be reduced in size.

We believe that leveraging neural graph networks that can capture the relation between entities (provided by our software) and these features could lead to interesting future research in context understanding for human-robot joint action.

Table 6.1: The additionnal features added to the nodes geometric information (extracted from visual data). Note that the features related to faces are only extracted from detected faces.

features name	**dimension**	**comment**	**Usage**
color	180	The hue histogram	simple color recognition
facial landmarks	68	The *dlib* facial landmarks	head pose estimation and emotion recognition
facial description	128	The facial features extracted from *openface*	facial recognition
appearance	2048	The deep features extracted from a ResNet50 network trained on ImageNet	multi purpose visual property recognition
semantic	300	The word embedding of the object label	used to benefit from the label inferred by object recognition in other tasks

6.2.4 Monitoring

The monitors are the component that analyses over time the entities present in the scene. They integrate the logic needed to make inferences about the entities relations, which could be machine states, petri-nets, HMMs or POMDPs. In our case, we mainly use simple state machines by using a threshold when dealing with probabilistic input or distances.

The monitors are generally task-dependent, even if some are more general than others depending on the task envisioned in an HRI context. Several monitoring has been implemented during this thesis: physical monitoring for manipulation scenario (see chapter 5), engagement monitoring for dialogue scenario (see chapter 4), and perspective monitoring (see chapter 7).

6.2.5 One-shot object recognition

To create a custom object dataset, we developed a dataset generator that benefits from the foreground detector to be able to show to the robot completely unknown objects and train object recognition networks to work on object property recognition (see Fig. 6.5). This aspect is not finished yet (see future work).

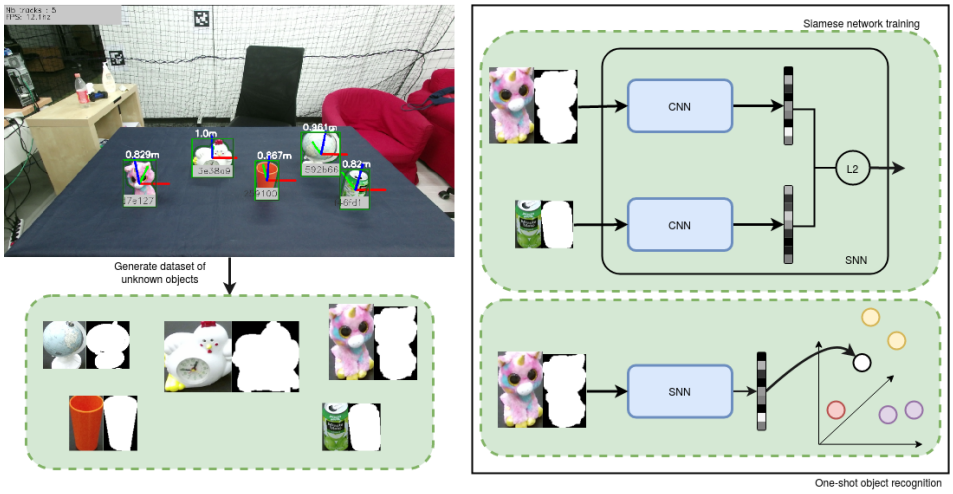


Figure 6.5: Pipeline for one-shot object recognition. First semantically rich objects are extracted by the forground detector, then a SNN (composed by a backbone trained on imagenet and a fully connected layer with sigmoid activation) is trained using a constrastive loss. A inference time, a KNN is used in order to infer new labels at runtime.

6.3 Conclusion

In this chapter, we introduced a library that is built to be naturally compatible with the framework UNDERWORLDS. This work has been done to work on closer

interaction between the perception system and benefit from recent deep learning progress on the representation problem. Indeed we think that a better interaction with the higher-level reasoning and low-level perceptual systems can lead to exciting work for human-robot interactions. In the following section, we discuss future work that could be interesting to do in this direction.

6.3.1 Future work

In the following paragraphs, we present future developments of this library and discuss the potential future use of this work. First, we explore the topic of learning objects properties by interacting with a robot.

Learning object properties by interacting Such reasoning is made possible by mapping deep representations into a euclidean space with the help of Siamese Neural Networks, allowing the robot to detect when a new property is presented and generalize better to new objects.

The robot needs to detect objects to extract visual features and detect that the property is unknown using a KNN approach to ask questions about the property. For example: "What colour is that object?" "What is its purpose?" without any previous training.

Object and property recognition has been part of the overall design; however, we did not have the time to fully develop this aspect (the integration with supervision in particular). However, we implemented all the tools: weakly supervised recording scheme based on the foreground detector, deep features extractor, siamese networks using classic transfer learning approach, KNN inference, and the link with the ontology (see chapter 7). SNN has been used intensively in the last years, and there is no novelty in using them. However, we think a clever integration with the ontology and the supervision system can develop interesting future work in learning from the demonstration.

Enhancing the human model In this work, we keep the human model simple by only detecting the head and body using SSD detectors to run in real-time. In the future, we plan to integrate a 2D pose detector to reason more finely on human-object interactions. Also, providing a robust gaze estimation is planned to enhance the visual perspective-taking capabilities.

Chapter 7

Modern hybrid architecture for embedded cognition

Contents

In order to show the pertinence our the work on physical and geometric reasoning and the modular design of Underworlds, we worked on the integration of the software with higher-level reasoning. In particular, we focus on the integration with ontology-based reasoners.

In this work, we developed a component that binds the situation-assessment representation with the ontology to answer SPARQL queries over the data structure generated from visual inputs.

For that purpose, we designed an architecture for embedded cognition and belief reasoning. This architecture uses CNN-based detectors to extract entities, a physics engine to correct the world state and detect actions, efficient yet straightforward geometric relations detection, and image rendering for perspective-taking level 2 combined with an ontology-based reasoner to be able to query the generated data with SPARQL queries. SPARQL language is a semi-specified query language for RDF graphs (see examples in table 7.5). The ontology can be then modified or queried by using SPARQL queries.

Also, we started preliminary work to illustrate why we need an ontology in modern cognitive architecture and why it is pertinent that tomorrow's robot combines machine learning and formal representation. In this direction, we explored how

to use deep language models to map natural language (NL) over SPARQL queries usually used with ontology-based software. This approach allows us to benefit from logic-based inferences and create question answering systems that reason about the state of the world and do not just generate the most probable answer.

The following chapter will present this ongoing work and discuss the benefit and limits of using deep learning in this context. Also, we introduce the future challenges and opportunities that bring the integration of spatial-temporal and physical reasoning with symbolic logic in the context of a collaboration between humans and robots.

7.1 Introduction

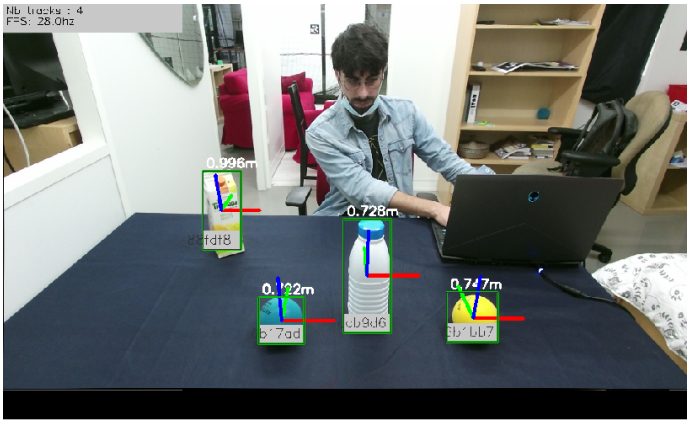
Ontologies are systems that formalize a particular knowledge domain by its concepts, categories, properties, and relations. Ontology-based reasoners can analyze and infer logic-based properties on Ressource Description Framework (RDF) graphs based on the semantic description provided in the ontology. RDF graphs aim to represent triplets of information like our situation assessment software.

This technique has been used successfully in human-robot interaction by modelizing the robot's knowledge and other's knowledge. In [Lemaignan 2012], the authors endowed the robot with part of ToM processes: explicit representation of other's beliefs and the ability to modify or answer that knowledge using multiple independent RDF graphs that represent the robot and the estimated human beliefs.

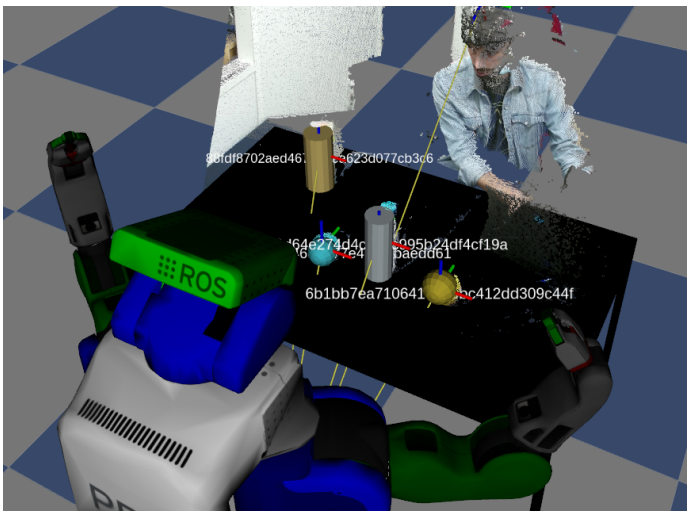
Because UNDERWORLDS already build a sparse knowledge graph with edges representing triplets (See Fig. 7.2), it is naturally compatible with RDF-based reasoners, and the integration is made easy by the similarity of the data structure. Moreover, the RDF can represent abstract concepts and entities, where the situation assessment can only represent Spatio-temporal entities. While the situation assessment reason on the correctness of the world with simulation-based physics, the ontology provided along the RDF graph allows reason on the semantic correctness of the situations and detect eventual inconsistencies that the physics engine did not correct.

7.2 Design

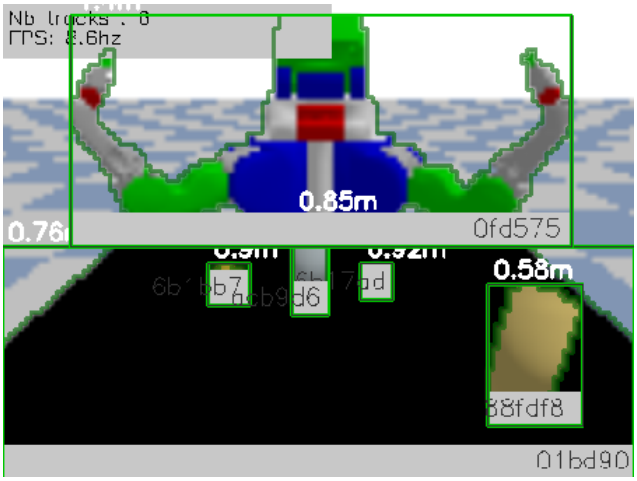
In the following sections, we present the overall aspect we focus on during this component's design. We first start to present the ontology used to model our domain and then present the main advantage of using ontology-based reasoner: SPARQL queries. We then present preliminary work on integrating deep language models that benefit from the SPARQL language's expressiveness to bridge natural language and logic-based inferences.



