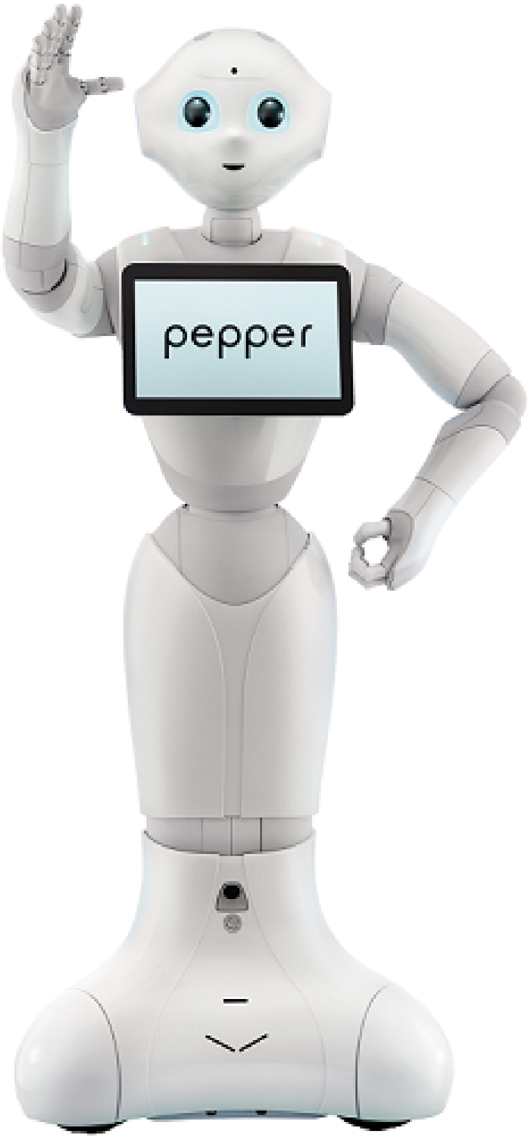
Eliciting user engagement with a social robot

Drawing attention techniques for Pepper

Edoardo Amadei

Delft University of Technology



TU Delft Delft University of Technology

Challenge the future

Eliciting user engagement with a social robot

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering - BioMechanical Design at Delft University of Technology

Edoardo Amadei

December 12, 2018

Faculty of Mechanical, Maritime and Materials Engineering (3mE)
Delft University of Technology

Eliciting user engagement with a social robot

Abstract

The elicitation of user engagement is one of the current challenges of human-robot interaction, alongside with the identification of appropriate metrics to evaluate users' experience. Several studies focused on strategies aimed at maintaining engagement throughout an interaction, but limited research has been done on how to initiate it by drawing users' attention. The use of social cues in nonverbal human-human communication has been identified as a reliable source of information to determine if a person is engaged. These social cues can be used not only to understand more about human behavior, but also to design robot behaviors that can successfully draw the attention of humans. In this project we look to investigate what are effective techniques to draw attention and elicit initial engagement with a social robot at the entrance of a building. The robot proposed to display these behaviors is the humanoid robot *Pepper*, from SoftBank Robotics, as it has been specifically designed for human-robot interaction. Initially, the on-board functionalities of the robot are going to be tested. Secondarily, state-of-the-art techniques are going to extend those functionalities to improve *Pepper*'s interactive skills. Eventually, robot behaviors are going to be designed and displayed to participants during an experiment. We aim at understanding the reactions of people to identify the most effective drawing attention behaviors and examine if the encounter with the robot is affected by a novelty effect. Different metrics are proposed to measure both these phenomena in our results. Our system is developed in Python with features extracted from *Pepper*'s Naoqi framework.

