MASTER THESIS

A Domestic Service Robotics study case: from Object Manipulation to Perception and Learning

Étude de cas d'une application de robotique domestique : Manipulation, Perception et Apprentissage

Maëlic NEAU

September 9, 2021

Master of Computer Science
International Section

Université de Bretagne Occidentale

Supervisor(s)	Prof. Cédric BUCHE
Teacher(s)	François MONIN Philippe LE PARC

Abstract

Domestic Service Robotics is one of the biggest growing domains in robotics. In this document I present my research with Domestic Service robots in two distinct work packages: Object Manipulation and Learning from perceptual data. The first work package was conducted in the context of an international competition in robotics: the RoboCup@Home. This year the competition consisted in two tasks of picking and placing objects in a simulated environment. My contribution to the competition has led the team to win the third place in the Social Standard Platform League. In this contribution up-to-date solutions have been used for Object Detection and Manipulation in a new optimized architecture. The second work package is a literature overview in the field of Machine Learning from multimodal perceptual data. This contribution is part of a new project proposal "Multimodal analysis and learning of human interactions by a companion robot: detecting and fulfilling user needs".

Acknowledgments

First of all, I want to thank my supervisor, Pr. Cédric Buche (CNRS / ENIB, CROSSING) who gave me free choice on my research focus and supported me in this project and for almost three year of work with the RoboBreizh team.

I also want to give a particular thanks to all the RoboBreizh team which helped me to carry out this project: Amélie Legeleux (PhD student, ENIB), Antoine Dizet (PhD student, ENIB), Natanael Wondimu (PhD Student, ENIB), Cédric LeBono (PhD student, IMT Atlantique), Maël Bouabdelli (Research Engineer, INSA Rouen), Sandratra Rasendrasoa (PhD Student, INSA Rouen) and Mihai Andries (Associate Professor, IMT Atlantique). Support also came from across the seas with Paulo Santos, Professor at Flinders University, who gave me useful insights in some robotics paradigms.

Finally I want to thank the RAMBO team from Lab-STICC for supporting the RoboBreizh project and the 2021 RoboCup@Home organizing committee for making the competition possible in the pandemic situation.

Contents

Introduction

Nowadays, service robotics has been identified as having the largest potential market for years to come according to the international statistics about robotics [1]. Service robots are made to assist human in dirty, dull, distant, dangerous or repetitive tasks in daily-life environments. There is a lot of different platforms in service robotics (e.g. vacuum cleaning robots, elderly care robots, industrial robots). In this domain, the most promising one is probably the companion robot. A companion robot is an autonomous humanoid robot that can provide natural interactions with humans and interact with its environment. As it share the same characteristics than humans, the companion robot is made to help humans in domestic tasks.

For the French laboratory Lab-STICC (Laboratoire des Sciences et Techniques de l'Information, de la Communication et de la Connaissance) this problematic is of utmost importance. It is inside the RAMBO team (Robot interaction, Ambient system, Machine learning, Behaviour and Optimization) at Lab-STICC that the RoboBreizh initiative is born in 2019. This project is composed of PhD students, engineers and Master students and has been leaded for 3 years now by Cédric Buche, full professor at the École Nationale d'Ingénieurs de Brest (ENIB). The goal of RoboBreizh is to promote Artificial Intelligence research in robotics within 3 axes: Movement Learning, Perception and Human-Robot Interaction (HRI).

The present document resume my work for the RoboBreizh initiative this year from the 01/02/2021 to the 31/08/2021. The first work I have done is closely related to the RoboCup@Home project. This project took root last year after the victory of RoboBreizh team (of which I was part) at the RoboCup@Home Education Challenge, a small international competition of robotics. Me and the other members of the team then thought we could apply our knowledge and experience in a larger competition: the RoboCup@Home. Because the competition was scheduled in the last week of June this year I worked full time on the project from February to July.

1 https://ifr.org/ifr-press-releases/news/service-robots-record-sales-worldwide-up-32

Thanks to my works in Artificial Intelligence and Robotics I was recommended by my supervisor Cédric Buche for a PhD position in July 2021. Afterwards, I worked for the last two months on a proposal for this PhD. This proposal discuss about the literature overview related to the PhD topic, "Multimodal analysis and learning of human interactions by a companion robot: detecting and fulfilling user needs".

This document will be structured as follow: the first part will explained the RoboCup@Home competition and described my contribution inside the RoboBreizh team. The second part will detailed my PhD proposal and give a short literature overview related to it.

Part I

Domestic Object Manipulation: Inside the RoboCup@Home 2021 Competition

1 The RoboCup@Home Competition

1.1 Overview

The RoboCup@Home is an international competition part of the RoboCup initiative to promote robotics and AI research around the world. The RoboCup was founded in 1996 with one main challenge: create a robotic soccer competition in parallel to the human FIFA's WorldCup. 25 years later the initiative has evolved in no more than 6 majors leagues and 18 minors leagues outside of the scope of soccer. We can count for example the RoboCup Rescue league (focused on search and rescue robots), the RoboCup Logistics (factory robots) and the RoboCup@Home (domestic service robots). Even the main domain has changed, the ambition is still the same: promote and share the research in robotics to one day, being able to create robots that can compete with humans in a lot of different areas.

The RoboCup@Home was created in 2006 with a focus on personal domestic applications of service robotics. Challenges in RoboCup@Home typically take place in a small apartment where a companion robot has to fulfill human needs (e.g. Cleaning a room, finding an object, having a conversation...). Today the competition is divided into 3 minor leagues: the Domestic Standard Platform League (DSPL), the Social Standard Platform League (SSPL) and the Open Platform League (OPL). The first one uses the platform Human Service Robot (HSR) from the firm Toyota while the second one uses Softbank Robotics' Pepper robot. Finally, in the Open Platform League, teams are given the choice of the platform they want to use. Because the platform differs slightly in the three leagues, rules and regulation are not the same. Traditionally, the DSPL and OPL challenges focused more on object manipulation and navigation where the SSPL focused on Human-Robot Interaction (HRI), thanks to the social abilities of Pepper.

This year, due to the pandemic situation, the choice was made from the organizing committee to move the competition online. This decision highly impacted the rules and schedule of the competition. For instance, a choice was made to use a unique simulator environment for the three leagues. In the next

sections of this document, we will focus on the Social Standard Platform League as it is the one the RoboBreizh team choose to participate.

SSPL challenges focus on Human-Robot Interaction, Natural Language Processing, People Detection and Recognition, Reactive Behaviors and Safe Indoor Navigation and Mapping. A non-exhaustive list of the desired abilities is shown below:

- Navigation in dynamic environments
- Fast and easy calibration and setup, the ultimate goal is to have a robot up and running out of the box
- Object recognition
- Object manipulation
- Detection and Recognition of Humans
- Natural Human-Robot Interaction
- Speech recognition
- Gesture recognition
- Ambient intelligence, e.g., communicating with surrounding devices, getting information from the internet etc

1.2 Qualification

The competition is scheduled in two phases: the qualifiers and the competition week (also call "finals"). As the rules change every year the qualification process is independent from the actual competition. Traditionally, it started in October with a call of participation from the RoboCup@Home committee. Then, from November to March, teams can send their application materials as follow:

- A team Website
- A team description paper
- A qualification video

The team website should contain a description of the team, the name of the team members and a summary of relevant contributions. The team description paper needs to focus more on the technical approach to solve the main challenges of the competition (as listed above). Finally, the qualification video is supposed to show a robot solving some of the tasks of the competition. In addition, it needs to show the technical abilities described in the paper in a clear way, as a neophyte could understand it.

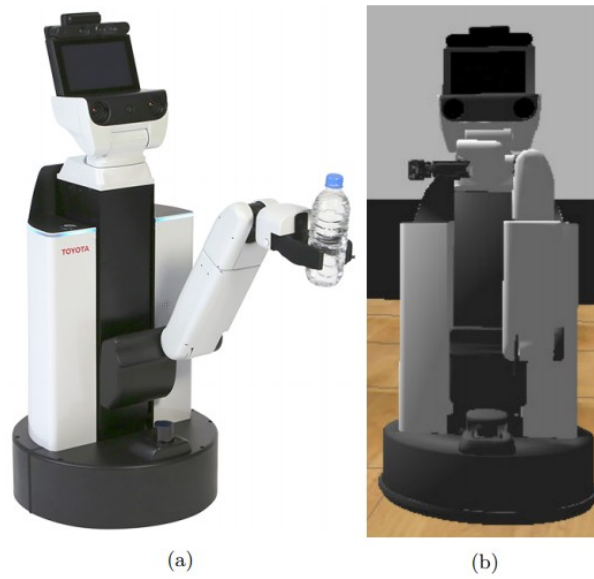

Figure 1.1: (a) The Toyota Human Support Robot (HSR) and (b) the Simulated model.

1.3 Environment

Due to its online form the competition took place in the simulated environment Gazebo via the open-source framework ROS (Robot Operating System). Gazebo is a simulator design for robotics application and already supports a lot of different robots like the Toyota HSR. It is also simple to use and flexible because of its compatibility with ROS. In fact, ROS is, more than a framework a way for robotics developers to build an application on a computer, test it with Gazebo and then deploy the exact same program on the real robot. Before this year, a lot of teams were already using this powerful tool for the previous editions, the choice of Gazebo was then logical for the organizing committee. For the SSPL, another problem was remaining: Gazebo doesn't have a model for the Pepper robot. Then, instead of trying to build and test a new model, the organizing committee chose to switch the platform of SSPL to the Toyota HSR, already in use by the DSPL. By this choice the social aspect of the league has been abandoned, and the challenges became more similar to DSPL. The original and simulated robot are shown in figure 1.1. A detailed description of the robot is shown in appendix C.

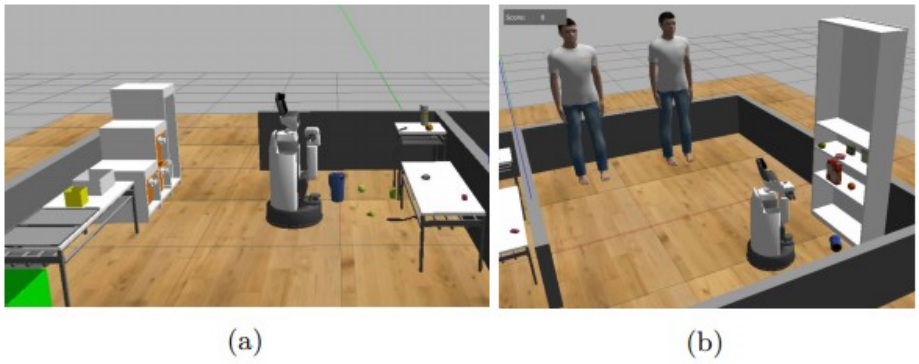

Figure 1.2: (a) The first room with the kitchen, (b) the second room with the shelf and the humans standing.