(a) Objects detected with SSD which gives the object classes and 2D position. The shape is estimated using the class labels (sphere for balls, cylinder otherwise) and 3D position using the depth image of the RGBD sensor.

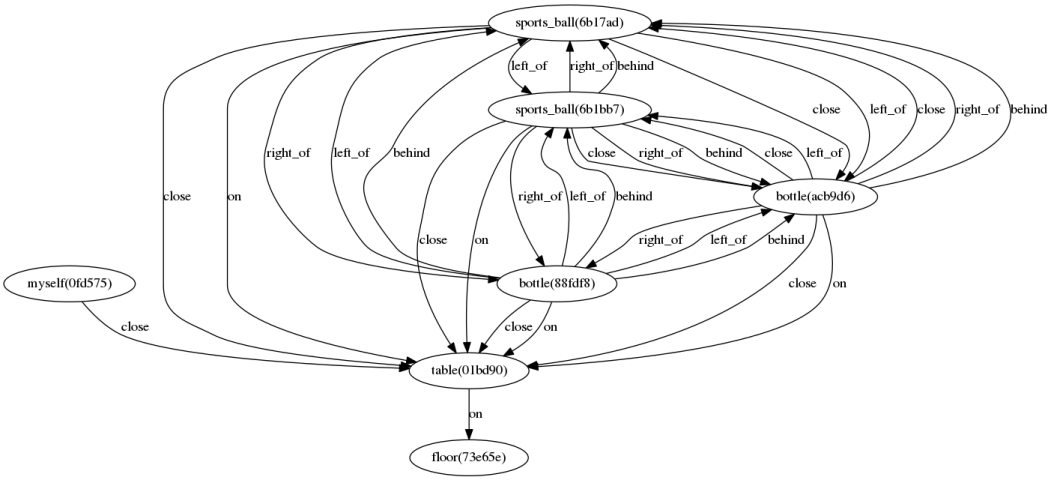


(b) The 3D scene generated by UNDERWORLDS and corrected by the physics engine (objects lie corrrecly on the table).

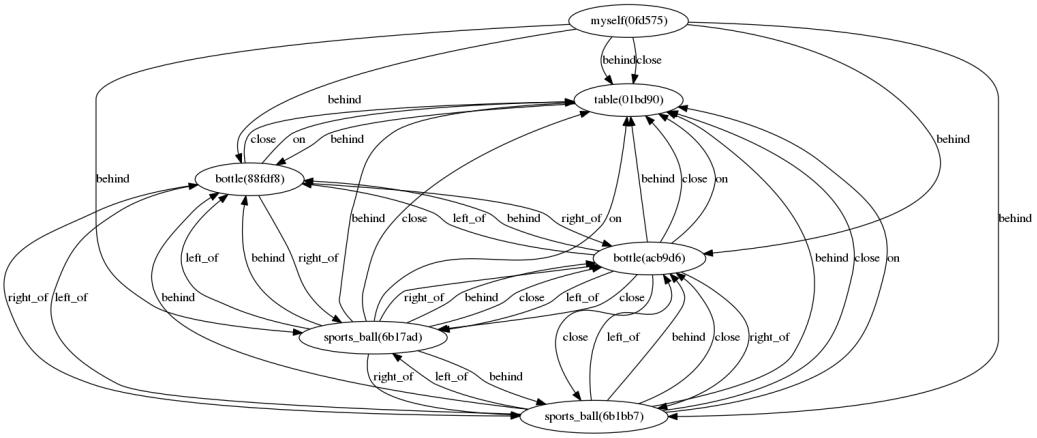


(c) The estimated view of the scene from the human perspective used to compute egocentric relations.

Figure 7.1: An example of geometric scene extracted by UNDERWORLDS. See Fig 7.2 for the corresponding semantic information.



(a) The semantic scene graph that represent the robot beliefs with allocentric relations computed based on the 3D model and egocentric relations computed from 2D geometry in the robot view.



(b) The scene graph that represent the estimated beliefs of the human based on perspective taking level 2. This graph contains the allocentric relations of the visibles entities (view independent thus computed only once) and the egocentric relations computed from the human estimated view.

Figure 7.2: An example of semantically rich scene extracted by UNDERWORLDS.

7.2.1 Data-structure correspondance

Both situation assessment and ontology-based reasoners have a sparse representation of knowledge. This fact makes the binding between both representations straightforward: The scene nodes correspond to RDF individuals and the temporal situations to RDF object properties.

In order to express the semantic of UNDERWORLDS data-structure, we first designed a domain-specific ontology (See Fig. 7.3). For that purpose, we adapted the common-sense ontology to fit our data type description. Individuals with the same ID represent the nodes generated by UNDERWORLDS, then all the knowledge about the node is added based on the information extracted. Additionally, RDF object properties represent the temporal situations, which allow the system to reason on the semantic meaning of relations using SPARQL queries.

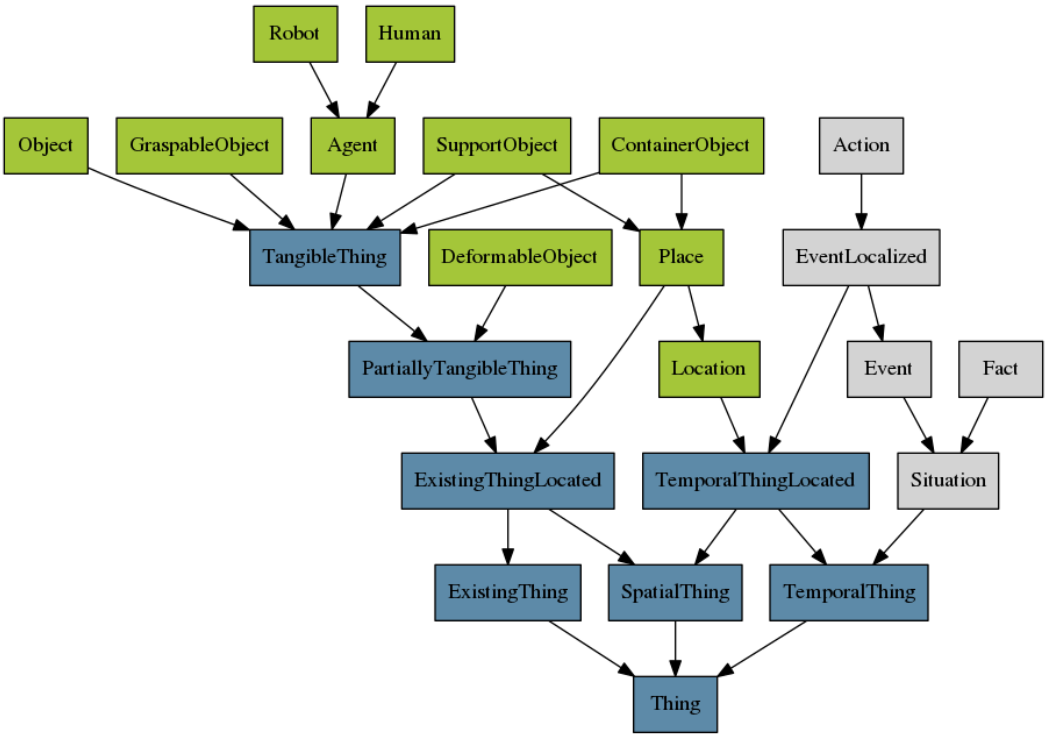


Figure 7.3: The ontology from [Lemaignan 2012] adapted for UNDERWORLDS. In blue are represented the abstract classes, in green the classes used in the underworlds 3D scene graph and in gray the one used in the timeline.

7.2.2 Inferring from physics

To learn affordances from the physical reasoner that we developed in [Sallami 2019] we implemented a simple binding with *oro* that learn the affordances relative to the actions detected by the physical monitoring (pick, place, release). In a nutshell, if an object has something on top of it, we tell the reasoner that this object is a

Table 7.1: The predicates computed by the physical monitoring.

predicates	**input data**	**rule**
isOnTop(Object, SupportObject)	aabb test on physically consistent scene	True when the AABB lie on another, False otherwise.
isInside(Object, ContainerObject)	aabb test on physically consistent scene	True when the AABB is overlaping with another AABB, False otherwise
isClose(Object, Object)	aabb test on physically consistent scene	True when the AABB is closer than 2 times the diagonal, False otherwise
hasInHand(Agent, Object)	actions from the physics reasoner	True an agent pick something, False when the object is released or placed

Table 7.2: The predicates computed in the image (robot and perspective).

predicates	**input data**	**rule**
isRightOf(Object, Object)	2D bbox test	True when the bbox is the right of another bbox (x dimension), False otherwise.
isLeftOf(Object, Object)	2D bbox test	True when the bbox is the left of another bbox (x dimension), False otherwise
isBehind(Object, Object)	relative depth extracted by perspective taking	True the depth relative to the view is greater, False otherwise
isVisible(Agent, Object)	perspective taking	True for all objects extracted by rendering

SupportObject. If an object is picked, we memorize that it is a *GraspableObject*. Furthermore, if an object contains other ones, we tell that it is a *ContainerObject*. Another benefit of using our physically consistent scene is that the relations detected are consistent regarding the scene's physics.

7.2.3 SPARQL queries and neural translater

Answering questions based on the robot's knowledge is fundamental, and the expressiveness of the SPARQL allows us to request the robot's knowledge efficiently and, in our case, also other's knowledge.

Traditionally, when asking a knowledge base in natural language, the query is parsed in order to extract sub-questions into triples patterns like in [Lemaignan 2012]. This technique requires first identifying verbs and nouns to extract triplets but can fail at understanding contextual information at a subtle

Table 7.3: The predicates inferred by the ontology based reasoner.

predicates	**rule**
supports(SupportObject, Object)	inverse of isOnTop
contains(ContainerObject, Object)	inverse of isInside
isHeldBy(Object, Agent)	inverse of hasInHand
inFrontOf(Object, Object)	inverse of isBehind
isUnder(SupporObject, Object)	alias of supports
isAt(Object, Place)	alias for isOnTop, hasInHand and isInside
hasName(Object, String)	alias for rdf label object property

level or merely being robust to the noise of the speech to text algorithms.