Thesis Committee:

Chair:	Prof. David Abbink, Faculty 3mE, TU Delft
University supervisor:	Prof. Koen Hindriks, Faculty EWI, TU Delft
Committee Member:	Dr. Dick Plettenburg, Faculty 3mE, TU Delft

Preface

I chose robotics as the finishing touch for the specialization of my career, because I am convinced that robots will play a fundamental role in human society in the nearby future. During my studies, I realized how this conviction of mine is no more than a myth for most of the supposed users of these technologies, so I found my focus of interest in bridging between these two realities that still strive to encounter. Starting this project was, therefore, a natural decision to take; in particular considering the opportunity it offered to learn about *human-robot interaction* and *user engagement*. Working with *Pepper* was definitely one of the most motivating characteristics of this project, because it taught me how to work on different fields of robotics to improve the encounter between humans and robots. Identifying effective attention drawing techniques for a humanoid social robot felt the appropriate challenge to deepen my knowledge and test my engineering skills.

Acknowledgements

I would like to express my gratitude for my supervisor Dr.ir. K.V. Hindriks, for his guidance and assistance during this year of work together. At many stages in the course of this project I benefited from his advice, particularly so when exploring new ideas. His careful editing contributed enormously to the production of this thesis.

I would also like to show gratitude to Elie Saad, for his always friendly support and advice in the fulfillment of this project. His positive outlook and confidence in my research motivated me and encouraged me. This project would have not been possible without his contribution.

It is also important to recognize the support of Ruud de Jong, who provided me with the facilities and technical assistance through various stages of the implementation of this project.

I also want to thank my family in Italy, my Mom, Dad and Brother, for their support and patience through this project. Thanks also to my Aunt, Uncle and Cousins, who often reached me here in Delft with their warmth and encouragements. A special thank you to my Grandmother, for her unwavering confidence in me and for motivating me so strenuously with her optimism.

Finally, a huge thank you to my family of friends all around the world: from Modena to Torino, from Castelfranco to Shanghai, from Delft to Pensacola. There are not enough words to express my gratitude for all of you, I feel so blessed for how rich you make me feel. I will never forget your support during hard times, neither your enthusiasm to celebrate the good ones. This achievement is a tribute to your friendship and love.

Edoardo Amadei
Delft, University of Technology
December 12, 2018

Contents

List of Figures

Chapter 1

Introduction

In the last decades, the development of robotics has remarkably broadened the application fields of artificial agents. From the initial employment in factories for industrial production, robots started to be used in domains that imply closer contact with users, such as search and rescue, entertainment and hospital care. These new contexts motivated the implementation of robots with socially interactive skills, in order to promote their social acceptance and make human-robot interactions as natural as possible. In fact, the major requirements for social robots are set by human social psychology, not only in terms of appearance and movements that robots should have, but also as capability to express a certain degree of ”intelligence” through their behaviors [3] [28]. However, it is still hard to understand what social skills a robot must have according to human expectations, because they vary depending on people and contexts of application.

Approaching a perfect stranger is an eventuality that easily happens in everyday life, whenever we need to ask for information, directions, or simply to order food. We have naturally developed several strategies to draw the attention and make contact with our designated interlocutor, doing so according to the context we are set within and the social standards related to it. The communicative skills we use are progressively refined through multiple experiences of interactions in the society we live in, which makes it complex to explain objectively why we behave in certain ways under specific circumstances. However, it may happen that our verbal and nonverbal behaviors are not effective in contexts that developed from different societies, so we might perceive these cultural differences as weird and uncomfortable. The encounter with a diverse culture can be improved if the new standards are understood and integrated into our behaviors. Human-robot interactions can be interpreted as one of these encounters, where humans and machines need to tune their communicative skills in an effective way to ensure mutual understanding. In this case, we want technology to adjust to our expectations, so we have to provide machines with enough knowledge about our social rules to behave in an appropriate way among humans.

Encountering a robot in a train station, hospital or hotel is not a plot for science fiction anymore, but a reality that is getting closer every year. Robots built with the specific aim of interacting with humans are growing in popularity and sophistication, many of them are already available to the public. If human-robot interactions are becoming part of everyday life, we need to understand how we want the encounter with technology to be for a common user. There are several aspects that affect the quality of an interaction and multiple expectations to meet according to its context, but in this project the focus is going to be the establishment of initial engagement. Therefore, we are going to analyze what are the strategies that a robot could adopt in a daily life scenario to draw the attention of humans in a way that feels natural and safe. Users should feel at ease with a social robot, with the way it makes contact with them and behaves using shared social cues. In general, the robot should draw the attention leaving a positive first impression to the user, being discrete and avoiding behaviors that could feel inappropriate or awkward.