All the tasks are taking place in a single environment, created for the competition. It is composed of two rooms, one with a kitchen and two tables, the other one with a bookshelf and two static humans, see figure 1.2. The robot will start at the beginning of the first room and should navigate through it to go to the second room. A map of this arena is shown in figure 1.3.

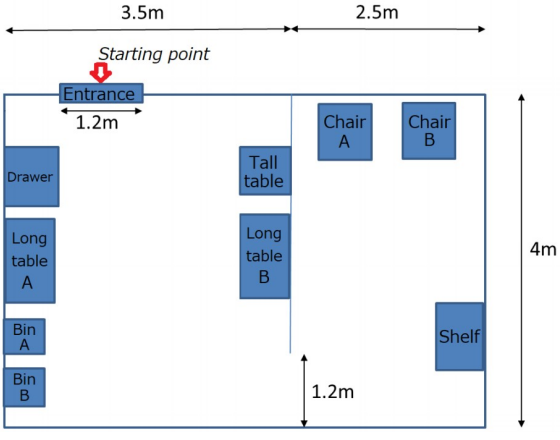

Figure 1.3: Map of the arena (starting point in red).

1.4 Schedule & Awards

The competition week is usually divided in two stages: in the Stage 1 all qualified teams compete when only the top 50% go to Stage 2. Each Stage is made of different tasks, each task can score points and it is the team with the maximum score on the 2 Stages that win the competition. There is a maximum points we can get per task so if two teams from Stage 1 in the middle of the leaderboard

scored almost the same points they will both be accept in Stage 2. Finally, at the end of the competition, there is also an award for the 2nd and 3rd team.

For Stage 1 each team will have 3 runs, distributed on the 2 tasks. For example one can do 2 runs for task 1 and 1 for task 2 or the opposite. This rule have been made to give a second chance to teams that encountered a simulator crash on one of their runs. The code for each task should be publish each day before the beginning of the competition and this code is run on the competition servers and display via a live stream.

1.5 Tasks & Scoring

Traditionally, there are around 10 different tasks in the competition (from Stage 1 and Stage 2). This year, due to the limitation of the simulated environment, only 2 tasks remain: Clean Up and Go And Get It. Stage 1 is composed of one run of each task while Stage 2 is only the Clean Up task, with an increased difficulty. Scoring is automatic and has been implemented directly in the simulator sources. A short overview of the two tasks is given in the following paragraphs.

1.5.1 Task 1: Clean Up

For the Clean Up task the robot needs to clean the first room (kitchen) by placing all misplaced objects to their corresponding location. It can be up to 30 different objects misplaced on the tables and the floor. For Stage 1 the time limit for this task is 5min, while for Stage 2 it is 15min. An excerpt of the score sheet from the 2021 rulebook is shown in figure 1.4.

#	Penalties				Points				
			Hit		Correct				
	Drop	Restart	Object	Furni-ture	Deli-very	Cate-gory	Orien-tation	False	Total
1	x0.5	x0.5	x0.5	x0.5	10	+10	+10	x0.0	
	Notes:								

Figure 1.4: Score for one object on the Clean Up task. The minimum score for one object is 0, maximum is 30 (delivery + category + orientation and no penalties).

1.5.2 Task 2: Go And Get It

For the Go And Get It task the robot should move to the second room. Then it should find on the shelf a specific object and bring it to one of the people while navigating through obstacles on the ground (i.e. small objects). See figure 1.5 for scoring.

Task 2a		
Successfully entering to the dining room without collisions	100	
Task 2b		
Taking any food item in the shelf	40	
Taking the requested object among many objects in the shelf	+60	
	SUM_1	
HIT_1: 1 2 3 4+		
	$Task\_2b_1 = (1 - 0.25xHIT_1)xSUM_1$	

Figure 1.5: Score for the Go And Get It task. The maximum score for this task is 200 + 50 bonus points (250) for finishing the task within the time limit.

2 Contribution: Qualifiers

As explained in section 1.2, three different materials need to be submitted for the qualification phase: a website, a paper and a video. My contribution was as follows: I created the required resources for the website, for the paper I wrote one page (two paragraphs) and I directed the full video (scenario + shooting + editing). It is important to notice that this step was supposed to show our already existing solutions related to the competition needs. We did not want to develop more as the time scheduled was already short before the qualification deadline. Furthermore, at the time we produce the qualification materials the complete rulebook of the competition was not available. That mean that our proposal was mostly based on the competitions from past years, except for the simulation part as we knew the competition will be online.

2.1 Team Website

Our team RoboBreizh already had a website before the competition. However, this website was not fulfilling the requirements for the competition. I redid the team page including the new members from LITIS (INSA Rouen). I also completely restructured the page explaining our research with a lot of diagrams, videos and interactive panels. The idea here was to make something interesting to read for the competition committee but also for other people that may want to learn more about the team. For instance, a french version of all the content on the website has been implemented. The website also includes the Team Description Paper and the Qualification Video [2].

2.2 Team Description Paper

Regarding the Team Description Paper (TDP) we decided to structure it upon four modules: Perception, Navigation, Movement and Interaction. After analysis

2 Link to the website: https://www.enib.fr/~robobreizh/

of other TDPs this structure seems to be the most relevant. We also choose to talk about the Simulation Environment I have built with Gazebo and Pepper earlier this year to test our solutions for navigation and perception. As a result, I wrote two sections in this paper: Navigation and Human-Robot Interaction (HRI). The full paper is available in appendix A.

2.3 Qualification Video

The Qualification Video was without a doubt the most time-consuming material to produce. First of all, some of our existing solutions had to be adapted to a simulation environment. Because the competition was online it was important for us to show that all our algorithms can run both in simulation and in real life. For the Perception part, I developed a visualization package to show on a real-time video (from simulation and reality) what we are able to detect. For the Interaction part I use the in-production resources from our dialog pipeline package.

The video scenario was structured as follow: first, a quick introduction presenting the team and partners ; second, a look into our solutions for Perception, Navigation, Movement and Interaction (in simulation and reality) ; third, an overview of those capabilities in two different scenarios taken from the 2019 rulebook of the competition: the Barman and Receptionist. The Receptionist scenario and the Speaker Recognition (Interaction part) were shot by LITIS in Rouen and the rest by myself and some of the RoboBreizh team members at ENIB (CERV building). Then, I did the editing of the video including a music and some visual effects to make it more enthralling. The final version of the video is 4:35min long and is available on YouTube [3].

3 https://www.youtube.com/watch?v=FCbEyJuy4IQ

3 Contribution: Solution for the competition

3.1 Overview

At the end of March 2021, a week after submitting the qualifier materials, we got a positive response from the organizing committee about our candidature. Then, from the beginning of April to the end of June, I worked on a solution for the competition. I got some help from Cédric Le Bono (PhD, IMT Atlantique) and Mihai Andries (Associate Professor, IMT Atlantique) for the choice of technologies to use but the full development was made by myself. Indeed, a lot of the members from RoboBreizh were unavailable at the time the competition started.

From the analysis of the rules (see section 1.5), two main abilities were needed: object recognition and object manipulation. For the two tasks of the competition we need to retrieve the labels of objects in the room and pick and place any object. The robot should also be able to navigate without colliding objects on the ground but this ability was not very important as it only provided bonus points.

3.2 Architecture

As explained in section 1.3 the simulator is built on ROS, so our solution should also use it. The main interest of ROS is its asynchronous communication model, using Subscribers and Publishers we can build a Topic that can process data from acquisition. It is also important to notice that ROS provides synchronous communication in a client/server form: we call it a Service. Finally, an independent and automatic process called Master is ruling those communications via TCP-IP protocol. A schema of this architecture is shown in figure 3.1.

Our architecture will be composed of 3 main packages (called Node in the ROS environment): one for the Object Detection, one for the Object Manipulation and one for the overall behavior called Manager (this last one will also include a basic navigation). Both the Object Detection and Manipulation package will communicate with the Manager using Services. We will retrieve data from the

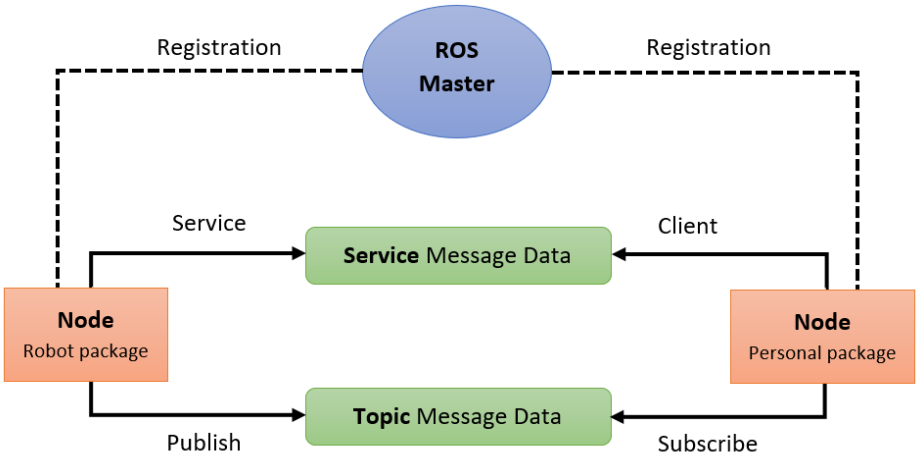

Figure 3.1: ROS communication for Topic and Service.

head RGB-D camera of the robot and send orders to the arm with dedicated topics.

ROS support two different programming languages: Python and C++. I have used Python for most of the solution because it is a language very flexible and easy-to-use. Unfortunately, some libraries and components of ROS were only available in C++ so I used this language as well for the Object Manipulation node.

3.3 Object Detection Node

In addition to correctly labelled all objects the Detection node also needs to compute their exact position in the 3D space. The complexity of the task here is to select a solution that provides coherent data to the Manipulation node. For instance, some algorithms for object manipulation require a 6D pose estimation of the object (3D pose + 3D rotation).

3.3.1 6D Pose Estimation Approaches

In practice, solutions for 6D Pose Estimation use RGB-D or RGB cameras, with better results for the RGB-D approach. To compare those solutions we will use

Figure 3.2: A set of 21 objects from the YCB Dataset - 3D models (Gazebo simulator).

the YCB-Dataset [Cal+15] as a reference. This dataset is composed of 80 000 images from 60 different objects of daily life (e.g. knife, can, sugar box...). All objects are small enough to be taken by a robotic arm such as the one from Toyota HSR. A set of 30 objects from this dataset, in a 3D models form, will be used for the competition (see figure 3.2). It is also important to notice that solutions using multiple cameras also exist. There are two cameras on the HSR robot, one on the head and one at the end of the arm (near the gripper). But using those two cameras for 6D Pose Estimation is quite difficult, because they do not share the same resolution and FOV (Field Of View). Also with the movement of the arm it will be difficult to track the 3D position of the arm camera (an exact pose is required in order to compare the two images). We then choose to test only single camera solutions.