Recently deep learning approaches have emerged to handle these problems using sequence-to-sequence (seq2seq - input a sequence and generate a sequence) models where natural language is translated into a query understandable for the knowledge base: a SPARQL query. In [Soru 2017a] they used a Bi-directional LSTM model and start to formalize the problem. In [Soru 2018], they explore how to format the training set to learn the compositionality of the NL queries. They conclude that the network needs to see in the training set composed and simple queries to learn to compose new patterns (in our case, it could be "what objects are green, in the fridge and inside a red box?").

This approach of using deep language models to map natural language over SPARQL queries have never been explored in human-robot interactions, as far as we know. We also think this use case illustrates the synergies between the situation assessment and ontology-based reasoner and our vision of a hybrid architecture that uses neural networks to deal with the input noise (visual or from speech to text) and relies on high-level reasoning on more classic formal approach.

By building a neural language model that maps natural language expressions towards SPARQL queries, one can create at some point (if enough data is provided) a language model that encapsulates in the network's latent representation the comprehension to efficiently answer questions or ground verbal expression relative to the knowledge present in the RDF graphs.

This methodology benefit from the extensibility of a weakly supervised training approach like in [Soru 2017a] or [Soru 2017b] consisting of using manually annotated templates (see Table 7.5) filled with random individuals in order to augment the data available for training. This approach makes it possible to add more data by editing simple files.

source(NL queries)	**target**(SPARQL-like queries)
<Courtesy> give me the <Color> <Object> that is on the <SupportObject>	SELECT ?x WHERE ?x hasColor <Color> . ?y hasName <SupportObject> . ?x isOnTop ?y
<Courtesy> give me the <Color> <Object>	SELECT ?x WHERE ?x isA Object ?x hasName <Object> . ?x hasColor <Color>
<Courtesy> give me the <Object> in the <ContainerObject>	SELECT ?x WHERE ?x hasName <Object> . ?y hasName <ContainerObject> . ?x isInside ?y
<Courtesy> what is on the left of the <Object>	SELECT ?x WHERE ?x isA Object . ?y hasName <Object> . ?x isLeftOf ?y
<Courtesy> what is on the right of the <Object>	SELECT ?x WHERE ?x isA Object . ?y hasName <Object> . ?x isRightOf ?y

Table 7.4: Example of templates used to generate pairs of NL queries with the associated SPARQL queries. The variables tags are then filled with random individuals to generate the dataset. The <sender> and <receiver> tags are replaced at runtime by the ID of the person that speak (which emit the query) and the receiver by the ID of the robot (which receive the query).

source(NL queries)	**target**(SPARQL-like queries)
<Courtesy> what is on the left of the <Object> and close to <FurnitureObject>	SELECT ?x WHERE ?x isA Object . ?y hasName <Object> . ?x isLeftOf ?y . ?z hasName <FurnitureObject>
<Courtesy> what objects are <Color>, in the <ContainerFurniture> and inside a <Color2> <ContainerObject> ?	SELECT ?x WHERE ?x isA Object . ?x hasColor <Color> . ?y hasName <ContainerFurniture> . ?x isInside ?y . ?z hasName <ContainerObject> . ?z hasColor <Color2> . ?x isInside ?z

Table 7.5: Exemple of composed queries used to learn complexe patterns.

7.3 Implementation

7.3.1 Ontology based reasoner

In this work, we used two different ontology-based reasoners: ORO([Lemaignan 2010]) and more recently ONTOLOGENIUS([Sarthou 2019])[1] that have both the particularity of storing multiple RDF graphs that can represent the robot knowledge but also others people knowledge by maintaining alternative RDF graphs. This capability makes that reasoners candidates of choice to be used in conjunction with UNDERWORLDS perspective-taking capabilities.

7.3.2 Underworlds reader

The integration with the overall reasoning pipeline was easy thanks to UNDERWORLDS. We only had to implement a client that subscribes to the physically consistent world state and the human perspective state to communicate with the ontology system. This type of client is called READER as they only subscribe to the data and do not modify the world states. Fig. 7.4 represents the complete architecture integrated with our reasoning pipeline for physical reasoning.

7.3.3 Dataset generation

In order to perform queries in natural language (thus allowing the robot to ground verbal expressions or answer questions), we developed a dataset generator[2] based on template filling (See Table 7.5 for examples). We used the same methodology as [Soru 2017b] by training a seq2seq neural network on an artificially augmented dataset. For our preliminary dataset we have 15 templates that allow us to generate 1540 different data pairs.

7.3.4 Neural translation

In the first stage of this work, we implemented the dataset generation to be compatible for openNMT ([Klein 2017]), a framework for seq2seq models. We tested our model with a Transformer architecture ([Vaswani 2017]). However, recently in [Yin 2019], the authors compare different seq2seq architecture over several datasets for this task and conclude that CNN based encoder-decoder seq2seq models ([Gehring 2016]) perform better than complex Transformer models (The current state-of-the-art in neural translation). For this reason, we are currently migrating from openNMT (which only have Transformer models) to FairSeq ([Ott 2019]) a framework for efficient neural translation, which possesses CNN-based seq2seq architecture in order to verify these results.

[1] https://github.com/LAAS-HRI/uwds3_ontologenius_bridge
[2] https://github.com/LAAS-HRI/sparql_translater.git

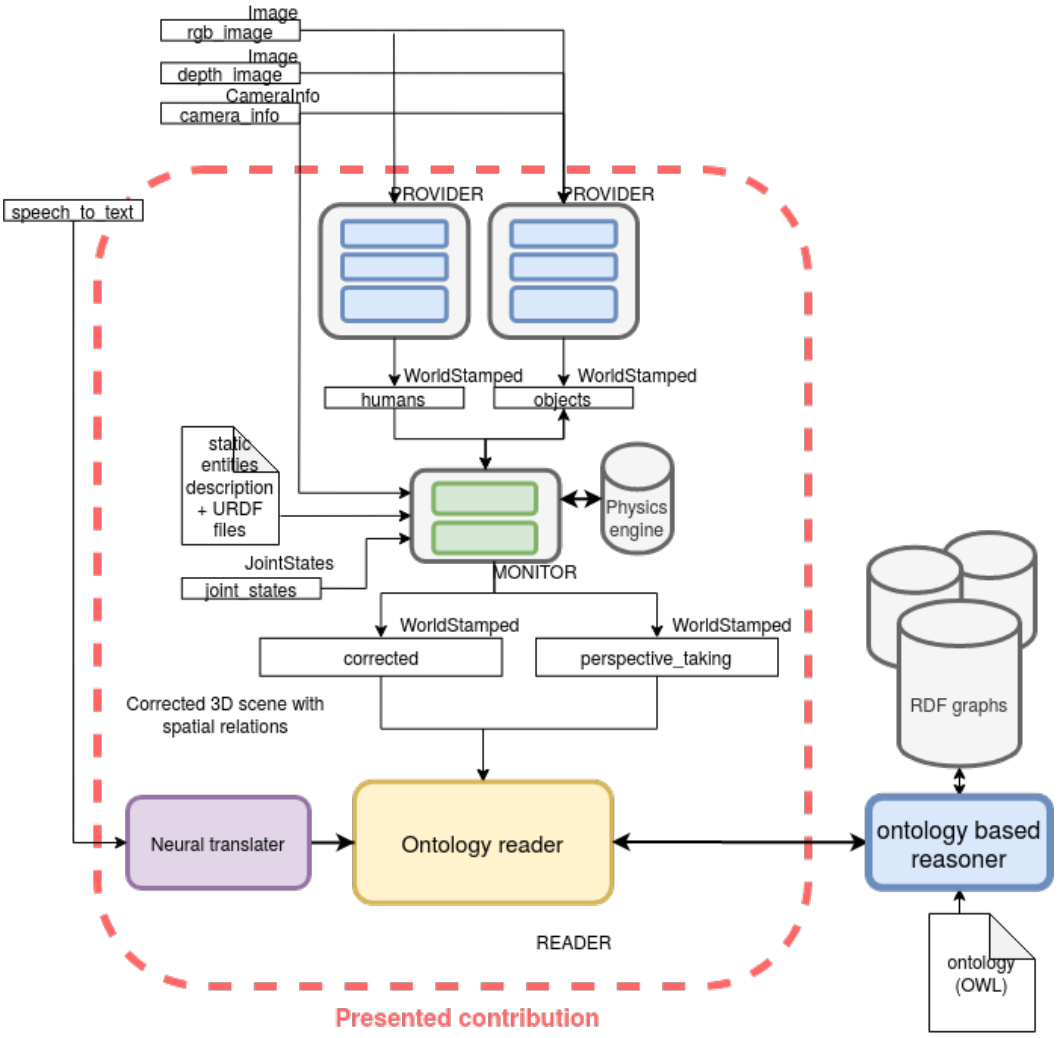


Figure 7.4: The architecture overview.

7.4 Ongoing work and conclusions

In this chapter, we presented the integration of our work on physical, geometric, and visual reasoning with ontology-based reasoners, which allows us to query UNDERWORLDS with SPARQL and reasoning on semantic correctness of the world.

Along with this integration, we provide a preliminary work that uses deep learning to map natural language into SPARQL queries to show the pertinence of using ontology-based reasoners to answer questions.

Currently, we are migrating from openNMT to Fairseq in order to verify the surprising results of [Yin 2019]. We are also adding more templates to generate more data. We do not doubt its interest in the future because our approach's extensibility makes it easy to add new templates to the system. Also, as the knowledge domain of the ontology will grow, the queries possible to implement will be more meaningful,

and at some point, we hope to create a robust way to capture NL semantic with a deep language model.

7.4.1 Future work

The dataset currently limits our neural translation. In the following future work, we present here solutions to enhance the data's quality and quantity.

Use more elaborate SPARQL queries In this work, we limited ourselves to simple SELECT/WHERE queries, one direction for the future could be to use the counting and ordering capabilities of SPARQL language to create more diversity in the dataset and ask, "how many apples are in the bag ?".

Extending the domain and dataset In this preliminary work, we trained a seq2seq neural network that captures semantic information of natural language in order to be able to query the RDF graphs (that represent the robot and others knowledge) while being robust to noise in the sentences (as it is usually the case when using speech to text technology).

We hypothesize that the overfitting observed during inference is due to the lack of diversity in the dataset. By extending the ontology domain and the dataset for neural translation, we hope to tackle this problem. One exciting direction would be to integrate abstract mental states such as shared plans like in [Devin 2016]. For the robot to be able to answer about the shared plan execution, for example: "What is the next thing to do?" or "Where do I have to put that?".

Integrating the timeline temporal and spatial information The situation-assessment situations are currently integrated into the ontology using RDF object properties, limiting us in reasoning. We plan to integrate in future work the situations as an instanced individual (in grey in the Fig.7.3). By doing so, we could also link the actions with the place it appends.