1.1 Problem description and motivation

The goal of this project is to analyze what are effective techniques to elicit initial engagement with the humanoid robot *Pepper*. The robot has been specifically designed to establish human-robot interactions and already displays social features that can be further implemented for this purpose. The first contact that *Pepper* will make with a human could occur in various ways and contexts. For simplicity, only specific circumstances are going to be studied in this research. In our ideal scenario, the robot will be stationary and try to draw the attention of people who are passing in front of it. To do so, it will adopt different behaviors (e.g. face tracking, short greeting etc.) and display social cues. The encounter is expected to last less than a minute, because *Pepper* will not interact with participants but only aim at eliciting initial engagement.

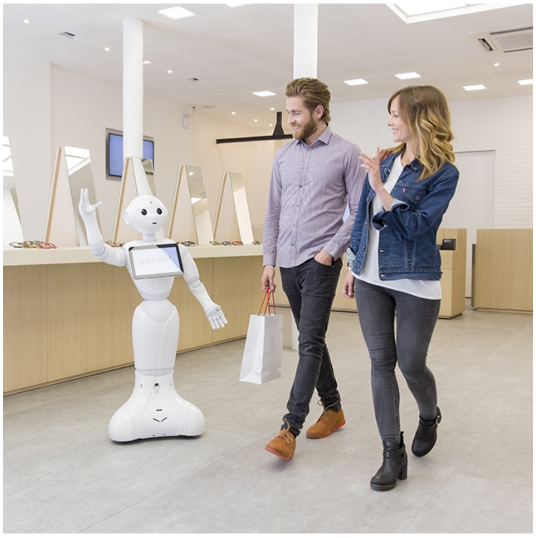


Figure 1.1: An example of how an interaction will occur in our ideal scenario [6].

The environment in which *Pepper* is going to be deployed has to be chosen according to the expectations that users associate to it. In a setting like a train station where people are often hurrying and there is a lot of background noise, entertaining a conversation with a robot might not be what a person is looking for. There are also scenarios that may have an ambivalent interpretation like a public building, where people usually start interactions when entering rather than on the way out. People have expectations from the contexts they are set within, so those contexts that maximize the willingness of interaction should be selected to make users perceive *Pepper* as socially acceptable. An appropriate location that satisfies these requirements is a place that people are entering where they may expect to greet or have a short conversation with someone. This could be the entrance hall of a building, a break area with coffee machines or a cafeteria.

The main motivation of this project is the growing popularity of social robots among the public, whose number on the market is expected to considerably increase in the upcoming years [43]. If more users can be exposed to this new technology, it is important that the first encounter with it is smooth and natural, even for people who never came in contact with robots or know about them. Initial engagement is one of the most characterizing features of user experience [56], and can contribute to the social acceptance of robots among people. However, the evaluation of human-robot encounters is a complex matter that involves several aspects often difficult to assess individually and in combination with each other. They need to be defined a priori because the elicitation of initial engagement is very susceptible to changes in these variables, and may affect the validity of the result. Therefore, it is fundamental to specify that the whole application with *Pepper* is going to be indoor and stationary, and that navigation will not be taken into account in this study. Under these circumstances, the robot will have to draw users' attention using social cues and combining them in different nonverbal behaviors. These are expected to make the robot behave in a human-like manner, and users' reactions to them are going to be studied to fine tune them toward social acceptability.

1.2 Research question

The objective of this project poses the motivating research question of identifying effective techniques for initial engagement with the humanoid robot *Pepper*. Taking into consideration the several aspects that will contribute to shape the quality of the interaction, this main question can be addressed focusing on the following subquestions:

1. Which social cues should be implemented on *Pepper* to draw users' attention and ensure successful initial engagement?

2. Is *Pepper* affected by a novelty effect that induces users to approach regardless of the behaviors it displays?

3. How can initial engagement be evaluated effectively in these scenarios and which metrics should be used for this purpose?