The best solution for 6D Pose Estimation using RGB camera is a deep neural network (DNN) approach called DeepIM [Li+18]. This approach reports a 70% accuracy on the YCB-Dataset. There is actually only one implementation of the approach describe in [Li+18] and it is using a specific version of Python and PyTorch [4] that was complicated to integrate inside our ROS environment. As well as the other deep learning solutions, DeepIM requires the use of a good GPU for training and inference. This was a problem for us because our solution is supposed to run on some servers of the organizing committee that we do not

4 Deep Learning framework for Python, see https://https://pytorch.org/.

have access to. It is then a big risk to use this solution if the server does not have access to a GPU or the GPU is not powerful enough to run DeepIM.

For RGB-D approaches the best solution is MaskedFusion with a 93.3% accuracy [PA20]. Yet here the problem is the same as the approach is a DNN that needs a good GPU. Overall there are no other reliable solutions than DNN to the 6D Pose Estimation task for a simple reason: from a single camera view-point it is impossible to get an estimation of the orientation of the object without a massive training of images from this object in every orientation. The size of this kind of network also makes the inference computation time very expensive (from a few seconds to over a minute), far away from a real-time perspective. To resume, the idea behind 6D Pose Estimation approach was:

1. retrieving the label and 6D pose of an object
2. given the pose aligning the 3D model (downloaded offline in a separate file) and compute the best gripper position (i.e. grasp pose) on the object.

The reason why the 3D model of the object is needed is that with a single camera we can only see a maximum of 50% of the object so computing the grasp with those partial data will often result in an inaccurate or impossible grasp pose. This approach would have been the simplest and more accurate way of overcoming the challenge of the competition.

3.3.2 3D Object Detection Approach

Description

As the 6D Pose Estimation approach was not possible I decided to switch to a classical 3D Detection using the RGB-D head camera. To get the less computationally expensive approach I choose to first do a 2D object detection on the RGB image and then compute the depth value manually using the depth image. Then the problem was very easy as I only needed a solution to retrieve the bounding boxes and labels of all objects on an image. This task is called Object Detection and is one of the oldest and easiest tasks in computer vision. The most relevant solution for the competition needs was You Only Look Once (YOLO) [Red+16] because a version of this model, tiny-YOLO, doesn't require a GPU and is very fast. Actually this version of YOLO has been created for embedded devices, with a speed-up of 442% for a lost of accuracy of around 20% (data from the original

paper based on the COCO dataset, not YCB). YOLO has an architecture with a convolution base (CNN), the main difference between YOLO and tiny-YOLO is the number of hidden layers: 137 for YOLO and 29 for tiny-YOLO.

Training

To train tiny-YOLO I needed to label at least a few thousands of images from the dataset, a task that was impossible for me in the short delay. Instead, I used images from the pre-labbelled dataset from Kyouma Sun [5]. Unfortunately, this dataset contains labelled images from only 21 objects of the YCB dataset. I then tried to train the others objects with images from others dataset but it was not efficient. The main reason for this was the format of images between the YCB datasets and others dataset.

The training took 36h on my personal computer (GPU RTX 2060, 16go RAM) for an accuracy of 90% (mAP - mean Average Precision metric). I used a set of 93 755 images for training (all the labelled images) and 40 181 images for validation. This split is about 2/3 images for training and the rest for validation, the most common split in training of CNNs. Others parameters like the batch size have been chosen according to the YOLO documentation [6]. I stopped the training manually after 30 000 iterations when the loss function was no longer decreasing for a few consecutive hours, see figure 3.3. This loss function compute the difference between (1) the center of the predicted bounding box and (2) the dimension of the predicted bounding box with the actual bounding box.

Figure 3.4: Inference on two objects of the dataset, inside the simulator. Caption of the bounding boxes: object_class [accuracy of the prediction].

5 https://kyouma9s.com/ycb-video-dataset-download-mirror/#more-48
6 https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects

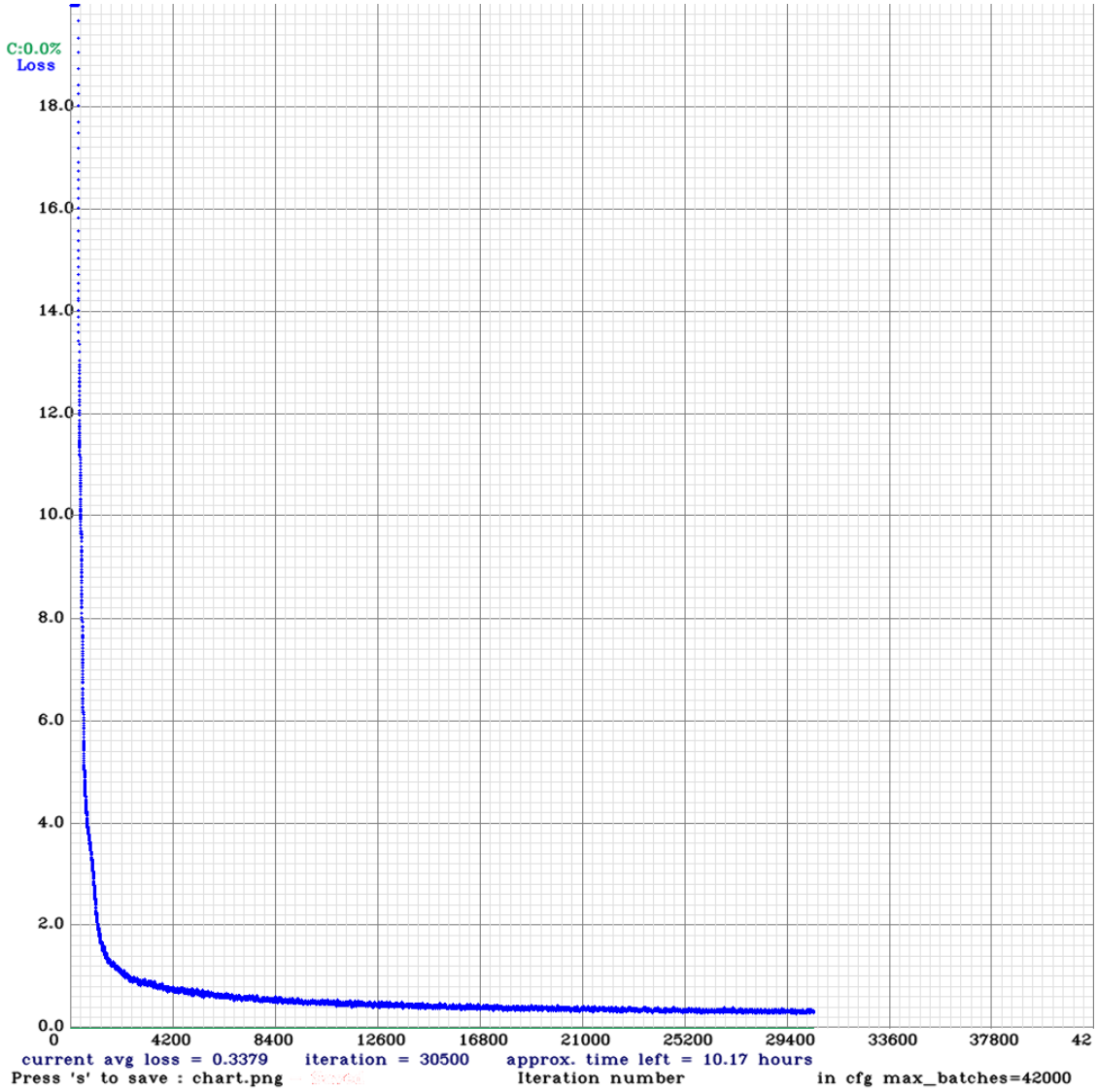

Figure 3.3: Chart graph of the average loss (in blue) for the training of tiny-YOLO (30500 iterations).

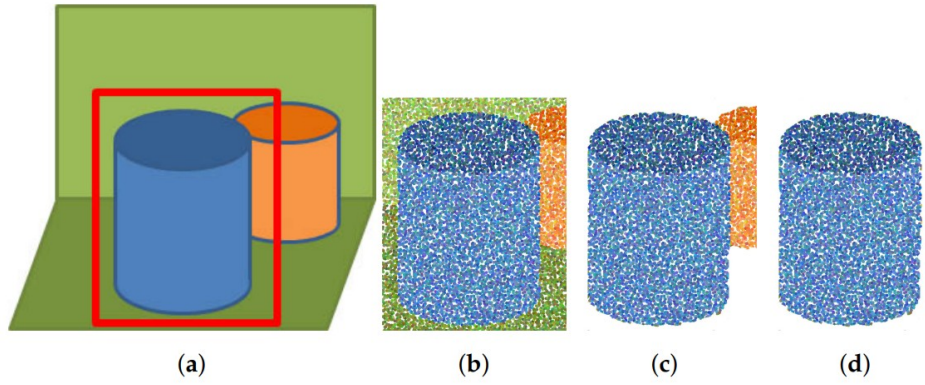

Figure 3.5: (a) 2D image, (b) cropped 3D image (pointcloud format), (c) first clustering: we remove the plane background, (d) second clustering: we remove the outer clusters (smaller objects). Source: [Yi+20a]

Unfortunately, in the simulator environment the results were slightly different, mostly because the 3D models differ from the original objects (see figure 3.4). I do not have metrics to show the difference between performance in simulation and reality but from my experiments it is a lost of 20-30% accuracy.

3.4 Object Manipulation Node

3.4.1 Grasp Pose Detection

For the Manipulation node the challenge was to compute an accurate grasp pose given only a bounding-box of the object on a 3D image. First problem was the background, as you can see in figure 3.4 the bounding boxes sometimes contain a big proportion of background (table or wall) that the system will not be able to differentiate from the actual object. The idea is then to segment the 3D image and remove the useless pixels. The method I used is inspired by the one described in [Yi+20a], see figure 3.5.

After cropping the 2D image (a), we interpolate the pixels with the depth image to recreate the 3D image (b). Then segmentation is performed in two steps: first we remove the planar surfaces using Ransac algorithm (c), second we do a clustering to remove outliers (d). To remove horizontal and vertical planar surfaces the Ransac algorithm is applied two times: one time on the z axis

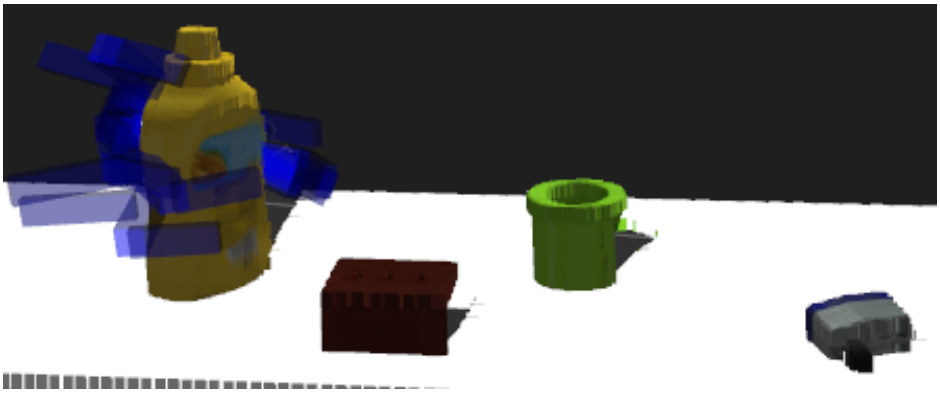

Figure 3.6: Grasp pose estimations (in blue) of the left object only (mustard bottle) using our solution.