Chapter 8

Unexpected Daily Situation Dataset

Contents

8.1 Introduction

This chapter presents the progress recently published ([Sallami 2020]) in building a new dataset of unexpected daily situations (like someone tripping on a box while carrying a tray to the kitchen or someone burning him/herself with hot water and dropping a mug). Each situation involves one or two humans in a familiar, structured environment (e.g. a kitchen, a living room) with rich semantics. Correctly interpreting the situation (including recognising an error, undesired effect or incongruity when it occurs, as well as selecting the best repair action) requires beyond-state-of-art Spatio-temporal, semantic and socio-cognitive modelling. As such, the dataset aims to offer (i) a natural source of data to train and test such novel algorithms and (ii) provide a new benchmark against which algorithms can be demonstrated.

8.1.1 Motivation

In a landmark study where infants were shown to help an adult who proactively looked like he was struggling to put books on a shelf, Warneken and

Tomasello ([Warneken 2006] [Warneken 2007]) demonstrated that 2-year old infant can readily interpret intentions, recognise and predict error situations (in the case of Warneken's study, the door to the bookshelf being closed), and come up with effective altruistic helping behaviours.

In order to exhibit the same level of reasoning capabilities, a robot would likewise need to infer the intention of the observed human, detect unexpected or error situations, and decide what action to perform next to be a good, altruistic collaborator. To help to address this challenging problem, we provide a dataset – which doubles as a benchmark – and a methodology to record the data with a robot in incongruous yet daily life activities. This benchmark aims to help researchers to investigate the link between intention recognition, error detection, contextual understanding and mechanisms that are related to joint action by providing recordings of situations that *lead to a natural collaboration* (e.g. do not require explicit communication for the cooperation/collaboration to emerge). Examples include lifting a stack of cardboard, picking up a heavy table or trying to open a door while having both hands already busy.

8.1.2 Inspiration

Data-driven approaches to subsets of these problems have already led to promising results in action/intention recognition ([Simonyan 2014, Sigurdsson 2017, Zhao 2019, Xie 2018]) in real-world settings. We hope to do the same in the context of studying helping behaviour for robots. However, the need to have a robot record the data, combined with the fact that people do not typically have RGB-D cameras, we cannot use [Sigurdsson 2016] approach to creating a large-scale dataset.

Our benchmark is not intended for use with massive data-consuming algorithms, but more by applying representation learned on other datasets to enhance it with spatial/geometric reasoning (provided by the depth sensor/robot localisation) and lessons learned from developmental psychology.

We hope that this benchmark will help the HRI community publish more replicable results to build valuable knowledge and algorithms.

8.1.3 Related work

Charade ([Sigurdsson 2016]) is a complete dataset for action recognition and intention as it features sequential tasks in a real-world environment. Another dataset that is is often used is named Kinetics ([Carreira 2018], [Carreira 2019], [Smaira 2020]). This dataset is usually used for transfer learning like ImageNet, as it features video sequences of individual actions.

The problem that we encounter with these datasets in a robotic context is that many of them are based on Youtube videos that was made to be understandable—making them not generalisable in a context where people may go completely out of the view of the camera while been tracked by the robot, or where objects of interest may be occluded or behind the camera. Here we want to benefit from the

multi-sensory aspects of the tasks. A robot can rely not only on the camera image but also on a laser scan, for example.

Also, the action labels do not fit our purpose. In the context of a joint task, we do not need to detect if someone is "eating nachos" (example of action label from Kinectic 700). However, they are useful to be used in transfer-learning approaches.

In this work, more than action/intention recognition, we want to work on the underlying mechanism of joint action by studying actions that naturally lead to collaboration. To our knowledge, no previous work has been done in this direction, neither in AI or robotic context.

8.2 Methodology

Figure 8.1: Example snapshots from the dataset

8.2.1 Task selection

To study the joint action mechanism, we voluntarily focus on tasks where the collaboration naturally emerges in an altruistic way. In Warneken studies, because the actor was performing the task for young infants and apes, he had to exaggerate the social and emotive signals by making noises or pretending to be sad (because he failed the task). In our case, we want to study joint action between robots and human coworkers, and we cannot exaggerate social signals.

Also, recording videos in the presence of robots can be challenging with naive people, and they may change their behaviour unintentionally just because a robot is

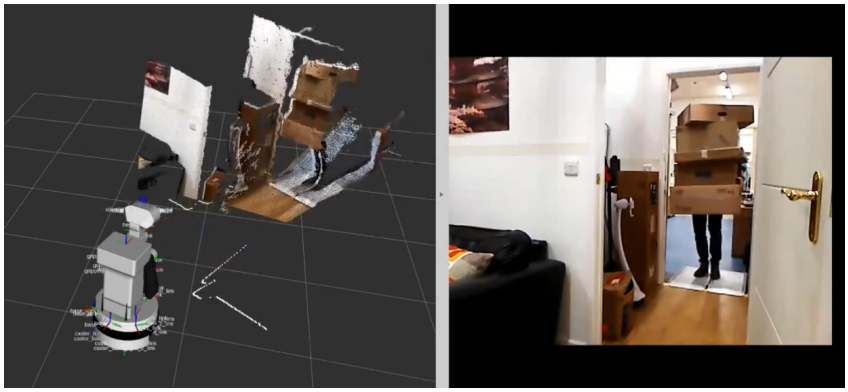


Figure 8.2: The session been replayed which contains RGBD data along with the laser scan and microphones.

looking at them. For that concern, we chose to records the videos with an actor that could eventually perform social signals to the robot (looking at him, for example).

We empirically spent much time testing the tasks that we chose (See Table. 8.1) by performing it in the laboratory without telling our coworkers that we were working on them. We noticed that naturally, people came to help us without having been told about the tests. Of course, this needs further experiments. A human-human study could be conducted to verify the pertinence of the tasks that we selected. For example, a participant could be placed in a waiting room without knowing that there was already in the experiment. An actor could then came to perform the task assigned, and we could analyse the pertinence of the tasks by analysing if people help naturally or not the actor without any instructions, like in Warneken experiment.

8.2.2 Recording methodology

8.2.2.1 Materials

The scenario involves a robot running ROS, which evolves in a mock-up or real studio. For this dataset, we used the Tiago robot. It features an RGB-D sensor mounted on a pan/tilt head, used to track the humans in the scene, and a 2D LIDAR for localisation. Audio from the robot's stereo microphone, as well as localisation information, are recorded as well. The calibrated sensor streams are synchronised and recorded using `rosbag`.

8.2.2.2 Scenario

The scenarios are grouped into two categories: (I) situations in which the human actor cannot complete the desired goal without assistance, and (II) situations in which assistance is not required but may have a positive impact on the scenario. Type (I) situations are further subdivided into (I-a) those which are impossible/will fail because of something that cannot be seen by the human actor and (I-b) the

Table 8.1: The four types of scenarios, with examples of tasks present in the dataset.

Scenario sub-group	Example
(I-a) Perspective	Identify an obstacle, not visible to the human, that prevents her/him to reach the desired location
(I-b) Physical Capability	Identify the human lifting a heavy object that he wants to move but cannot lift alone
(II-a) Task Related	Detect someone burning him/herself and dropping the mug while making a coffee
(II-b) Not task-related	Detect that an important item (e.g. phone) has been inadvertently dropped while performing another task

human actor's physical capability. Type (II) situations are subdivided into (II-a) those in which actions would directly contribute to the human actor's goal, and (II-b) improve the situation in a way that is not directly task/goal-related. Table 8.1 gives examples of each of these situations.

The recordings are 1-3 min long. They involve up to two humans in the same scene and as many objects as needed to perform the task, plus furniture/objects that are not being used during the task. The robot starts at an initial position and heads to the recording position controlled by a joystick while mapping the environment. The human actor enters the scene and performs the action assigned. During the whole scene, the robot tracks the human and what happens in the environment. The tracking is done by detecting the humans in the video and following them with the robot's pan/tilt head. In case of a salient event (like an object being dropped), the robot head is manually oriented towards the event.

Salient events are besides annotated in the recordings and used as checkpoints for the crowd-sourcing of the baseline of assistive behaviours (see below).

8.2.2.3 Variables

In order to add variety to the benchmark, we identified several situational/environmental variables than can be adjusted to create multiple variations of the same base scenario. These can be categorised as actor-related:

- Whether the face can be seen during the task or not
- The use of social signals (e.g. looking at the camera for assistance) or not
- The appearance and gender

or as environment-related:

- Location & related semantics (e.g. kitchen versus living room)
- Presence of a second human (and whether they are attentive) or not
- Position of robot within the room
- Presence of obstacles and additional furniture

8.2.3 Crowd-sourced annotations for possible assistive behaviours

To establish a human baseline of what assistive behaviour would be appropriate in a specific situation, we will run an online study on the Amazon Mechanical Turk (AMT) platform.

Before implementing such reasoning in a real robot, we need to know if we can generalise human-level annotations for robots in this context, which lead to two conditions, for the robot (Condition A) and the human (Condition B).

A set of randomly picked videos will be presented to each participant in the study.

When replaying the recordings, the video stop when a time the stamp is reached, and the questionnaire is prompted (the last frame of the video will still be displayed).

The following questions will be asked:

1. What was the goal of the person in the video?
2. (A: Should the robot / B: Would a person)[1] take any action at this point? If so, what?
3. Can you explain what made you suggest this action (or lack of action)?

The answers of (1) will be used as ground truth for intention recognition, (2) will be used to know if we can generalise human-level annotation, as well as serving as repair action ground truth and (3) will be used to explore contextual understanding.

The user study's annotations will be made available alongside the main `bag` files of the dataset.

8.3 Conclusions

To build the user study, we made a website using *jspsycho* a framework for psychologists. This framework allows us to generate an ecologic prompt with data associated and a text field for the answers; it records user-id, time to respond, and the study results into a PHP file, making it easy to use the Mechanical Turk.

8.3.1 Pre-analysis and annotation

We recorded in a total of 32 videos. Few shots were unusable because of several bugs (low fps from the camera during the recording) or because the robot was not tracking the correct person for various reasons (it was programmed to track the closest person). After analysing each video's content, we kept 24 videos that represent 41 minutes of recording for a total of 58907 frames. For each video, we chose an average of 18 frames of interest where a decision could occur. To be consistent in the way we selected them, we choose to select them when entering the room, leaving the room, going to act or acting. In total, we collected 434 frames of interest that will be used to prompt the questionnaire. The table 8.2 resume the total frames of interest in the dataset. Some recordings have a low frame per

[1] depending on the condition randomly picked for each participant

In this experiment, videos **recorded with this robot** will be played.