1.3 Approach

In order to better conduct this research the human factor has to be taken into consideration using a user-centered approach. This has been defined as ”a set of methods to gain powerful insights into the ’actual’ practices, habits, needs and values of the users you are designing technology for, rather than purely having to rely on your own perceptions, assumptions and preconceptions” [39]. As *Pepper* will be drawing people’s attention, it is reasonable to investigate what is users’ general feeling about its behavior. Users’ feedback will help understand if there are aspects of the robotic platform to improve and where to orient the research focus. I will therefore conduct a preliminary study to investigate this before developing new features on the robot.

A baseline interactive application will be prepared and tested in advance, to immediately deploy the social skills of *Pepper* with users and analyze their performance. The basic features from the NAOqi framework will be incorporated in the application, so the robot will be using its embedded functionalities to detect, track and talk to people. The application is going to be implemented on *Choregraphe*, a graphical programming tool that allows to control Softbank robots, and will simulate a simple conversation with an unknown interlocutor. This is going to be tested in a pilot experiment described in Chapter 4. The analysis will determine whether on board functionalities are sufficient for social interaction with *Pepper*. Eventually, software and hardware limitations will be taken into account to understand the extent up to which they can be developed.

Another crucial issue to be addressed is the definition of appropriate metrics to evaluate initial engagement. Thanks to several examples retrieved from the literature they will be selected from the following fields:

- Behavioral measures: analyze participants’ nonverbal behaviors during the experiment providing a reliable indication of their level of engagement. These include user proximity, head orientation and gaze direction, rhythm of interaction and response times;

- Self-reported measures: rely on participants’ report of the experience through post-experimental questionnaires and interviews. These are the most common metrics even if hard to validate and corroborate;

- Task performance measures: analyze the experience from a more practical and utilitarian point of view, but contribute to complete the assessment of user experience. Among the main ones, the percentage of successful initial engagement is particularly relevant, to have an estimate of how effectively *Pepper* can draw attention.

The focus of this research is the initiation of engagement, not its maintenance, so *Pepper*’s task performance is going to be measured only on the first encounter with users. Therefore, behaviors like approaching the robot, looking at it or touching its tablet are going to be analyzed in relation to the social cues that the robot displays.

To answer the main research question, an experiment with participants is going to be conducted. A pilot will both test the main behaviors that *Pepper* will display during the experiment and analyze whether a novelty effect influences users' behavior toward the robot. During this phase, it is important to observe if there will be significant changes in the number of people who approach the robot and spend time trying to interact with it. In this way it will be possible to reduce the influence of the novelty effect, which might induce users to react positively to the robot just because it is novel, regardless of how it behaves. The focus of the experiment is going to be on user's behavior: *Pepper* is going to display some behaviors to participants and their reactions will be recorded by the experimenter. The motivating assumption of this approach is that users experience is going to be less affected by the novelty effect, hence results will be more reliable in this phase of the experiment.

1.4 Outline

The outline of the report follows the structure of the project, with Chapter 2 describing the related work in the field of initial engagement, drawing attention and novelty effect. Chapter 3 continues with a technical overview of the humanoid robot *Pepper* and describes the preliminary experiments conducted on it. Chapter 4 explains the baseline application developed to deploy the robot's on-board interactive skills and the preliminary study conducted to test their performance. Chapter 5 compares the different state-of-the-art algorithms considered to extend *Pepper*'s functionalities and presents advantages and limitations of those finally selected. The design choices of the experiment and the behaviors developed for *Pepper* are analyzed in Chapter 6, that also explains which data are going to be collected during the experiment. Chapter 7 describes the set up, the procedure and the objectives of the final experiment, while Chapter 8 provides a qualitative evaluation of the results and presents the conclusions of this project.

Chapter 2

Related work

There is limited work that investigates initial user engagement with social robots, because most of related HRI studies focus on its maintenance throughout an interaction rather than on its establishment. In the same way, research on nonverbal behaviors has often aimed at making robots display social cues to preserve users' attention instead of drawing it. Moreover, there is little knowledge about how human-robot interactions evolve in the long-term, so it is not clear how they can be affected by the novelty effect. Even if these fields still need further exploration, the existing body of related work can be used to better structure this project and orient its research. Articles with similar goals or methods are hereby presented to illustrate the main contributions to this work. They belong to the three main topics of interest: *initial user engagement*, *drawing user attention* and *novelty effect*.