(horizontal) and one time on the x axis (vertical). Before removing the vertical plane we check if it size is inferior to 50% of the image to not remove the object if this one contain a planar surface. This methodology is inspired by the Plane Model Segmentation from the PointCloud Library [7].

Then, to compute the grasp pose (i.e. the position of the gripper or end effector in a 6D space), we can not use conventional approaches as described in section 3.3 because we are missing the 3D rotation of the object. Fortunately, other solutions exist like Grasp Pose Detector (GPD) [Pas+17]. The idea behind GPD is simple: given a 3D image in a point cloud format and the dimension of an end effector it will compute all the possible grasp positions. Then, a trained neural network will rank those positions from most likely successful to less likely successful. Because this network has been trained with a lot of objects it is able to find a good grasp pose even with partial data (i.e. only the portion of the object that is visible). An example is shown in figure 3.6.

This solution is not perfect, for example due to the clustering step the algorithm does not take into account the surroundings of the object (other objects or table), resulting in potential collisions. To overcome this issue I apply a filter on the generation phase to keep only positions in a range center on the axis between the camera view-point and the object. This way the algorithm will compute only positions where the arm is close to be aligned with the camera-object axis. GPD

7 https://pointclouds.org/documentation/tutorials/planar_segmentation.html

has a lot more parameters that we can tweak like the number of samples for the generation. An overview of the parameters file used during the competition is available in appendix B.

3.4.2 Manipulation

Once a valid grasp pose has been found, the system execute the arm movement that results in the desired position for the end effector. To do so, it needs to solve the Inverse Kinematics equation. A robot arm like the HSR's one is made of multiple joints, in this case 6. Given a set of original positions Σ of all the joints we want to compute a final set of positions Δ where the end effector is at the desired position. Inverse Kinematics is a mathematical process that can compute Δ by using Σ and the constraint for each joint (e.g. the tip joint can only rotate for 50° on the x axis).

In mechanics, we represent a joint as a Degree of Freedom (DoF), usually the Inverse Kinematics equation becomes difficult to solve with a number of DoFs of 7 and more. In our case the number of DoFs of the HSR arm equals 6, 5 rotations and 1 translation (see figure 3.7).

To find and compute the Inverse Kinematics we use the kinematics solver BioIK [Sta+16] inside the package MoveIt! [Col+14]. MoveIt! is a ROS package that implement solutions for manipulation on several different robotic platforms. More than the Inverse Kinematics solver, MoveIt! integrate an efficient control and planning algorithm to send orders to each joint. This algorithm will compute the require force to apply to move one joint from an initial position to a desired position. Our solution uses MoveIt! out-of-the-scope with only a small parameters tweaking for a smoother movement. To get a better precision, I decomposed the movement in two phases: first a fast "pre-movement" is per-

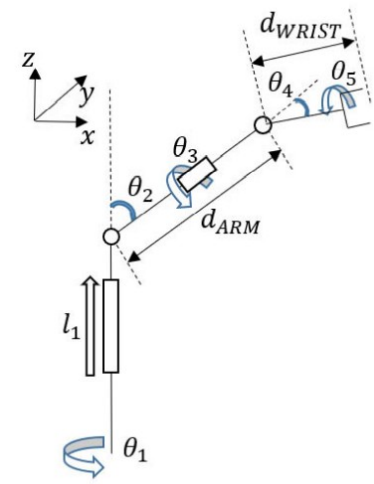

Figure 3.7: Arm configuration of the HSR robot: rotation on z axis θ_1, rotation on x axis θ_3 and θ_5, rotation on y axis θ_2 and θ_4 and translation on z axis l_1.

formed at 10cm of the object on the arm-trajectory axis, then the final movement is executed slowly close to the object. This decomposition of the movement increase the success rate while keeping a good speed of execution. When the arm has reached the desired position with MoveIt!, we send an order to close the gripper and grab the object. For this step it is MoveIt! controller that will stop the gripper when the mechanical constraint on the joint is increasing too much (to not destroy the object while trying to grab it). Finally we move back the arm to the initial position (arm close to the torso) for an easier navigation.

3.5 Manager Node

The Manager node implements a different behavior for each task. While some components remain the same (e.g. communication with other nodes), the main program differ. The navigation is integrated in this node as it is a in-built component or ROS: the ROS navigation stack [Mar+10]. A map of the arena with a coordinate system was given with the simulator to be used with this navigation stack. The map displays the walls, shelves and deposits areas that the navigation stack is able to interpret to send order to the wheels controllers of the robot. This stack also implements a position tracking of the robot using Odometry [8]. At an high-level our program only needs to send a position (x, y) to the navigation stack and the robot will moves to the position avoiding known obstacles.

3.5.1 Task 1

For the first task (Clean Up) the behavior is structured as a state-machine where the states are as follow:

- INIT:
  The robot move to the initial position in the center of the room: switch to state LOOK
- LOOK:
  The robot move the head until it detect an object using the Detection node:

8 "Odometry is the use of motion sensors to determine the robot's change in position relative to some known position." Source: https://groups.csail.mit.edu/drl/courses/cs54-2001s/odometry.html#:~:text=Odometry%20is%20the%20use%20of,how%20far%20it%20has%20traveled.

Figure 3.8: Our robot grasping an object during Task 1.

- If multiple objects are detected the system compare the distance from the robot to each object and pick the closest one as the one to grasp: switch to state GRASP
- If no object are detected after a countdown the robot move 30cm toward: switch to state LOOK

- GRASP:
  - If the object is beyond the reach of the arm we navigate toward the axis robot-object until we reach a better distance, then performed the grasp using the Manipulation node (figure 3.8):
  - If the grasp is successful: switch to state DEPOSE
  - If the grasp is unsuccessful (the object drop): switch to state GRASP

- DEPOSE:
  Go to the deposit and open the gripper to let the object fall into the deposit: switch to state INIT.

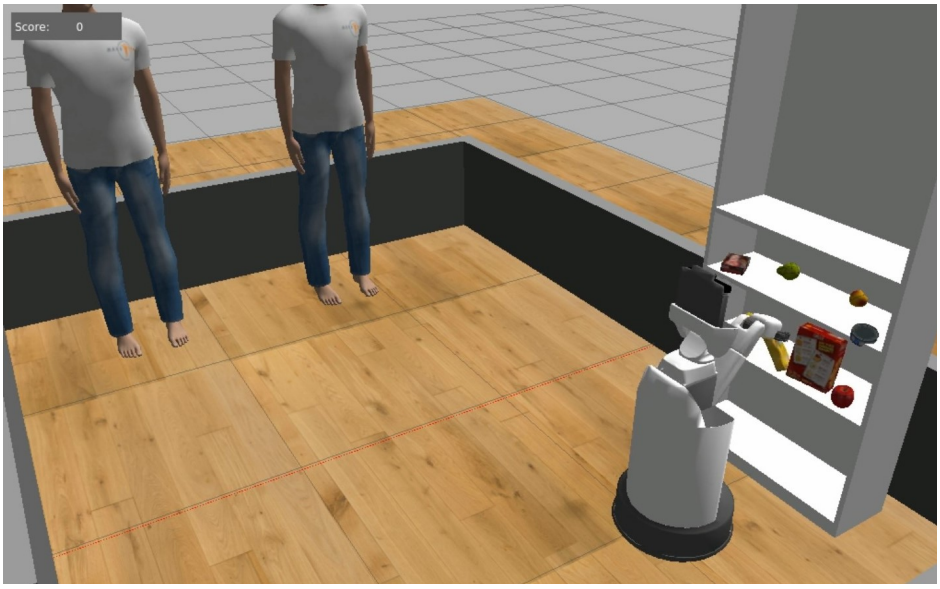

Figure 3.9: Our robot taking the object in the bookshelf during Task 2.

3.5.2 Task 2

For the second task (Go And Get It) the scenario is sequential so a state machine was unnecessary. The main challenge for this task was the navigation. When the robot enters the second room there are some small objects randomly scattered on the ground that the robot needs to avoid. Those obstacles are chosen on purpose too small to be detected by the lasers of the robot. My idea was then to use the GPD algorithm to detect grasp poses on the whole image and then cluster those positions to retrieve the position of obstacles. Of course this solution was not the most efficient but it was the simplest to use at this point. Given the positions of obstacles it was easy to write a basic navigation controller.

Then, once the robot was in front of the bookshelf we used the Detection node to detect and the Manipulation node to pick the requested object (figure 3.9). If we are not able to detect it, we pick instead a random object as it still grants points. The last step of bringing the object to the person is a classical navigation as in Task 1.

4 Result and Outcome

4.1 Performance

Thanks to our solution, RoboBreizh team scored 3rd overall at the competition. In the first Stage we scored 2nd with a total of 160 points (10 points on Task 1 and 150 points on Task 2). In Stage 2 we scored 3rd with a total of 20 points (Task 1 only), see figure 4.1.

Our performance on Task 1 was not good because of the random position and orientation of objects on the ground and the two tables. GPD was not able to find a good grasp pose without colliding surrounding objects and we lose points. The number of objects (21) that we were able to detect was another problem, as the competition used a set of 30 objects. This plus the lack of accuracy on the detection (due to the training on real images) result in a correct detection of 30% to 40% of the objects.

In Task 2 we were able to correctly take the object but the navigation failed in some runs (collision with objects on the ground), due to a wrong interpretation of obstacles position in some cases.

	Place	Team Name	Stage I			Stage II		Final
			Clean Up	Go Get-it!	**Total**	Clean Up + Go Get-it!	**Total**	
SSPL	1	UChile Peppers	30	250	**280**	20	20	**300**
SSPL	2	LyonTech SSPL	–	100/100/40	**100**	110	110	**210**
SSPL	3	RoboBreizh	10	150/100	**160**	20	20	**180**
	–	TJArk@Home				(Drop)		

Figure 4.1: Final competition board (SSPL).

4.2 Encountered issues

The main issue I encountered during the competition was the lack of information from the organizing committee. As this year was the first attempt on a virtual only competition, the organizing committee members were unprepared and the rules took a lot of time before being completely established. Most of the time it was team members that were reporting bugs or issues to the organization to adapt the rules.

Another recurrent problem was the simulator: as this one was not designed for the competition the support was difficult to reach. For example, changes have been made on the simulator until 2 weeks before the competition.

Finally, as we were running the simulator on our own machines for development and testing, we were unable to preview the result of our solution on the competition's servers. For instance, during the competition I report an important change of frame per seconds that modify the physics of objects.