Press any key to continue.

(a) Example of instruction from the robot condition. The robot is displayed in order to help the subject to infer the robot capabilities

What is the goal of the person in the video?

Should the robot take any action at this point? If so, what?

Can you explain what made you suggest this action (or lack of action)?

Continue

(b) Exemple of question prompt with the robot condition. The last image of the video is keeped in order to avoid memorization issues for the participant.

Figure 8.3: Examples from the website developed for the Mechanical Turc.

second due to ROS topics' asynchronous nature (the data is recorded in a bag file). When processes happen in parallel, the camera bandwidth can be reduced (which is why that data is stamped in ROS).

8.3.2 Conclusion and Future work

This chapter presents an ongoing work on building a novel dataset of semantically and socially rich unexpected situations. With this dataset, we hope to supports the development of socially aware assistive robots by pushing state of art in Spatio-temporal, semantic, and social modelling. The scenarios were selected to exhibit situations where the robot needs socio-cognitive abilities to detect problematic domestic human activity situations and select the appropriate repair action.

This work is the very first step towards altruistic robots, and many works need to be done. For this work, we want to take inspiration from the methodology used in computer vision communities that lead to impressive results and spend time formalising the problem well. Here we discuss different aspects that we need to emphasise in order to continue this work.

Collecting more data

We were planning to record additional data at LAAS, but it has been delayed. Because more data is always useful for supervised learning, we need to augment this dataset constantly. We are currently working on analysing the already recorded data and enhancing the protocol to make it easier to reproduce. For example, we noticed that some records are unusable due to the high speed of the robot's head, causing frames drops or very low frames per second due to the recording with ROS. We are working to fully teleoperate the robot for those issues and not use person trackers to avoid frame drops during recording. Also, limiting the head velocity could help to have better data.

Annotation

Before annotating the data, we have to be sure that everything is exploitable and run a pilot experiment to see if naive users understand well the instructions. The data will be made public as soon as the user-study/annotations are completed. We are also thinking about having fixed-sized video clips (a few seconds) for each frame of interest, making it easier to handle the data with machine learning frameworks.

Context modeling

In this dataset, where illumination in the scene is not always right, object detection could have some issues. To prevent that, it is better to only detect the persons in the video and use their local context to reason about their objects. Also, we think that the RGB frames associated with the dense optical flow (used to model the surrounding motion) associated with CNN architectures could be the way to

go because high motion (the person reflexes, or an object falling) often induce something wrong. Also, adding engineered features (how many humans are in the scene) could help the classification problem. Attention-based approaches that are nowadays very mature could help understand what the robot learns from the data to decide.

When before What

A reasonable approach to solve this problem could be first to use the dataset to learn when the robotic agent needs to act and after trying to learn what to do (our dataset is not intended to solve the how). The first problem is a simple classification problem when the last one could be a captioning problem (because our annotation will be open questions). The data will certainly not be sufficient for the last problem, and transfer learning approaches will be mandatory.

Table 8.2: The table that describe the data recorded for the dataset.

ID clip	Duration (s)	frames count	frames per seconds	FOI count
1	40,99	1171	28,5	17
2	73,15	1147	15,6	28
3	244,02	7220	29,5	25
4	86,16	1434	16,6	8
5	196,33	2556	13,0	13
6	136,23	2431	17,8	20
7	83,66	1197	14,3	18
8	120,86	2463	20,3	20
9	122,91	2567	20,8	22
10	75,82	1712	22,5	14
11	170,55	3617	21,2	23
12	83,07	1746	21,0	17
13	126,39	2300	18,1	15
14	75,11	2034	27,0	14
15	107,59	3164	29,4	32
16	109,99	3158	28,7	29
17	79,45	2326	29,2	14
18	40,43	1164	28,7	9
19	53,80	1587	29,4	18
20	47,64	1406	29,5	9
21	117,34	3379	28,7	27
22	139,51	4039	28,9	14
23	85,94	2538	29,5	16
24	87,67	2551	29,0	12
Total	2504,61	58907	578,5	434
Average	104,35	2454,4	24,1	18

Conclusion

This thesis presents the work that has been done on physical and social reasoning for human-robot joint action during the last four years.

At the same time, we started a long-term international collaboration to work on embodied cognition with the Bristol Robotics Laboratory. The first work from this collaboration developed a framework for cascading situation-assessment components that we validated on a real-world scenario during the MuMMER project. This thesis technical contribution was supported with more theoretically interesting work: The first integration of simulation-based reasoning in a cognitive and interactive architecture for human-robot interaction to reason about out-of-view objects.

The integration of simulation-based reasoning for human-robot collaboration was inspired by the early physical cognition of infants. Thanks to the recent progress on real-time physics engines, we implement a simple yet efficient monitoring algorithm that analyses the physical plausibility of objects and infers actions *at runtime*.

Also, we emphasize perspective-taking capabilities that directly benefit from the corrected world state. We also show the first implementation of perspective-taking that extracts object detections with the same data structure as traditional perception pipelines, allowing us to use it as input to the same reasoning process as the robot perceived them.

We upgraded the framework to store visual features in the data structure and prepare future work on the interface between perceptual and higher cognitive reasoning to benefit from deep learning progress.

To show the pertinence of the physical reasoning and the choices made with the Underworlds framework, we integrate our work with ontology-based reasoners and explore the integration of deep language models.

To conclude this work, we recently started an effort to build a dataset that emphasizes the problematics that we face in human-robot interactions by taking inspiration from an influential work on infants' natural tendency to be altruistic. This last effort has been made to develop benchmarks to help HRI researchers build more replicable research to face tomorrow's challenges.

Conclusions with regards to research questions

During this thesis, we tried to answer several research questions summarized here:

- What are the core capabilities needed to endow a robot to perform situation assessment in a human-robot joint action context?
- What are the mechanisms and models needed to perceive the environment and reason on what is not visible?

- Is it possible to use simulation-based physical reasoning at runtime to reason finely about object persistence and physical inconsistencies?
- How can these mechanisms be integrated with a deep learning approach for intuitive physics?
- How to bind natural language models and knowledge-based systems for questions answering in the context of an embodied agent?
- How to go further in studying human-robot collaboration, and what are the next steps?

In this thesis, we chose to look at the literature in developmental psychology because their systemic approach, combined with recent findings in this field, allows roboticists to focus on early human cognitive abilities and understand how these capabilities emerge.

This approach allows us to build up architectures that possess modules inspired by human reasoning while implementing the modules with a roboticist approach.

One fundamental aspect of human reasoning about objects concerns physics and geometry. While several concepts like continuity are already used in every tracking system, solidity or gravity is absent. With our work, we hope that the use of physics simulation for the tracking system in the context of tabletop manipulation will help to build more robust interactions that integrate gravity and solidity principles. From our perspective, we need object-level representation, a classic perception system coupled with a graph representation for relational data, to perform situation assessment in a human-robot context. In addition to a physics engine that can help build up consistent models of the worlds to compute spatial relations in tabletop manipulation scenarios.

We think that combining language models with ontologies like it is already done today in the web-semantic can be interesting for robots that interact with humans. For sure, different aspects needs to be emphasized as the robot is an embedded agent. We hope that our work will help researchers to go further in this direction.

Situation assessment is a complete problem where multidisciplinarity is essential. This thesis combines computer vision, 3D graphics, physics engines, machine learning, developmental psychology, social signal processing, and natural language processing. In our view, future research in HRI will need international collaboration to create meaningful challenges, datasets and benchmarks where replicability is possible.

Future directions

This section emphasizes directions that could be interesting to continue to work on apart from staying up-to-date with perceptual systems that evolve very quickly.

Voxel based representation was part of the original design but has never been explored. In the future, this reasoning could be used to complement the mesh-based model and detect possible inconsistencies in the object-level representation. Also, voxel-based representation can model the areas never perceived and make it a promising candidate for active perception systems.

Link between perceptual and higher cognitive process Many works have been already done in this direction but needs further dedication in order to close the loop of the entire system and, in particular, with supervision and planning. The SPARQL query language and the ability to verify symbolic consistency make ontologies useful for robot agents. In the future, working more on the synergies between planning, symbolic reasoning, and supervision system with the help of a situation-assessment component that distributes the models needed could lead to exciting research in our point of view.

Extending ontology and neural language models This aspect is very promising as it allows the robot to interact with people naturally. However, this work is still in its first stages, and it needs to be completed by extending the ontology and dataset to combine deep language models with knowledge-based systems for human-robot collaboration.

Graph Neural Networks One of the steps of this work is to use the graph generated by Underworlds and encode it into fixed-size features with the help of graph neural networks. This will allow other reasoning, particularly context-based ones, to benefit from the graph's vector representation. One concrete example is using the graph features as an additional input to the language model to integrate contextual information into the SPARQL translation.

Bibliography

[Agrawal 2016] Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik and Sergey Levine. *Learning to Poke by Poking: Experiential Learning of Intuitive Physics.* In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 5074–5082. Curran Associates, Inc., 2016. (Cited in page 17.)

[Ajay 2019] Anurag Ajay, Maria Bauza, Jiajun Wu, Nima Fazeli, Joshua B Tenenbaum, Alberto Rodriguez and Leslie P Kaelbling. *Combining physical simulators and object-based networks for control.* In 2019 International Conference on Robotics and Automation (ICRA), pages 3217–3223. IEEE, 2019. (Cited in page 16.)

[Amos 2016] Brandon Amos, Bartosz Ludwiczuk and Mahadev Satyanarayanan. *OpenFace: A general-purpose face recognition library with mobile applications.* Technical Report, CMU-CS-16-118, CMU School of Computer Science, 2016. (Cited in page 74.)

[Baillargeon 1987] Renee Baillargeon. *Object permanence in 3 1/2-and 4 1/2-month-old infants.* Developmental psychology, vol. 23, no. 5, page 655, 1987. (Cited in page 8.)

[Baillargeon 1994] Renée Baillargeon. *Physical reasoning in young infants: Seeking explanations for impossible events.* British Journal of Developmental Psychology, vol. 12, no. 1, pages 9–33, 1994. (Cited in pages 8 and 9.)

[Baillargeon 2002] Renée Baillargeon. *The acquisition of physical knowledge in infancy: A summary in eight lessons.* Blackwell handbook of childhood cognitive development, vol. 1, no. 46-83, page 1, 2002. (Cited in page 11.)

[Baillargeon 2009] Renée Baillargeon, Di Wu, Sylvia Yuan, Jie Li and Yuyan Luo. Young infants' expectations about self-propelled objects, pages 285–352. Oxford University Press, 03 2009. (Cited in page 11.)