2.1 Initial user engagement

One of the most widely accepted definitions of user engagement is provided by Sidner *et al.* [57], who describe it as ”the process by which individuals involved in an interaction start, maintain and end their perceived connection to one another”. This process involves both nonverbal and verbal behaviors as well as low-level and high-level cognitive processes. Similarly to Sidner *et al.*, also O'Brien and Toms [45] suggest that engagement can be divided into distinct stages: point of engagement, period of sustained engagement, disengagement and re-engagement. In particular, the point of engagement stage corresponds to the instant in which the engagement process begins, which is usually triggered in conjunction with the arousal of users' interests and attention. It can occur either to satisfy a specific goal or out of curiosity and desire to have new engaging experiences.

User engagement is often measured by observing the display of specific social cues throughout an interaction. In particular, proxemic behavior, gaze direction and gestures are going to be analyzed more in detail because indicative of the degree of engagement at its beginning [53]. In addition, these behaviors are the easiest to detect and interpret when user's intention to interact is manifested.

2.1.1 Proxemic behavior

Michalowski *et al.* [41] study the establishment of initial engagement within a context very similar to the one of this project. They investigate what are the best behaviors for a Roboceptionist (robotic receptionist) to attract people and initiate an interaction. During the research, they also focus on the interpretation of users' nonverbal behaviors to infer their willingness to interact. Being the interaction stationary, proxemic behavior is of particular interest because humans choose spontaneously their conversational distance from the robot, revealing their degree of comfort around it. The authors classify users using the proximity framework developed by E.T. Hall [25]. Together with social cues like user's direction, head visibility and pose, user proximity has been used to decide what behavior the robot should adopt. However, the study concludes that more parameters (i.e. users' speed of motion and direction) should be taken into consideration to achieve a better users classification and improve the interpretation of human intentions.

A similar analysis of proximity was performed by Bergström *et al.* [4], who addressed the problem of a robot that had to detect people and approach them naturally. This work goes beyond the methodology of Michalowski *et al.*, employing the mobile robot *Robovie-II* that also considered direction and speed of motion of people to assess their willingness to interact. Results show that an accurate classification of users was performed, especially when people showed hesitation and indecisiveness (i.e. they kept distant from the robot but looked toward it). The authors made use of off board floor sensors positioned around the robot to achieve such precise classification. However, the sensors sometimes discouraged people to approach the robot because they had to walk on them.

2.1.2 Gaze direction

Okuno *et al.* [46] developed a system to improve social interaction based on audio-visual tracking. In their experiments, they use a robot capable of gazing in the direction of a sound source (i.e. a person speaking) and tracking faces. They analyzed two different scenarios, in which the robot was either approached by a participant or had to initiate an interaction itself. In both cases, the new tracking features made the robot look more aware of the context and increased the engagement of users. The authors found out that participants felt engaged even when the behavior of the robot was rather passive. Hence, results show that following the interlocutor's face or turning toward him when he is speaking makes the interaction more natural. An attentive behavior makes the robot behave in a human-like manner even when waiting for an interaction, becoming advantageous to draw users' attention.

Similar conclusions were drawn by Sidner *et al.* [57], who analyzed the effects of visual cues on user engagement during human-robot interactions. Their study focused on a collaboration task between participants and a robot, which could display nonverbal behaviors like blinking, nodding, gaze aversion and face tracking. To initiate the interaction, the robot searched for a face and estimated user's willingness to interact through proximity and user's motion. Then offered greetings to ensure engagement and started a conversation to maintain it. Timing of gazes was confirmed to be a crucial factor to make the robot feel more real, as participants assessed its behavior as ”appropriate” and ”on time”. Face tracking and gaze aversion represented the most relevant aspects of the interaction, while blinking and nodding resulted to be the least salient. In summary, gaze cues combined with face tracking are essential to maintain engagement but are also functional to its establishment, because they make the robot look like an aware and competent interaction partner.