4.3 Outcome and future works

After the competition RoboBreizh was congratulated by the RAMBO team, the ENIB and Lab-STICC. A few press articles have been written on our performance [6] [7] [8].

For everyone this performance was encouraging as it was our first time attending the competition, unlike most of the other teams. The competition was also a really good opportunity for us to get in touch with other robotics research laboratories around the world. We are planning to do a workshop with two other teams (UChile - Chili and LyonTech - France) to discuss our solutions and the future of the competition.

Finally, our performance has encouraged the leader of the team, Cédric Buche, to participate again next year in the already scheduled competition in Bangkok, Thailand. With my experience from this year, I will be part of this new initiative.

6 https://www.afran.org.au/single-post/a-french-and-australian-collaboration-for-the-robocup
7 https://www.letelegramme.fr/finistere/plouzane/l-equipe-robobreizh-de-plouzane-termine-troisieme-au-concours-mondial-de-rob
php
8 https://www.ouest-france.fr/bretagne/brest-29200/brest-l-equipe-bretonne-robobreizh-termine-troisieme-a-une-competition-inter

Part II

From Perception to Learning: A Literature Overview

5 Project Proposal

5.1 Context

Thanks to my works in Artificial Intelligence and Robotics I was recommended by Cédric Buche for a PhD position in June 2021. This project is a collaboration from the École Nationale d'Ingénieurs de Brest (Brest, France) and the Flinders University (Adelaide, Australia). It is part of the French-Australian initiative AFRAN and the international laboratory CROSSING in Adelaide, Australia. The proposal for this position was established by myself, Cédric Buche and Paulo Santos (Professor at Flinders University). The literature overview in chapter 6 was conducted only by myself. This version is not the final state of the art for this proposal but a first literature overview. The initial purpose was to support the project for funding within the Flinders University.

The proposal title is: "Multimodal analysis and learning of human interactions by a companion robot: detecting and fulfilling user needs". The idea behind this proposal is that human-robot behaviors can be learnt from the observation of human-human interactions. In this perspective, we are particularly interested in the companion robot platform as it is the one that share the most characteristics with humans.

5.2 Approach

The understanding of human-human interactions by a companion robot can be related to the Symbol Grounding Problem as it is described by Stevan Harnad in [Har90]. By definition, the Symbol Grounding Problem could be summarized as "how to infer the meaning of a symbol (e.g. a word or group of words) from perceptual data". An example could be: how can we ground the action "waving the hand" with the words "asking for help"? In the earlier experiments in this area (i.e. Searle's "Chinese room argument" [Sea80]), symbols was referring to abstract concepts but a new definition has been proposed by Paul Vogt in "The Physical Symbol Grounding Problem" [Vog01]. Vogt define the Physical Symbol

Grounding as "the grounding of symbols to real world objects by a physical agent interacting in the real world". Our scenario (i.e. human-human interaction in a home environment) will use those two definitions, the first one to ground abstract concept such as "the need of help" and the second one to ground physical entities (e.g. objects, person) in the real world.

Thus the analysis of human-human interaction could be visualized as a dual symbol grounding problem where the system should understand the meaning of human behavior with its surroundings using multi-modal sensors data.

In order to obtain an effective and natural HRI, it is of utmost importance for the artificial agent to be able to understand and mimic human-human interaction. To this end, machine behaviour models can take direct inspiration from human social actions. A first challenge that motivates this research is the multimodal analysis of human interactions. Human typically interact using 5 modalities: sight, hearing, touch, smell, and taste. In Robotics, modalities can be extend to interaction channels such as accelerometer, depth camera or LIDAR. Each of those modalities can be used to obtain different data: for example, using a single RGB camera we can infer body poses and facial expressions. A multimodal system is a system that take multiple data as input. For a robotic platform the most reliable sensor source is the RGB-D camera (Red, Green, Blue and Depth data). It is then important for us to restrict our research to a computational perspective where only non-verbal, visually-observable, behaviors are taken into account. However, analysing human interactions is more complex than just reading sensors data, since human behavior goes beyond simple action-reaction mechanisms. It is for instance known that culture or emotions can have an impact on how people engage in social interactions [WKM93]. To address this issue, the use of commonsense reasoning or external labelled data could be considered. Finally, use of external sensors like a smartwatch or smartphone is a possibility as it could provide biological information (e.g. heartbeat) to infer emotions [Chu+19].

To summarize, given a sequence of 3D images (i.e. RGB-D video) the goal of the system is:

At a low-level retrieve data such as body key-points, face key-points, distances or gaze tracking.

At a feature-level process those data to obtain pose estimation, emotion analysis, actions being performed, instance segmentation or objects detected. We

can also feed the system with additional input such as labeled data on mental state or intentions of actors.

At a high-level we want two things: the actors intentions and the physical movements needed to provide help. In the present context, this manifests itself in learning intentions and affordances in human-human interactions from a robot sensor's standpoint.

A second challenge in this project comes from the complexity involved in faithfully reproducing the movements of a human by a robot, as the latter and the former do not share the same body shapes. Thus, the effective reproduction of human actions by a robotics agent is one important scientific question this work aims to solve. The main paradigm of this question is whether to learn the exact required movements to complete the action without any context understanding or learn the main goal and steps of the action and then compute robotic movement to achieve it. A representation of this architecture is described in figure 5.1.

Our proposal can be divided into three different goals as follow:

Objective 1: The multimodal analysis of human-human interactions: Detecting user need of help.

Objective 2: The multimodal analysis of human-human-object interactions: Understanding assisting action(s).

Objective 3: The effective reproduction of human actions by a humanoid robot.

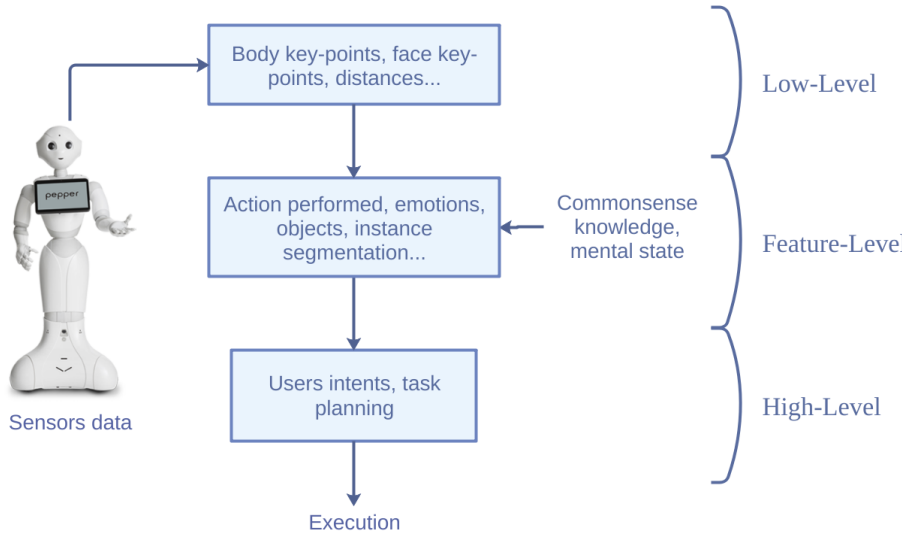

Figure 5.1: System perception-decision-action loop.

6 Contribution: Literature Overview

We want to analyse human-human interaction where one person is asking for help from another. In those interactions we should be able to extract the relevant data with the meaning "asking for help" from the first person and the relevant data for "providing help" from the second one. Then the robot should be able to reproduce the behavior of the second person (i.e. the help provider) using its own capabilities differing from the human ones. This section will be divided into three parts:

- In the first part we will review the state-of-the-art related to the low-level data we want to retrieve (i.e. body key-points, face key-points, hand-crafted features).

- In the second section we will review the literature at a features level. This section mostly covers classifying and labelling tasks to infer the mental state or actions being performed by actors. It also presents an overview of Object and Person Recognition in Computer Vision.

- Finally we review the state-of-the-art for the high-level decision-making solutions. This includes mutli-modal fusion, commonsense reasoning and decision-making but also planning the reproduction of human actions by a humanoid robot.

6.1 Low-Level

At a low-level, we retrieve hand-crafted features directly from images input. To detect those features SIFT descriptors [Low99] or optical flow [YYL12] could be used, with the drawback of the camera sensitivity to motion or dynamic background. Others solutions use bag-of-words (BoW) or Fisher vector [Gao+16] to increase the descriptors robustness. To isolate a specific feature like a person, Histograms of Oriented Gradients (HOG), Histograms of Oriented Flow (HOF) or Motion Boundary Histograms (MBH) are used [HNG16]. For the body pose, a new and promising approach is OpenPose [Cao+17], a Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. This solution can detect up to

135 body keypoints on an unlimited number of people, in real time. For facial expressions, OpenFace [Bal+18] has become a standard since 2018 as it is able to accurately detect face landmarks and poses in real time. The landmarks detection of OpenFace is based on the Convolutional Experts Constrained Local Model (CE-CLM).

6.2 Feature Level

6.2.1 Classifying Human Action

In our proposal, it is of utmost importance to be able to classify the current interaction being performed by the actors. We call this task Human Action Recognition (HAR). We are particularly interested in classifying the "Asking for help" and "Providing help" interactions. HAR has been leveraged with the use of a wide range of Computer Vision' Deep Learning solutions. Those solutions are trained with a lot of different datasets, like the Atomic Visual Actions (AVA) [Gu+18] and Kinetics [Car+19] datasets. All those datasets integrate shorts (10-15s) to long (15-30min) videos annotated with the current interaction being performed. Some also add human pose data. In Computer Vision, the widely known approach is the use of Convolutional Neural Networks (CNNs) [LeC+98]. CNN has been used for Visual Detection and Classification on RGB videos. Methods for human-human interaction classification use CNNs on a single frame of the video [GGM15] and extend it to sequences of frames using for example Transfer Learning [Ben12]. More advanced methods use Residual Networks (ResNet) [He+16] or Recurrent Neural Networks (RNNs) [Bag+17]. Finally, the latest solutions express the need of Long Short Term Memory (LSTMs) RNNs [Zhu+16], sometimes in combination with other methods [Liu+16]. Most of those methods implement tracking of the skeleton data to infer the type of interactions from human poses.

6.2.2 Labeling objects and zones of interest

In Computer Vision, Object and Person Detection has been the most studied problem from years [POP98]. Work from Google [STE13] described Object Detection as "a regression problem to object bounding box masks" and highlighted the importance of Deep Neural Networks (DNNs) in this field. In fact, nowadays, the most common approaches are You Only Look Once (YOLO) [Red+16] with the

use of CNN and Mask R-CNN [He+17] with the combination of Residual Network (ResNet) and CNN. Latest solutions also use Transformers-based architecture [Dai+21]. Main difference between YOLO and Mask R-CNN is the use of masks generation in the second one. Using instance segmentation on the detected bounding boxes, Mask R-CNN approach extract pixel-wise masks of objects. This is notably very important to have a precise understanding of object position for planning and manipulation. Object Detection and Scene Understanding tasks reached a new milestone with Semantic Segmentation. Semantic Segmentation is the task of segmenting an image by semantic zones (i.e. zones of interest). Latest solution for Semantic Segmentation uses Transformers architecture for dense prediction [RBK21]. The use of Transformers here leverages the state-of-the-art CNN solutions thanks to the attention mechanism that computes all sections of the image together, instead of doing it sequentially.