[Baillargeon 2010] Renée Baillargeon, Jie Li, Yael Gertner and Di Wu. *How Do Infants Reason about Physical Events?* The Wiley-Blackwell Handbook of Childhood Cognitive Development, pages 11–48, 2010. (Cited in page 11.)

[Baillargeon 2012] Renée Baillargeon and Susan Carey. *Core cognition and beyond: The acquisition of physical and numerical knowledge.* In In S. Pauen (Ed.), Early Childhood Development and Later Outcome. Citeseer, 2012. (Cited in page 10.)

[Baillargeon 2017] Renée Baillargeon and Gerald F DeJong. *Explanation-based learning in infancy.* Psychonomic bulletin & review, vol. 24, no. 5, pages 1511–1526, 2017. (Cited in page 11.)

[Bansal 2018] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa and Ajay Divakaran. *Zero-shot object detection.* In Proceedings of the European Conference on Computer Vision (ECCV), pages 384–400, 2018. (Cited in page 71.)

[Baron-Cohen 1985] Simon Baron-Cohen, Alan M Leslie, Uta Frith*et al. Does the autistic child have a "theory of mind".* Cognition, vol. 21, no. 1, pages 37–46, 1985. (Cited in pages 21 and 23.)

[Bates 2015] Christopher Bates, Peter W Battaglia, Ilker Yildirim and Joshua B Tenenbaum. *Humans predict liquid dynamics using probabilistic simulation.* In CogSci, 2015. (Cited in page 16.)

[Battaglia 2013] Peter W Battaglia, Jessica B Hamrick and Joshua B Tenenbaum. *Simulation as an engine of physical scene understanding.* Proceedings of the National Academy of Sciences, vol. 110, no. 45, pages 18327–18332, 2013. (Cited in pages 16, 18, 56, and 57.)

[Battaglia 2016] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende*et al. Interaction networks for learning about objects, relations and physics.* In Advances in neural information processing systems, pages 4502–4510, 2016. (Cited in pages 16 and 66.)

[Belhassein 2017] Kathleen Belhassein, Aurélie Clodic, Hélene Cochet, Marketta Niemelä, Päivi Heikkilä, Hanna Lammi and Antti Tammela. *Human-human guidance study.* 2017. (Cited in pages 45 and 46.)

[Bewley 2016] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos and Ben Upcroft. *Simple online and realtime tracking.* In 2016 IEEE International Conference on Image Processing (ICIP), pages 3464–3468. IEEE, 2016. (Cited in page 15.)

[Bogartz 1997] Richard S Bogartz, Jeanne L Shinskey and Cindy J Speaker. *Interpreting infant looking: The event set× event set design.* Developmental psychology, vol. 33, no. 3, page 408, 1997. (Cited in page 10.)

[Bolya 2019] Daniel Bolya, Chong Zhou, Fanyi Xiao and Yong Jae Lee. *Yolact: Real-time instance segmentation.* In Proceedings of the IEEE international conference on computer vision, pages 9157–9166, 2019. (Cited in page 14.)

[Breazeal 2009] Cynthia Breazeal, Jesse Gray and Matt Berlin. *An embodied cognition approach to mindreading skills for socially intelligent robots.* The International Journal of Robotics Research, vol. 28, no. 5, pages 656–680, 2009. (Cited in pages 25 and 26.)

[Carey 2000] Susan Carey. *The origin of concepts.* Journal of Cognition and Development, vol. 1, no. 1, pages 37–41, 2000. (Cited in page 10.)

[Carreira 2018] João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier and Andrew Zisserman. *A Short Note about Kinetics-600.* CoRR, vol. abs/1808.01340, 2018. (Cited in page 92.)

[Carreira 2019] João Carreira, Eric Noland, Chloe Hillier and Andrew Zisserman. *A Short Note on the Kinetics-700 Human Action Dataset.* CoRR, vol. abs/1907.06987, 2019. (Cited in page 92.)

[Cashon 2000] Cara H Cashon and Leslie B Cohen. *Eight-month-old infants' perception of possible and impossible events.* Infancy, vol. 1, no. 4, pages 429–446, 2000. (Cited in page 10.)

[Cho 2014] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau and Yoshua Bengio. *On the properties of neural machine translation: Encoder-decoder approaches.* arXiv preprint arXiv:1409.1259, 2014. (Cited in page 12.)

[Coumans 2020] Erwin Coumans and Yunfei Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning.* `http://pybullet.org`, 2016–2020. (Cited in page 40.)

[Dehaene 2006] Stanislas Dehaene, Véronique Izard, Pierre Pica and Elizabeth Spelke. *Core knowledge of geometry in an Amazonian indigene group.* Science, vol. 311, no. 5759, pages 381–384, 2006. (Cited in page 8.)

[Deng 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. *Imagenet: A large-scale hierarchical image database.* In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009. (Cited in page 14.)

[Devin 2016] Sandra Devin and Rachid Alami. *An implemented theory of mind to improve human-robot shared plans execution.* In 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 319–326. IEEE, 2016. (Cited in page 89.)

[Devlin 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805, 2018. (Cited in page 13.)

[Erhan 2014] Dumitru Erhan, Christian Szegedy, Alexander Toshev and Dragomir Anguelov. *Scalable object detection using deep neural networks.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2147–2154, 2014. (Cited in page 14.)

[Finn 2016] Chelsea Finn, Ian Goodfellow and Sergey Levine. *Unsupervised learning for physical interaction through video prediction.* In Advances in neural information processing systems, pages 64–72, 2016. (Cited in page 17.)

[Fischer 2016] Tobias Fischer and Yiannis Demiris. *Markerless perspective taking for humanoid robots in unconstrained environments.* In 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 3309–3316. IEEE, 2016. (Cited in page 24.)

[Fischer 2018] Tobias Fischer, Hyung Jin Chang and Yiannis Demiris. *RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments.* In European Conference on Computer Vision, pages 339–357, September 2018. (Cited in page 72.)

[Flavell 1977] J. Flavell. *The development of knowledge about visual perception.* Nebraska Symposium on Motivation. Nebraska Symposium on Motivation, vol. 25, pages 43–76, 1977. (Cited in page 20.)

[Flavell 1981] John H Flavell, Barbara A Everett, Karen Croft and Eleanor R Flavell. *Young children's knowledge about visual perception: Further evidence for the Level 1–Level 2 distinction.* Developmental psychology, vol. 17, no. 1, page 99, 1981. (Cited in page 20.)

[Foster 2016] Mary Ellen Foster, Rachid Alami, Olli Gestranius, Oliver Lemon, Marketta Niemelä, Jean-Marc Odobez and Amit Kumar Pandey. *The MuMMER project: Engaging human-robot interaction in real-world public spaces.* In International Conference on Social Robotics, pages 753–763. Springer, 2016. (Cited in page 43.)

[Foster 2019] Mary Ellen Foster, Bart Craenen, Amol Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, Ioannis Papaioannou, Andrea Vanzo, Jean-Marc Odobez, Olivier Canévet*et al.* *MuMMER: Socially Intelligent Human-Robot Interaction in Public Spaces.* arXiv preprint arXiv:1909.06749, 2019. (Cited in page 43.)

[Gehring 2016] Jonas Gehring, Michael Auli, David Grangier and Yann N Dauphin. *A convolutional encoder model for neural machine translation.* arXiv preprint arXiv:1611.02344, 2016. (Cited in pages 12 and 87.)

[Gers 1999] Felix A Gers, Jürgen Schmidhuber and Fred Cummins. *Learning to forget: Continual prediction with LSTM.* 1999. (Cited in page 12.)

[Girshick 2015] Ross Girshick. *Fast r-cnn.* In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015. (Cited in page 14.)

[Hadsell 2006] Raia Hadsell, Sumit Chopra and Yann LeCun. *Dimensionality reduction by learning an invariant mapping.* In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 1735–1742. IEEE, 2006. (Cited in page 13.)

[Hamrick 2011] Jessica Hamrick, Peter Battaglia and Joshua B Tenenbaum. *Internal physics models guide probabilistic judgments about object dynamics.* In

Proceedings of the 33rd annual conference of the cognitive science society, volume 2. Cognitive Science Society Austin, TX, 2011. (Cited in page 16.)

[He 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár and Ross B. Girshick. *Mask R-CNN*. CoRR, vol. abs/1703.06870, 2017. (Cited in page 14.)

[Hespos 2001] Susan J Hespos and Renée Baillargeon. *Reasoning about containment events in very young infants*. Cognition, vol. 78, no. 3, pages 207–245, 2001. (Cited in page 8.)

[Hiatt 2011] Laura M Hiatt, Anthony M Harrison and J Gregory Trafton. *Accommodating human variability in human-robot teams through theory of mind*. In Twenty-Second International Joint Conference on Artificial Intelligence, 2011. (Cited in page 25.)

[Johnson 2005] Matthew Johnson and Yiannis Demiris. *Perceptual perspective taking and action recognition*. International Journal of Advanced Robotic Systems, vol. 2, no. 4, page 32, 2005. (Cited in page 24.)

[Kahneman 1992] Daniel Kahneman, Anne Treisman and Brian J Gibbs. *The reviewing of object files: Object-specific integration of information*. Cognitive psychology, vol. 24, no. 2, pages 175–219, 1992. (Cited in page 11.)

[Karthik 2020] Shyamgopal Karthik, Ameya Prabhu and Vineet Gandhi. *Simple Unsupervised Multi-Object Tracking*. arXiv preprint arXiv:2006.02609, 2020. (Cited in page 15.)

[Khambhaita 2020] Harmish Khambhaita and Rachid Alami. *Viewing robot navigation in human environment as a cooperative activity*. In Robotics Research, pages 285–300. Springer, 2020. (Cited in page 45.)

[Klein 2017] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart and Alexander M. Rush. *OpenNMT: Open-Source Toolkit for Neural Machine Translation*. In Proc. ACL, 2017. (Cited in page 87.)

[Koch 2015] Gregory Koch, Richard Zemel and Ruslan Salakhutdinov. *Siamese neural networks for one-shot image recognition*. In ICML deep learning workshop, volume 2. Lille, 2015. (Cited in page 13.)

[Kuhn 1955] Harold W Kuhn. *The Hungarian method for the assignment problem*. Naval research logistics quarterly, vol. 2, no. 1-2, pages 83–97, 1955. (Cited in pages 15 and 27.)

[Kunze 2017] Lars Kunze and Michael Beetz. *Envisioning the qualitative effects of robot manipulation actions using simulation-based projections*. Artificial Intelligence, vol. 247, pages 352–380, 2017. (Cited in page 17.)

[Lake 2017] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum and Samuel J Gershman. *Building machines that learn and think like people.* Behavioral and brain sciences, vol. 40, 2017. (Cited in page 2.)