2.1.3 Gestures

In their analysis on visual cues during human-robot interaction, Sidner *et al.* [57] also explored the effects of engagement gestures to initiate an interaction. In particular they compared two scenarios, in which the robot was displaying head and hand gestures in a collaboration task. The use of gestures encouraged users to approach the robot and respond to it, making them more involved in the interaction. On the contrary, when the robot was only talking, participants paid less attention to it and assessed its behavior as less ”appropriate”. Engagement gestures improved users’ perception of the robot and made it look more competent for the interaction.

Schwarz *et al.* [55] developed an algorithm that estimates user’s intention to interact taking into account body pose, motion and facial features. More specifically, they analyzed what behaviors people displayed to naturally engage with each other and to interact with a vision-based interface. It emerged that looking toward the interface and using gesture such as waving or raising the hands above the head were widely used. These techniques express participant’s intention of drawing the attention of the interlocutor and interact with it. It is important to notice that the same technique of ”wave to engage” was used by most of the subjects, and can be clearly connected to the attempt of establishing initial contact. Therefore, the understanding of such gestures by social robots and their ability to reproduce them can considerably improve the quality of the interaction and clarify user’s intentions.

2.2 Drawing user attention

In order to elicit initial engagement, *Pepper* needs to use effective techniques of attention drawing. Many studies have analyzed which strategies are more successful to manipulate user’s attention combining different social cues. Pitsch *et al.* [48] investigate how to secure and sustain user’s engagement in a real world scenario using the *Sony Aibo* robot as a museum guide at the Ohara Museum of Art. The purpose of this experiment is exploring effective methods to enter an interaction in a dynamic environment solely exploiting user’s face orientation. The robot employs a basic mechanism to dynamically break up the predesigned talk, to simulate a human conversation in which the addressee is distracted or not attending. Whenever a nearby user was detected, the robot lifted its head, flashed the eyes and started to talk to get user’s attention. Once the visitor was looking at the robot, it would restart the opening sentence and pronounce it completely. This ”pause and restart” procedure attempts to handle dynamic and contingent interactions in which engagement is built in a stepwise process of mutual adjustments, and therefore cannot be predicted in advance.

Hoque *et al.* [29] explore techniques to draw and control human attention in different social situations. Depending on the field of view of the targeted person, the behavior of the robot has to be adjusted to attract attention. Results showed that head turning and shaking is sufficient if the robot is in the near peripheral field of view, while more abrupt motions are necessary once in the far peripheral field of view. Speaking is also used if a human is turning back to the robot. Once attention is attracted, eye contact and blinking are used to strengthen the connection. At this stage, target's attention can be controlled by the robot by gazing to a particular point in space, making the human mimic the same behavior. This suggests that gestures and utterances have a major role in attention drawing, but that gaze awareness and other visual cues are fundamental to establish and manipulate joint attention.

Torta *et al.* [62] used the *NAO* robot to analyze how to attract attention with visual and auditive cues. The authors designed four different communication behaviors on the robotic platform: (1) attempting to establish eye contact, (2) blinking, (3) waving and (4) uttering the word ”Hello”. The first three are nonverbal actions of visual nature, while greeting users is purely auditive. However, due to the noise generated by the motors of the robot while doing gestures, waving has also been classified as partially auditive. Once drawn the attention, the robot would present information to participants in the form of short video-clips. Results showed that reaction times differed considerably among the four behaviors. In particular, auditive cues like speech and waving had the fastest response, while blinking and trying to establish eye contact were the least salient cues. This suggests that attention drawing techniques relying on cues such as waving and speech may be more successful. Nevertheless, gaze cues have a stronger impact when users are already focused on the robot and can help ensure that attention is maintained.

2.3 Novelty effect

The novelty effect represents the trend of users' performance to initially increase when a new technology is introduced, which is not related to any actual improvement but to the increased interest in the technology itself [59]. Because of its temporary nature, it is important to observe this phenomenon in long-term interactions and wait it wears off to obtain more reliable results. The findings of Gockley *et al.* [20] highlight the relevance of long-term observations in human-robot interaction to evaluate real user engagement. They researched how users interacted with a roboceptionist able to give directions, tell stories and looking up weather forecasts over a period of nine-months. Results showed that the average number of interactions decreased throughout the experiment as well as their average duration, which reduced to half the time of initial interactions. This is mainly due to the novelty effect, which was observed to last until the 5th week of the experiment. Afterwards, the number and duration of interactions reached a steady state value with slight variations. While it is hard to know a priori how long the novelty effect will last, it is possible to measure its influence observing significant differences in metrics from the beginning of the experiment.