6.2.3 Infer emotions

Emotions and sentiments play a big role in human decision making [LL03]. Emotion Recognition is an important area of research to enable effective human-machine interaction. Human emotions can be detected using speech signals, facial expressions, body language, and electroencephalography [ULQ17]. Based on our proposal, only the visual input solutions (facial expression and body language) will be reviewed. In [Ko18], Byoung Chul Ko reviewed the solutions for Facial Emotion Recognition (FER). Handcrafted-features and deep-learning-based approaches are compared, with a net advantage on deep learning solutions (+ 10% accuracy according to this review). Handcrafted solutions mostly rely on SVM while deep learning approaches use CNN and LSTM models. Lately, the use of the attention mechanism for FER has improved the state-of-the-art results [MMA21] on different datasets (the FERG, JAFFE and CK+ datasets). The proposed model creates blank masks on random regions of the image to learn which one is important.

6.3 High-Level

6.3.1 Multi-modal fusion

The use of multi-modality in the understanding of human behavior has been widely studied [SMD12]. The main goal of this approach is to find correlation

between modalities that will improve the accuracy of a prediction. Furthermore, in [Sal+12] the authors point out the multi-modality as a solution in vague interaction (i.e. interaction where some of the signals are not relevant). The gap here is to build a system that is able to choose how to complete the vague information from one modality with another one.

In [Tur14] Matthew Turk reviews different solutions to design and build multimodal systems. One of the key questions in multi-modality design is the choice between two approaches: early fusion or late fusion. Early fusion will typically fuse sensor data directly from acquisition. Late fusion will fuse the result of processing from each modality at a decision level. A good example is the work from Gunes et al. [GP05] that compares early and late fusion for body pose and facial expression. Different methods exist for fusion, in [Atr+10] the authors list rule-based (e.g. Linear weighted fusion) and classification-based (e.g. Support Vector Machine, Bayesian Networks). But the most used method nowadays is neural network fusion as in [SOG19]. For this kind of approach Mixture of Experts (MoE) could be used. Finally, [Tur14] explores other parameters for integration of multimodal systems. For instance, parallelism or sequentiallity of such systems are reviewed.

6.3.2 Learning from human

The task of inferring intentions and mental states in human-human interaction is called Visual Commonsense Reasoning. In [Zel+19] the authors highlight the difficulty of this task compared to visual recognition tasks. There is different types of visual reasoning:

- Temporal (what happened before, what might happened next)
- Spatial (where is this object located?)
- Emotional (what is the first person relationship with the second person?)
- Mental (what is this person thinking?)
- Causal (why is this event happening?)
- Hypothetical (what would happen if... ?)

To address those questions different approaches have been studied. In [Yu+19] Heterogeneous Graph is used to infer relation between objects. Transfer learning is also used in a multi-level knowledge network in [WP20]. Furthermore, what all of those approaches state is the difficulty of retrieving a rational explanation from a prediction. The idea is then to form a couple with the predicted answer

(A) and a rational explanation (R) in the learning process. This way the system is able to justify its decision.

The second learning process of our proposal will be the learning of human action from a robot perspective. Multiple approaches exist to learn body motion from humans: demonstration and reinforcement learning for instance. [Pas+09] describe the learning by demonstration methods in robotics. Learning by demonstration is useful to rapidly learn a movement or sequence of movements from a few demonstrations. The issue with this approach is the context-dependence as the movement is learnt without learning the actual goal of it. Another study reviews the reinforcement learning solutions in robotics [AL20]. In Reinforcement Learning, thanks to a simple goal-reward mechanism the robot is able to compute its own movements. The issue here is to be able to understand the precise goal of the action.

6.3.3 Motion Planning

In motion planning for manipulation, MoveIt! [Col+14] has become the standard in the robotic community. MoveIt! is a robotic manipulation framework with different capabilities:
- trajectories planning, even with high degree of freedom arms
- inverse kinematics computation
- collisions avoidance (with the use of RGB-D camera)
- low-level precision control
- adaptability to a lot of platforms (thanks to ROS - Robot Operating System).

For this proposal, MoveIt! seems to be one of the best solutions for motion planning and execution of the action, mostly because it is easily adaptable to any robot. If we want to use another approach we will have to adapt the design to our specific platform (degree of freedom and joint capabilities are not the same for different robots). For custom solutions, Rapidly-exploring Random Tree (RRT) and Probabilistic Roadmap (PRM) have been wwidely used [KF11]. Latest approaches describe neural-network solutions [Qur+19] to address the computational-intensive issue of RRT and PRM approaches.

Conclusions & Outlook

This internship has brought great experiences and has been a source of many opportunities. Amongst those experiences there have been some challenges. My participation in one of the biggest international competitions in robotics was testing, both the time scheduled I was put under and the difficulty of the task. Fortunately, I already had a lot of experience with the technologies involved - ROS and Gazebo - due to my previous work with the Lab-STICC. On the other hand, I was completely new in the Computer Vision and Object Manipulation domains. For the choice of the solutions in Computer Vision I received decisive help from Cedric LeBono (PhD student at IMT Atlantique) and in Object Manipulation by Mihai Andries (Associate professor, IMT Atlantique). Moreover, this competition was an occasion for me to meet researchers in robotics all around the world. I talked with team leaders in Chili or France but also the organizing committee members from China and Japan.

As an additional work, I have submitted a paper with the RoboBreizh team in arXiv [9], a platform for non-reviewed articles. This paper is a summary of our achievements with the RoboBreizh team last year in the RoboCup@Home Education Challenge. This was for me the first scientific publication writing and I learned a lot about syntax, rules and grammatic structures for this task. This paper is available at the following address: https://arxiv.org/abs/2107.02978.

In the meantime, I also supervised a Master student (first year) for a two months internship from May to June. This intern (Thomas UNG, Université de Bretagne Occidentale) was working on a Navigation algorithm in Gazebo simulator with ROS, two technologies that I was familiar with. I have been asked by my supervisor Cédric Buche to help him as he was completely new in robotics and Artificial Intelligence. This was the first time I supervised another student, I scheduled meetings every week to help him when he was struggling.

At the end of my internship and thanks to my performance at the RoboCup@Home, I have been offered a PhD position inside the new initiative CROSSING in Adelaide, Australia, starting in October 2021. This position is a co-tutelle

9 https://arxiv.org/

between the French school ENIB and the Australian Flinders University. It is for me particularly interesting because it is highly related to my previous works and I had the opportunity to discuss and reformulate the proposal with the supervisors (Cédric Buche and Paulo Santos).

In conclusion, this internship was for me a great way to end my degree as it provided me with a strong experience of being a researcher in robotics and AI. Due to the current pandemic situation, I have been working mostly in distance for the competition phase as I was working in simulation. For the qualification phase I spent some time at the European Center of Virtual Reality (CERV, Plouzané) to use the real robots for the Qualification Video. This in-distance work was not a big issue for me as I was used to talking in distance with my supervisor in Australia before. All the meetings with the RAMBO and RoboBreizh team were also already scheduled in distance.

I have currently 3 future works in progress: first my PhD starting in October, second the participation to the RoboCup@Home competition next year (in the organization or as a participant with RoboBreizh) and third the writing of a new article about my research in Human-Robot Interaction (HRI) for the past three years. In fact, I have been building a new type of dialog pipeline for ROS including a lot of state-of-the-art Natural Language Processing (NLP) tools but also some newly developed architecture like the use of Commonsense reasoning for next command prediction.

Bibliography

[AL20] Neziha Akalin and Amy Loutfi. **Reinforcement learning approaches in social robotics**. *arXiv preprint arXiv:2009.09689* (2020) (see page 35).

[Atr+10] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. **Multimodal fusion for multimedia analysis: a survey**. *Multimedia systems* 16:6 (2010), 345–379 (see page 34).

[Bag+17] Timur Bagautdinov, Alexandre Alahi, François Fleuret, Pascal Fua, and Silvio Savarese. **Social scene understanding: End-to-end multi-person action localization and collective activity recognition**. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 4315–4324 (see page 32).

[Bal+18] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. **Openface 2.0: Facial behavior analysis toolkit**. In: *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE. 2018, 59–66 (see page 32).

[Ben12] Yoshua Bengio. **Deep learning of representations for unsupervised and transfer learning**. In: *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings. 2012, 17–36 (see page 32).

[Cal+15] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. **Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set**. *IEEE Robotics & Automation Magazine* 22:3 (2015), 36–52 (see page 14).

[Cao+17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. **Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields**. In: *CVPR*. 2017 (see page 31).

[Car+19] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. **A short note on the kinetics-700 human action dataset**. *arXiv preprint arXiv:1907.06987* (2019) (see page 32).

[Chu+19] Seungeun Chung, Jiyoun Lim, Kyoung Ju Noh, Gague Kim, and Hyuntae Jeong. **Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning**. *Sensors* 19:7 (2019), 1716 (see page 28).

[Col+14] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. **Reducing the barrier to entry of complex robotic software: a moveit! case study**. *arXiv preprint arXiv:1404.3785* (2014) (see pages 20, 35).

[Dai+21] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. **Dynamic Head: Unifying Object Detection Heads with Attentions**. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 7373–7382 (see page 33).

[Gao+16] Chenqiang Gao, Luyu Yang, Yinhe Du, Zeming Feng, and Jiang Liu. **From constrained to unconstrained datasets: an evaluation of local action descriptors and fusion strategies for interaction recognition**. *World Wide Web* 19:2 (2016), 265–276 (see page 31).

[GGM15] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. **Contextual action recognition with r* cnn**. In: *Proceedings of the IEEE international conference on computer vision*. 2015, 1080–1088 (see page 32).

[GP05] Hatice Gunes and Massimo Piccardi. **Affect recognition from face and body: early fusion vs. late fusion**. In: *2005 IEEE international conference on systems, man and cybernetics*. Vol. 4. IEEE. 2005, 3437–3443 (see page 34).

[Gu+18] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. **Ava: A video dataset of spatio-temporally localized atomic visual actions**. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, 6047–6056 (see page 32).

[Har90] Stevan Harnad. **The symbol grounding problem**. *Physica D: Nonlinear Phenomena* 42:1-3 (1990), 335–346 (see page 27).

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. **Deep residual learning for image recognition**. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778 (see page 32).

[He+17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. **Mask r-cnn**. In: *Proceedings of the IEEE international conference on computer vision*. 2017, 2961–2969 (see page 33).