[Lallement 2014] Raphaël Lallement, Lavindra De Silva and Rachid Alami. *HATP: An HTN Planner for Robotics.* In 2nd ICAPS Workshop on Planning and Robotics, Portsmouth, United States, June 2014. (Cited in page 25.)

[Lemaignan 2010] Séverin Lemaignan, Raquel Ros, Lorenz Mösenlechner, Rachid Alami and Michael Beetz. *ORO, a knowledge management platform for cognitive architectures in robotics.* In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3548–3553. IEEE, 2010. (Cited in pages 25 and 87.)

[Lemaignan 2012] Séverin Lemaignan, Raquel Ros, Emrah Akin Sisbot, Rachid Alami and Michael Beetz. *Grounding the Interaction: Anchoring Situated Discourse in Everyday Human-Robot Interaction.* 2012 International Journal of Social Robotics, vol. 4, no. 2, pages 181–199, April 2012. (Cited in pages 45, 80, 83, and 84.)

[Lemaignan 2017] Séverin Lemaignan, Mathieu Warnier, E. Akin Sisbot, Aurélie Clodic and Rachid Alami. *Artificial cognition for social human–robot interaction: An implementation.* Artificial Intelligence, vol. 247, pages 45–69, 2017. (Cited in pages 1, 25, 27, 29, and 30.)

[Lemaignan 2018] Séverin Lemaignan, Yoan Sallami, Christopher Wallbridge, Aurélie Clodic, Tony Belpaeme and Rachid Alami. *UNDERWORLDS: Cascading Situation Assessment for Robots.* In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, October 2018. (Cited in pages 27, 32, and 49.)

[Leslie 1993] Alan M Leslie. A theory of agency. Citeseer, 1993. (Cited in page 10.)

[Leslie 1994] Alan M Leslie. *ToMM, ToBy, and Agency: Core architecture and domain specificity.* Mapping the mind: Domain specificity in cognition and culture, vol. 29, pages 119–48, 1994. (Cited in pages 18 and 22.)

[Li 2016] Wenbin Li, Seyedmajid Azimi, Aleš Leonardis and Mario Fritz. *To Fall Or Not To Fall: A Visual Approach to Physical Stability Prediction.* arXiv preprint arXiv:1604.00066, 03 2016. (Cited in page 17.)

[Lin 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C Lawrence Zitnick. *Microsoft coco: Common objects in context.* In European conference on computer vision, pages 740–755. Springer, 2014. (Cited in page 71.)

[Lin 2020] Yi Lin, Maayan Stavans and Renée Baillargeon. Infants' physical reasoning and the cognitive architecture that supports it. Cambridge University Press, 2020. (Cited in page 10.)

[Liu 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu and Alexander C Berg. *Ssd: Single shot multibox detector.* In European conference on computer vision, pages 21–37. Springer, 2016. (Cited in page 14.)

[Luo 2005] Yuyan Luo and Renée Baillargeon. *When the ordinary seems unexpected: evidence for incremental physical knowledge in young infants.* Cognition, vol. 95, no. 3, pages 297–328, 2005. (Cited in page 11.)

[Mayima] Amandine Mayima, Aurélie Clodic and Rachid Alami. *Toward a Robot Computing an Online Estimation of the Quality of its Interaction with its Human Partner.* In 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pages 291–298. IEEE. (Cited in page 48.)

[Michelon 2006] Pascale Michelon and Jeffrey M Zacks. *Two kinds of visual perspective taking.* Perception & psychophysics, vol. 68, no. 2, pages 327–337, 2006. (Cited in page 20.)

[Mikolov 2013] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. *Efficient estimation of word representations in vector space.* arXiv preprint arXiv:1301.3781, 2013. (Cited in page 13.)

[Milliez 2014] Grégoire Milliez, Matthieu Warnier, Aurélie Clodic and Rachid Alami. *A framework for endowing an interactive robot with reasoning capabilities about perspective-taking and belief management.* In The 23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 1103–1109. IEEE, 2014. (Cited in pages 1, 25, and 30.)

[Moll 2006] Henrike Moll and Michael Tomasello. *Level 1 perspective-taking at 24 months of age.* British Journal of Developmental Psychology, vol. 24, no. 3, pages 603–613, 2006. (Cited in page 21.)

[Mösenlechner 2013] Lorenz Mösenlechner and Michael Beetz. *Fast temporal projection using accurate physics-based geometric reasoning.* In 2013 IEEE International Conference on Robotics and Automation (ICRA), pages 1821–1827. IEEE, 2013. (Cited in page 59.)

[Mottaghi 2016] Roozbeh Mottaghi, Hessam Bagherinezhad, Mohammad Rastegari and Ali Farhadi. *Newtonian scene understanding: Unfolding the dynamics of objects in static images.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3521–3529, 2016. (Cited in page 17.)

[Mösenlechner 2009] Lorenz Mösenlechner and Michael Beetz. *Using physics- and sensor-based simulation for high-fidelity temporal projection of realistic robot behavior*. In 19th International Conference on Automated Planning and Scheduling (ICAPS'09), 2009. (Cited in page 17.)

[Mösenlechner 2011] Lorenz Mösenlechner and Michael Beetz. *Parameterizing actions to have the appropriate effects*. In 2011 IEEE International Conference on Intelligent Robots and Systems (ICRA), pages 4141–4147, 2011. (Cited in page 17.)

[Needham 1993] Amy Needham and Renee Baillargeon. *Intuitions about support in 4.5-month-old infants*. Cognition, vol. 47, no. 2, pages 121–148, 1993. (Cited in pages 9 and 10.)

[Newcombe 1999] Nora Newcombe, Janellen Huttenlocher and Amy Learmonth. *Infants' coding of location in continuous space*. Infant Behavior and Development, vol. 22, no. 4, pages 483–510, 1999. (Cited in page 8.)

[Onishi 2005] Kristine H Onishi and Renée Baillargeon. *Do 15-month-old infants understand false beliefs?* science, vol. 308, no. 5719, pages 255–258, 2005. (Cited in pages 21 and 22.)

[Ott 2019] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier and Michael Auli. *fairseq: A Fast, Extensible Toolkit for Sequence Modeling*. In Proceedings of NAACL-HLT 2019: Demonstrations, 2019. (Cited in page 87.)

[Parkhi 2015] Omkar M Parkhi, Andrea Vedaldi and Andrew Zisserman. *Deep face recognition*. 2015. (Cited in page 14.)

[Pennington 2014] Jeffrey Pennington, Richard Socher and Christopher D Manning. *Glove: Global vectors for word representation*. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014. (Cited in page 13.)

[Peters 2018] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee and Luke Zettlemoyer. *Deep contextualized word representations*. arXiv preprint arXiv:1802.05365, 2018. (Cited in page 13.)

[Piaget 1952] Jean Piaget and Margaret Cook. The origins of intelligence in children, volume 8. International Universities Press New York, 1952. (Cited in page 8.)

[Piaget 1954] Jean Piaget and Margaret Trans Cook. The construction of reality in the child. Basic Books, 1954. (Cited in page 8.)

[Piaget 1956] Jean Piaget and Barbel Inhelder. *The child's conception of space.* FJ Langdon & JL Lunzer, trans.). London: Routledge & Kegan Paul, 1956. (Cited in pages 19 and 20.)

[Redmon 2016] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. (Cited in page 14.)

[Redmon 2018] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement.* CoRR, vol. abs/1804.02767, 2018. (Cited in page 14.)

[Reimers 2019] Nils Reimers and Iryna Gurevych. *Sentence-bert: Sentence embeddings using siamese bert-networks.* arXiv preprint arXiv:1908.10084, 2019. (Cited in page 14.)

[Ros 2010a] Raquel Ros, Séverin Lemaignan, E Akin Sisbot, Rachid Alami, Jasmin Steinwender, Katharina Hamann and Felix Warneken. *Which one? grounding the referent based on efficient human-robot interaction.* In 19th International Symposium in Robot and Human Interactive Communication, pages 570–575. IEEE, 2010. (Cited in pages 24 and 25.)

[Ros 2010b] Raquel Ros, E Akin Sisbot, Rachid Alami, Jasmin Steinwender, Katharina Hamann and Felix Warneken. *Solving ambiguities with perspective taking.* In 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 181–182. IEEE, 2010. (Cited in page 24.)

[Sallami 2019] Yoan Sallami, Séverin Lemaignan, Aurélie Clodic and Rachid Alami. *Simulation-based physics reasoning for consistent scene estimation in an HRI context.* In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 7834–7841. IEEE, 2019. (Cited in pages 27, 38, 40, 55, 62, 64, and 83.)

[Sallami 2020] Yoan Sallami, Katie Winkle, Nicola Webb, Severin Lemaignan and Rachid Alami. *The Unexpected Daily Situations (UDS) Dataset: A New Benchmark for Socially-Aware Assistive Robots.* In Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, HRI '20, page 427–429, New York, NY, USA, 2020. Association for Computing Machinery. (Cited in page 91.)

[Sanchez-Gonzalez 2020] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec and Peter W Battaglia. *Learning to simulate complex physics with graph networks.* arXiv preprint arXiv:2002.09405, 2020. (Cited in page 16.)

[Sarthou 2019] Guillaume Sarthou, Rachid Alami and Aurélie Clodic. *Semantic Spatial Representation: a unique representation of an environment based on an ontology for robotic applications.* 2019. (Cited in pages 48 and 87.)

[Scarselli 2008] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner and Gabriele Monfardini. *The graph neural network model.* IEEE Transactions on Neural Networks, vol. 20, no. 1, pages 61–80, 2008. (Cited in page 16.)

[Schober 1993] Michael F Schober. *Spatial perspective-taking in conversation.* Cognition, vol. 47, no. 1, pages 1–24, 1993. (Cited in page 20.)

[Schroff 2015] Florian Schroff, Dmitry Kalenichenko and James Philbin. *Facenet: A unified embedding for face recognition and clustering.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015. (Cited in pages 13 and 14.)

[Scott 2009] Rose M Scott and Renée Baillargeon. *Which penguin is this? Attributing false beliefs about object identity at 18 months.* Child development, vol. 80, no. 4, pages 1172–1196, 2009. (Cited in page 22.)

[Sigurdsson 2016] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev and Abhinav Gupta. *Hollywood in homes: Crowdsourcing data collection for activity understanding.* In European Conference on Computer Vision, pages 510–526. Springer, 2016. (Cited in page 92.)

[Sigurdsson 2017] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi and Abhinav Gupta. *Asynchronous temporal fields for action recognition.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 585–594, 2017. (Cited in page 92.)