[HNG16] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. **Fast temporal activity proposals for efficient detection of human actions in untrimmed videos**. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 1914–1923 (see page 31).

[KF11] Sertac Karaman and Emilio Frazzoli. **Sampling-based algorithms for optimal motion planning**. *The international journal of robotics research* 30:7 (2011), 846–894 (see page 35).

[Ko18] Byoung Chul Ko. **A brief review of facial emotion recognition based on visual information**. *sensors* 18:2 (2018), 401 (see page 33).

[LeC+98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. **Gradient-based learning applied to document recognition**. *Proceedings of the IEEE* 86:11 (1998), 2278–2324 (see page 32).

[Li+18] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. **Deepim: Deep iterative matching for 6d pose estimation**. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, 683–698 (see page 14).

[Liu+16] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. **Spatio-temporal lstm with trust gates for 3d human action recognition**. In: *European conference on computer vision*. Springer. 2016, 816–833 (see page 32).

[LL03] George Loewenstein and Jennifer S Lerner. **The role of affect in decision making.** (2003) (see page 33).

[Low99] David G Lowe. **Object recognition from local scale-invariant features**. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, 1150–1157 (see page 31).

[Mar+10] Eitan Marder-Eppstein, Eric Berger, Tully Foote, Brian Gerkey, and Kurt Konolige. **The Office Marathon: Robust Navigation in an Indoor Office Environment**. In: *International Conference on Robotics and Automation*. 2010 (see page 21).

[MMA21] Shervin Minaee, Mehdi Minaei, and Amirali Abdolrashidi. **Deep-emotion: Facial expression recognition using attentional convolutional network**. *Sensors* 21:9 (2021), 3046 (see page 33).

[PA20] Nuno Pereira and Luís A Alexandre. **MaskedFusion: Mask-based 6d object pose estimation**. In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2020, 71–78 (see page 15).

[Pas+09] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. **Learning and generalization of motor skills by learning from demonstration**. In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, 763–768 (see page 35).

[Pas+17] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. **Grasp pose detection in point clouds**. *The International Journal of Robotics Research* 36:13-14 (2017), 1455–1473 (see page 19).

[POP98] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. **A general framework for object detection**. In: *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE. 1998, 555–562 (see page 32).

[Qur+19] Ahmed H Qureshi, Anthony Simeonov, Mayur J Bency, and Michael C Yip. **Motion planning networks**. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, 2118–2124 (see page 35).

[RBK21] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. **Vision transformers for dense prediction**. *arXiv preprint arXiv:2103.13413* (2021) (see page 33).

[Red+16] 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*. 2016, 779–788 (see pages 15, 32).

[Sal+12] Albert Salah, Javier Ruiz-del-Solar, Çetin Meriçli, and Pierre-Yves Oudeyer. **Human Behavior Understanding for Robotics**. In: Oct. 2012, 1–16. ISBN: 978-3-642-34013-0. DOI: 10.1007/978-3-642-34014-7_1 (see page 34).

[Sea80] John R. Searle. **Minds, brains, and programs**. *Behavioral and Brain Sciences* 3:3 (1980), 417–424. DOI: 10.1017/S0140525X00005756 (see page 27).

[SMD12] Yale Song, Louis-Philippe Morency, and Randall Davis. **Multimodal human behavior analysis: learning correlation and interaction across modalities**. In: *Proceedings of the 14th ACM international conference on Multimodal interaction*. 2012, 27–30 (see page 33).

[SOG19] Suwon Shon, Tae-Hyun Oh, and James Glass. **Noise-tolerant audio-visual online person verification using an attention-based neural network fusion**. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, 3995–3999 (see page 34).

[Sta+16] Sebastian Starke, Norman Hendrich, Sven Magg, and Jianwei Zhang. **An efficient hybridization of genetic algorithms and particle swarm optimization for inverse kinematics**. In: *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2016, 1782–1789 (see page 20).

[STE13] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. **Deep neural networks for object detection** (2013) (see page 32).

[Tur14] Matthew Turk. **Multimodal interaction: A review**. *Pattern Recognition Letters* 36 (Jan. 2014), 189–195. DOI: 10.1016/j.patrec.2013.07.003 (see page 34).

[ULQ17] Muhammad Usman, Siddique Latif, and Junaid Qadir. **Using deep autoencoders for facial expression recognition**. In: *2017 13th International Conference on Emerging Technologies (ICET)*. IEEE. 2017, 1–6 (see page 33).

[Vog01] Paul Vogt. **The Physical Symbol Grounding Problem**. *Cognitive Systems Research* 3 (Dec. 2001), 429–457. DOI: 10.1016/S1389-0417(02)00051-7 (see page 27).

[WKM93] Warren E Watson, Kamalesh Kumar, and Larry K Michaelsen. **Cultural diversity's impact on interaction process and performance: Comparing homogeneous and diverse task groups**. *Academy of management journal* 36:3 (1993), 590–602 (see page 28).

[WP20] Zhang Wen and Yuxin Peng. **Multi-level knowledge injecting for visual commonsense reasoning**. *IEEE Transactions on Circuits and Systems for Video Technology* 31:3 (2020), 1042–1054 (see page 34).

[Yi+20a] Jae-Bong Yi, Taewoong Kang, Dongwoon Song, and Seung-Joon Yi. **Unified Software Platform for Intelligent Home Service Robots**. *Applied Sciences* 10:17 (2020), 5874 (see page 18).

[Yi+20b] Jae-Bong Yi, Taewoong Kang, Dongwoon Song, and Seung-Joon Yi. **Unified Software Platform for Intelligent Home Service Robots**. *Applied Sciences* 10:17 (2020). ISSN: 2076-3417. DOI: 10.3390/app10175874. URL: https://www.mdpi.com/2076-3417/10/17/5874 (see page 53).

[Yu+19] Weijiang Yu, Jingwen Zhou, Weihao Yu, Xiaodan Liang, and Nong Xiao. **Heterogeneous graph learning for visual commonsense reasoning**. *Advances in Neural Information Processing Systems* 32 (2019), 2769–2779 (see page 34).

[YYL12] Gang Yu, Junsong Yuan, and Zicheng Liu. **Propagative hough voting for human activity recognition**. In: *European Conference on Computer Vision*. Springer. 2012, 693–706 (see page 31).

[Zel+19] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. **From recognition to cognition: Visual commonsense reasoning**. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, 6720–6731 (see page 34).

[Zhu+16] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. **Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks**. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016 (see page 32).

A Team Description Paper

RoboBreizh 2021 Team Description Paper

A. DIZET [1] C. LE BONO [1] A. LEGELEUX [1] M. NEAU [1]
N. WONDIMU [1] S. RASENDRASOA [2] Y. OMAR [4] M. BOUABDELLI[2]
A. PAUCHET [2] D. DUHAUT [1] C. BUCHE [3]

March 5, 2021

Abstract. Our team, RoboBreizh, was founded in 2018. In 2020, we have won the Best Performance award at RoboCup@Home EDU. Currently, we have 11 members from four different laboratories based in France and Australia. This paper aims to introduce the activities that are performed by our team and the technologies that we use. Main contributions include efficient detection, new NLP pipeline and gestures learning by demonstration. Our team is able to work using real robot, qiBullet or Gazebo simulator. RoboBreizh develops his own Pepper Gazebo full environment.

1 Introduction

RoboBreizh is initialy a RoboCup French team of the Brest National Engineering School (ENIB). The team was founded in 2018. Since then, RoboBreizh has won the Best Performance award at RoboCup@Home EDU competitions in 2020. RoboBreizh became a joint French team between the ENIB and the National Institute of Applied Sciences (INSA) Rouen Normandy.

The 2021 team consists of the following persons :
Students: Maëlic NEAU, Antoine DIZET, Amélie LEGELEUX, Cédric LE BONO, Natnael WONDIMU, Sandratra RASENDRASOA
Post-doctorate & Engineer: Yasser OMAR, Maël BOUABDELLI
Leaders: Cedric BUCHE, Alexandre PAUCHET, Dominique DUHAUT
Website: https://www.enib.fr/∼robobreizh

This paper is divided as follow. Section 2 presents RoboBreizh's main research innovations. Next, section 3 describes the architecture and the platforms proposed. Section 4 focuses on perception, navigation, movement and human-robot interaction (NLP and gestures). Finally, section 5 concludes this article.

[1]Lab-STICC, France
[2]LITIS, France
[3]IRL CROSSING, CNRS, Australia
[4]CCIT, AATMT Cairo branch, Egypt

2 Team research focus

The team offers original and efficient solutions in various contexts. Notably, our robot perception is handled using state-of-art algorithms to detect people and objects. Pepper moves its arms to support natural interaction and picks up objects using a Learning by Demonstrations model. The NLP part offers an efficient pipeline combining various complex modules. The team works both on real environment and virtual environment (qiBullet/Gazebo).

3 Architecture and environment

3.1 System Architecture

Our architecture is built upon 5 ROS modules: Manager, Perception, Navigation, Movement and Interaction. The manager is a high level structure. Modules output are stored as object instances and available as input information, later. The manager executes tasks in the required order, scheduling Pepper behavior. It also handles task priority. This architecture was designed to easily include new robot behaviors, without editing any external modules. In order to bypass execution lags, orders are cancelled if they take too long to execute.

3.2 Platform

Our system has been tested using real and simulated environments. Concerning simulation two different simulators are used. First, qiBullet [1] is a new simulation environment provided by SoftBank Robotics for Pepper and Nao robot. The advantage of qiBullet is that it can emulate the Pepper system NAOqi. Moreover, this simulator provides a ROS wrapper that makes it possible to run our code. Then, we also use the simulator of ROS, Gazebo [2] (version 7) to emulate the Pepper robot (meshes and sensors) but not the NAOqi system.

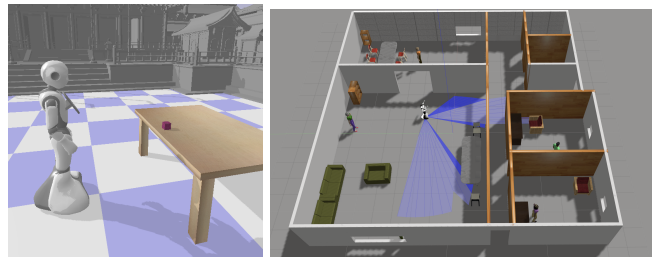

Fig. 1. 3D simulator: (left) QiBullet [1] and (right) Gazebo [2]

4 Approaches

4.1 Perception

YOLO [3] has become a standard in computer vision, especially for object and person detection. Other solutions are available, such as Mask-RCNN [4] that provides a mask generation of objects detected in addition to the bounding boxes. Beyond object and people detection, performing pose estimation was deemed necessary to detect people's movement (e.g. waving hands). An efficient tool in this domain is OpenPose [5], a real time multi-person system which can detect up to 135 different kinds of body keypoints. Mask R-CNN was chosen over other state-of-the-art object detection algorithms due to its ability to detect objects with pixel-level precision. Compared to other solutions, Mask R-CNN minimizes the noise added by the background and reduces the risk of inaccurate localisation of object. This level of precision is required when measuring the distance between Pepper and an object. Our module is also capable of understanding the current state of objects and person. For instance, by using the positions of chairs and persons in an image and how they overlap, it is possible to determine whether the chairs are available. Additionally, OpenPose is exploited to extract the positions of all the hands in an image and thus whether someone is waving. Also, gender and age estimation is performed using models proposed by [6].