[Simonyan 2014] Karen Simonyan and Andrew Zisserman. *Two-stream convolutional networks for action recognition in videos.* In Advances in neural information processing systems, pages 568–576, 2014. (Cited in page 92.)

[Sisbot 2007] Emrah Akin Sisbot, Luis F Marin and Rachid Alami. *Spatial reasoning for human robot interaction.* In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2281–2287. IEEE, 2007. (Cited in page 25.)

[Sisbot 2011a] E Akin Sisbot, Raquel Ros and Rachid Alami. *Situation assessment for human-robot interactive object manipulation.* In 2011 IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pages 15–20. IEEE, 2011. (Cited in pages 1 and 30.)

[Sisbot 2011b] E Akin Sisbot, Raquel Ros and Rachid Alami. *Situation assessment for human-robot interactive object manipulation.* In 2011 RO-MAN, pages 15–20. IEEE, 2011. (Cited in page 62.)

[Smaira 2020] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu and Andrew Zisserman. *A Short Note on the Kinetics-700-2020 Human Action Dataset.* arXiv preprint arXiv:2010.10864, 2020. (Cited in page 92.)

[Song 2008] Hyun-joo Song and Renée Baillargeon. *Infants' reasoning about others' false perceptions.* Developmental psychology, vol. 44, no. 6, page 1789, 2008. (Cited in page 22.)

[Song 2019] Lingxue Song, Dihong Gong, Zhifeng Li, Changsong Liu and Wei Liu. *Occlusion robust face recognition based on mask learning with pairwise differential siamese network.* In Proceedings of the IEEE International Conference on Computer Vision, pages 773–782, 2019. (Cited in page 14.)

[Soru 2017a] Tommaso Soru, Edgard Marx, Diego Moussallem, Gustavo Publio, André Valdestilhas, Diego Esteves and Ciro Baron Neto. *SPARQL as a Foreign Language.* arXiv preprint arXiv:1708.07624, 2017. (Cited in page 85.)

[Soru 2017b] Tommaso Soru, Edgard Marx, Diego Moussallem, Gustavo Publio, André Valdestilhas, Diego Esteves and Ciro Baron Neto. *SPARQL as a Foreign Language.* CoRR, vol. abs/1708.07624, 2017. (Cited in pages 85 and 87.)

[Soru 2018] Tommaso Soru, Edgard Marx, André Valdestilhas, Diego Esteves, Diego Moussallem and Gustavo Publio. *Neural machine translation for query construction and composition.* arXiv preprint arXiv:1806.10478, 2018. (Cited in page 85.)

[Spelke 1992] Elizabeth S Spelke, Karen Breinlinger, Janet Macomber and Kristen Jacobson. *Origins of knowledge.* Psychological review, vol. 99, no. 4, page 605, 1992. (Cited in pages 8 and 10.)

[Spelke 1995] Elizabeth S Spelke, Peter Vishton and Claes Von Hofsten. *Object perception, object-directed action, and physical knowledge in infancy.* The Cognitive Neurosciences, pages 165–179, 1995. (Cited in page 10.)

[Surian 2007] Luca Surian, Stefania Caldi and Dan Sperber. *Attribution of beliefs by 13-month-old infants.* Psychological science, vol. 18, no. 7, pages 580–586, 2007. (Cited in page 22.)

[Teja 2020] Phani Teja and Rachid Alami. *HATEB-2: Reactive Planning and Decision making in Human-Robot Co-navigation.* In International Conference on Robot & Human Interactive Communication, 2020, 2020. (Cited in page 47.)

[Trafton 2013] J Gregory Trafton, Laura M Hiatt, Anthony M Harrison, Franklin P Tamborello, Sangeet S Khemlani and Alan C Schultz. *Act-r/e: An embodied cognitive architecture for human-robot interaction.* Journal of Human-Robot Interaction, vol. 2, no. 1, pages 30–55, 2013. (Cited in pages 25 and 26.)

[Träuble 2010] Birgit Träuble, Vesna Marinović and Sabina Pauen. *Early theory of mind competencies: Do infants understand others' beliefs?* Infancy, vol. 15, no. 4, pages 434–444, 2010. (Cited in page 22.)

[Ullman 2017] Tomer D Ullman, Elizabeth Spelke, Peter Battaglia and Joshua B Tenenbaum. *Mind games: Game engines as an architecture for intuitive physics.* Trends in cognitive sciences, vol. 21, no. 9, pages 649–665, 2017. (Cited in page 16.)

[Vaswani 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser and Illia Polosukhin. *Attention is all you need.* In Advances in neural information processing systems, pages 5998–6008, 2017. (Cited in page 87.)

[Waldhart 2018] Jules Waldhart, Aurélie Clodic and Rachid Alami. *Planning human and robot placements for shared visual perspective.* 2018. (Cited in page 47.)

[Waldhart 2019] Jules Waldhart, Aurélie Clodic and Rachid Alami. *Reasoning on Shared Visual Perspective to Improve Route Directions.* In 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pages 1–8. IEEE, 2019. (Cited in page 47.)

[Wang 2004] Su-hua Wang, Renée Baillargeon and Laura Brueckner. *Young infants' reasoning about hidden objects: Evidence from violation-of-expectation tasks with test trials only.* Cognition, vol. 93, no. 3, pages 167–198, 2004. (Cited in page 10.)

[Warneken 2006] Felix Warneken and Michael Tomasello. *Altruistic helping in human infants and young chimpanzees.* science, vol. 311, no. 5765, pages 1301–1303, 2006. (Cited in pages 6 and 92.)

[Warneken 2007] Felix Warneken and Michael Tomasello. *Helping and cooperation at 14 months of age.* Infancy, vol. 11, no. 3, pages 271–294, 2007. (Cited in page 92.)

[Weitnauer 2010] Erik Weitnauer, Robert Haschke and Helge Ritter. *Evaluating a physics engine as an ingredient for physical reasoning.* In International Conference on Simulation, Modeling, and Programming for Autonomous Robots, pages 144–155. Springer, 2010. (Cited in pages 17 and 40.)

[Wellman 1988] HM Wellman and K Bartsch. *Young children's reasoning about beliefs.* Cognition, vol. 30, no. 3, pages 239–277, 1988. (Cited in page 21.)

[Wellman 1992] Henry M Wellman and Susan A Gelman. *Cognitive development: Foundational theories of core domains.* Annual review of psychology, vol. 43, no. 1, pages 337–375, 1992. (Cited in page 10.)

[Wilcox 1996] Teresa Wilcox, Lynn Nadel and Rosemary Rosser. *Location memory in healthy preterm and full-term infants.* Infant Behavior and Development, vol. 19, no. 3, pages 309–323, 1996. (Cited in page 8.)

[Wimmer 1983] Heinz Wimmer and Josef Perner. *Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception.* Cognition, vol. 13, no. 1, pages 103–128, 1983. (Cited in page 21.)

[Wojke 2017] Nicolai Wojke, Alex Bewley and Dietrich Paulus. *Simple online and realtime tracking with a deep association metric.* In 2017 IEEE international conference on image processing (ICIP), pages 3645–3649. IEEE, 2017. (Cited in page 15.)

[Wu 2015] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman and Josh Tenenbaum. *Galileo: Perceiving physical object properties by integrating a physics engine with deep learning.* In Advances in neural information processing systems, pages 127–135, 2015. (Cited in page 17.)

[Wu 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang and S Yu Philip. *A comprehensive survey on graph neural networks.* IEEE Transactions on Neural Networks and Learning Systems, 2020. (Cited in page 16.)

[Xie 2018] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu and Kevin Murphy. *Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification.* In The European Conference on Computer Vision (ECCV), September 2018. (Cited in page 92.)

[Xu 2016] Jiacheng Xu, Kan Chen, Xipeng Qiu and Xuanjing Huang. *Knowledge graph representation with jointly structural and textual encoding.* arXiv preprint arXiv:1611.08661, 2016. (Cited in page 16.)

[Yin 2019] Xiaoyu Yin, Dagmar Gromann and Sebastian Rudolph. *Neural Machine Translating from Natural Language to SPARQL.* arXiv preprint arXiv:1906.09302, 2019. (Cited in pages 87 and 88.)

[Yuan 2020] Tao Yuan, Hangxin Liu, Lifeng Fan, Zilong Zheng, Tao Gao, Yixin Zhu and Song-Chun Zhu. *Joint Inference of States, Robot Knowledge, and Human (False-)Beliefs.* In ICRA, 2020. (Cited in pages 27 and 28.)

[Zhang 2016] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li and Yu Qiao. *Joint face detection and alignment using multitask cascaded convolutional networks.* IEEE Signal Processing Letters, vol. 23, no. 10, pages 1499–1503, 2016. (Cited in page 72.)

[Zhao 2019] Hang Zhao, Zhicheng Yan, Lorenzo Torresani and Antonio Torralba. *HACS: Human Action Clips and Segments Dataset for Recognition and Temporal Localization.* arXiv preprint arXiv:1712.09374, 2019. (Cited in page 92.)

[Zhu 2016] Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos and Song-Chun Zhu. *Inferring forces and learning human utilities from videos.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3823–3833, 2016. (Cited in page 17.)

Abstract: In order to perform a collaborative task with a person, a robot needs to be able to reason about the objects and the people it interacts with.

Developmental psychology gives a good insight into how children develop models of the world, which can help to design new robotic architectures for efficient and robust human-robot interactions.

In the first place, we present an architecture based on a hybrid data structure that combines geometric and relational information and neural representations.

This architecture aims to benefit from recent progress in computer vision and natural language processing while enabling efficient 3D reasoning by building on top of that a consistent 3D model of the world, which allows image rendering from any point of view in the scene.

Then we explore two key reasoning modalities in the context of a human-robot joint action: physical reasoning and belief reasoning.

Physical reasoning allows the robot to use Newtonian physics to reason about objects that are not visible while monitoring what is physically plausible to infer actions.

In this thesis, we present a work inspired by developmental psychology in which we use a physics simulator to correct the position of perceived objects and infer the position of non-visible objects using Newtonian physics. The algorithm is also able to infer the human partner's actions by analyzing physical violations between the simulated world and the perceived one.

Beliefs reasoning is another key feature for robots that assist humans. At its core, this reasoning is based on visual perspective taking: the ability to reason from the point of view of another person.

In this thesis, we also show the modularity of the approach by binding ontology-based reasoners and the situation-assessment component developed that allows visual perspective-taking.

This interaction allows querying entities generated by the perceptual and physical system using SPARQL language. We show interest in this approach with preliminary work on using neural-based language models that benefit from the expressiveness of SPARQL queries.

We conclude with a discussion about the system's limitations and we open to future work that could lead to exciting research in this field.

Keywords:
Physical reasoning, Visual perspective taking, Human-Robot collaboration