4.2 Navigation

Our approach for navigation is based on a Pepper specific implementation of the ROS Navigation Stack [7]. First, mapping is throughout Gmapping ROS package which provides a tool to generate 2D occupancy map from laser sensors data using SLAM. Unfortunately, data provided by Pepper laser sensors is not sufficiently accurate to build a detailed map. Consequently, in addition to those inputs, we decided to feed the mapping node with data from the Pepper RGB-D camera. Once a valid map is obtained, it can be used for navigation. Thus, a real-time localization is performed using amcl ROS node [8]. The next step computes a path through a given goal and achieve it. This is performed by move_base Node [9] that defines global and local planner for the robot to follow.

4.3 Movement

Robots move their bodies to interact with theirs environments and with humans. Learning by demonstrations is one of the easiest way to teach a movement to a robot. Kinesthetic demonstrations (a human moves the robot arms) are exploited. With multiple demonstrations, the robot can generalize the movement. The learning is done at the trajectory level. We use Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR) to learn a movement with multiple demonstrations [10]. The initialization of the means is done with K-means algorithm and the selection of the number of Gaussians with the BIC score. The movement module is developed under some constraints. Pepper can

only take light objects because of its hands. It also has limited movements due to the robot reachable workspace. This module is developed to simplify movement learning. The learning movement module is composed of two parts: the learning phase and the movement phase. In the learning phase, the user can make multiple demonstrations to the robot for a single movement. Then the learning algorithm uses previous demonstrations. Figure 2 shows the generalization of the movement "Point a seat" of the joint 9 (corresponding to the right shoulder roll) with two demonstrations. The movement phase generates the movement with the result of the GMR. The movement module can replicate the learned movement in real time as the learning phase was previously performed. The movement cannot be adapted to any environment because the model used is a simple GMM/GMR. The modified GMM/GMR [11] can overcome this constraint and add an obstacle avoidance skill.

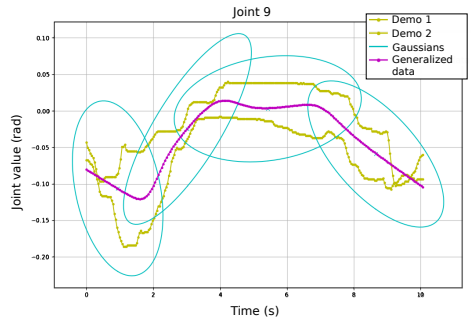

Fig. 2. Learned movement "Point a seat" of the joint 9 with the GMM/GMR. The two demonstrations are in green, the learned movement is in pink. The four Gaussians are displayed in blue.

4.4 Human-Robot Interaction (HRI)

Speech Recognition and NLP

RoboBreizh proposes a new pipeline designed upon Speech Recognition and Natural Language Processing (NLP). This pipeline is connected to Naoqi via a system detecting the user's input voice by analysing the evolution of ambient sound level. In our proposal, the Speech Recognition is handled by the Google API (Google Cloud Speech-To-Text) and NLP by using learning approaches such as Mbot [12] combined with rule-based APIs (e.g. Spacy [13]). In addition to those solutions, a Dialog Act classifier ([14]) is exploited. This classifier enables to adapt the system response to the type of dialog. For example, if the utterance is considered as an Action-Command the system runs Mbot as an intent analysis, otherwise the classic rule-based model is used. Using this classifier

as a pre-processing unit saves time and prevent intent classifier mistakes. We also implement a Sentiment Analysis Module [15] to classify user's utterance between "negative", "neutral" and "positive" sentiment. This enables to adapt Pepper's response using a rule-based system. An additional component enables to better understand the user intent through the use of commonsense knowledge base and deep-learning based pipeline [16]. Information inside a knowledge base can be represented in a tuple format e1, r, e2 where e1 and e2 are two entities in a relation r. An example of tuple related to the sentence "Going outside" would be: (E1 = "Go outside", R = Causes, E2 = "feel better"). In this example, the objective would be to generate E2 given E1 and R. First, we have selected specific relationships from the ConceptNet database. Then, through using the COMET model, we are able to generate common sense facts given the user's command.

Speaker recognition

We adapted classic techniques that works directly on the raw signal data without the need of handcrafted features [17]. The proposed model exploits SincNet, which requires as learning parameters only lower and higher cut frequencies, and therefore reduces the number of parameters learned per each filter and makes this number of parameters independent of the range of each filter. In addition, we combined both the SincNet and a Siamese Network with an algorithm to train Siamese neural networks in speaker identification. The algorithm is funded on the selection of the best anchor for each class. In addition, preparing negative pairs is done based on pairs that are nearest to the anchor class in features space. This selection enhances the performance of the Siamese network since it ensures to learn the confusing cases.

5 Conclusion

This paper describes the RoboBreizh team approach. A ROS architecture was developed to handle the competition tasks using a Pepper robot. Notably, our robot can move to a destination, point to an empty seat, take a bag in hand, compute a person's position, talk with someone and detect a waving hand. Our proposed architecture is also flexible and can be easily implemented on other robots. In the future, the team will develop additional features to increase Pepper usability in user-friendly environments.

Acknowledgments

This article benefited from the support of the project Prog4Yu ANR-18-CE10-0008 of the French National Research Agency (ANR), the French National Centre for Scientific Research (CNRS) and the INCA project (Natural Interactions with Artificial Companions) of the Normandy region. We also thank the City of Brest (BMO), CERVVAL company, Australian-French Association for Research and Innovation (AFRAN), Brittany Region and the National Engineering School of Brest (ENIB) for supporting this project.

References

1. Maxime Busy and Maxime Caniot. qibullet, a bullet-based simulator for the pepper and nao robots. *arXiv preprint arXiv:1909.00779*, 2019.
2. N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154 vol.3, 2004.
3. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
4. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.
5. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018.
6. Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 34–42, 2015.
7. Vittorio Perera, Tiago Pereira, Jonathan Connell, and Manuela M. Veloso. Setting up pepper for autonomous navigation and personalized interaction with users. *CoRR*, abs/1704.04797, 2017.
8. Brian P. Gerkey. amcl - ros wiki. http://wiki.ros.org/amcl, 2015.
9. E. Marder Eppstein. move_base - ros wiki. http://wiki.ros.org/move_base, 2016.
10. S. Calinon. *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.
11. Maria Kyrarini, Muhammad Abdul Haseeb, Danijela Ristić-Durrant, and Axel Gräser. Robot learning of industrial assembly task via human demonstrations. *Autonomous Robots*, 43(1):239–257, 2019.
12. Erick Romero Kramer, Argentina Ortega Sáinz, Alex Mitrevski, and Paul G Plöger. Tell your robot what to do: evaluation of natural language models for robot command processing. In *Robot World Cup*, pages 255–267. Springer, 2019.
13. Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.
14. Sujith Ravi and Zornitsa Kozareva. Self-governing neural networks for on-device short text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 887–893, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.
15. C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01 2015.
16. Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
17. Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *Spoken Language Technology Workshop*, pages 1021–1028. IEEE, 2018.

B GPD parameters file

```
# Path to config file for robot hand geometry
hand_geometry_filename = 0

# Path to config file for volume and image geometry
image_geometry_filename = 0

# ==== Robot Hand Geometry ====
#   finger_width: the width of the finger
#   outer_diameter: the diameter of the robot hand
#   (= maximum aperture + 2 * finger width)
#   hand_depth: the finger length
#   (measured from hand base to finger tip)
#   hand_height: the height of the hand
#   init_bite: the minimum amount of the object to be covered
#   by the hand
finger_width = 0.022
hand_outer_diameter = 0.280
hand_depth = 0.070                                  # 0.066
hand_height = 0.078
init_bite = 0.05

# ==== Grasp Descriptor ====
#   volume_width: the width of the cube inside the robot hand
#   volume_depth: the depth of the cube inside the robot hand
#   volume_height: the height of the cube inside the robot hand
#   image_size: the size of the image (width and height; image
#   is square)
#   image_num_channels: the number of image channels
volume_width = 0.135
volume_depth = 0.066
```

```
volume_height = 0.022
image_size = 60
image_num_channels = 15

#   (OpenVINO) Path to directory that contains neural network
#   parameters
weights_file = /workspace/src/dependencies/gpd/models/lenet/15channels/params/

###   Preprocessing of point cloud
#   voxelize: if the cloud gets voxelized/downsampled
#   remove_outliers: if statistical outliers are removed from the cloud
#   (used to remove noise)
#   workspace: the workspace of the robot (dimensions of a cube centered
#   at origin of point cloud)
#   camera_position: the position of the camera from which the cloud
#   was taken
#   sample_above_plane: only draws samples which do not belong to the
#   table plane
voxelize = 1
remove_outliers = 1
workspace = -3.0 3.0 -3.0 3.0 -3.0 3.0
camera_position = 0 0 0
sample_above_plane = 1

### Grasp candidate generation
#   num_samples: the number of samples to be drawn from the point cloud
#   num_threads: the number of CPU threads to be used
#   nn_radius: the radius for the neighborhood search
#   num_orientations: the number of robot hand orientations to evaluate
#   rotation_axes: the axes about which the point neighborhood
#   gets rotated
num_samples = 300                        # 500
num_threads = 6                         # 4
nn_radius = 0.01
num_orientations = 8                  # 8
num_finger_placements = 10         # 10
```

```
hand_axes = 2
deepen_hand = 1

### Filtering of candidates
#   min_aperture: the minimum gripper width
#   max_aperture: the maximum gripper width
#   workspace_grasps: dimensions of a cube
#   centered at origin of point cloud; should be smaller
#   than <workspace>
min_aperture = 0.0
max_aperture = 0.236
workspace_grasps = -2.0 2.0 -2.0 2.0 -2.0 2.0 # -0.5 0.5 -0.5 0.5 -0.5

#   Filtering of candidates based on their approach direction
#   filter_approach_direction: turn filtering on/off
#   direction: the direction to compare against
#   angle_thresh: angle in radians above which grasps are filtered
filter_approach_direction = 0
direction = 0 1 1
thresh_rad = 1.0

###   Clustering of grasps
#   min_inliers: minimum number of inliers per cluster;
#   set to 0 to turn off clustering
min_inliers = 1                          # 1

# Grasp selection
#   num_selected: number of selected grasps (sorted by score)
num_selected = 10                          # 50
```

C Toyota HSR description

Figure C.1: Sensors and actuators of the Toyota HSR robot [Yi+20b].