Since its release to the public in 2015, also *Pepper* has been employed in studies on human-robot long-term interactions. Rivoire *et al.* [52] analyzed how to program *Pepper* to become an engaging life companion focusing on the right degree of proactivity to use in its behaviors. The aim was to avoid that the robot was perceived as annoying (too much proactivity) or boring (too little proactivity), to make it more appealing to the public as a domestic robot. In particular, they analyzed the influence of the novelty effect over 8 weeks of experiment measuring the time that users spent interacting with the robot. Their results showed that users who had the robot for long periods used it less than users who got it for a shorter time. The way they interacted with it did not change through the experiment, but lasted less than at the beginning, reaching a steady state duration after the first 2 to 3 weeks of experiment.

Leite *et al.* [34] highlight the correlation between novelty effect and familiarity of users with the robot. The more users get used to the robot, the more they prefer novel behaviors. The authors suggest that the novelty effect can be measured with behavioral measures such as the time users spend looking at the robot. In one of their previous studies [35], the authors investigated the role of social presence in long-term human-robot interactions, analyzing also the novelty effect. In particular, they found that after the second week users spent looking at the robot half the time they spent at the beginning. These results were aligned with those obtained from questionnaires about social presence of the robot, which also decreased throughout the experiment. Similarly with the study of Gockley *et al.* and Rivoire *et al.*, it seems that a significant variation in such behavioral metrics indicates well if the novelty effect is wearing off.

2.4 Metrics

As anticipated in Section 1.3, self-reported measure like questionnaires or interviews are very often used to evaluate engagement during an interaction. Also in this study, in spite of their subjectivity, post-experimental questionnaires are going to be used to have a better insight into users' experience. However, behavioral measures are more objective indicators of initial engagement and do not compromise the naturalness of the interaction [3]. The related work hereby presented provides a large selection of behavioral metrics, which have been classified in the table below.

Metrics	Description	Relevant for IE	References
Eye contact	A direct look between the robot and the participant	Y	[29] [62]
Gaze direction	The synchronized movement of eyes and neck that indicates the direction of visual attention (also referred to as head pose)	Y	[3] [9] [26] [31] [44] [58]
Direction of motion	The direction of motion of a user	Y	[4] [41] [53]
Speed of motion	The speed of motion of a user	Y	[4] [41] [53]
Proxemic behavior	The different ways in which humans position and orient their body in relation to the people and objects around them in space	Y	[5] [15] [64]
Waving	Moving one's hand to and fro in greeting or as a signal	Y	[11] [55]
Speech and gestures	Pronouncing utterances in synchronization with gestures	Y	[1] [29] [30]
Body pose	The way people position their body and arrange their limbs	N	[3] [13] [55]
Facial expressions	A combination of eyes, lips, nose and cheek movements that help form different moods of an individual	N	[26] [55]
Blinking	Shut and open the eyes quickly	N	[26] [57] [29] [33]
Nodding	Lower and raise one's head slightly and briefly	N	[26] [33]
Face tracking	Following one's face with gaze	Y	[46] [57]

Table 2.1: A summary of the metrics for user engagement encountered in literature. The metrics are classified as relevant for initial engagement or not, based on the references provided.

Chapter 3

Experimental platform

3.1 Humanoid robot Pepper

Pepper is a humanoid commercial robot created by Softbank Robotics, presented in June 2014 and available to the public from February 2015. It has been designed to establish interactions with humans, so it has the capability to read basic human emotions, interpret the tone of voice and facial expression. Its functionalities make it more a daily companion rather than a domestic helper, in fact, it has been the first robot to enter Japanese homes and be employed as a receptionist in several offices in the UK [63]. Even if most of its characteristics are pre-programmed and sum up to very simple behaviors, its software framework allows to easily develop more advanced features. In addition, thanks to its friendly and anthropomorphic design, *Pepper* is the ideal robot to work with to study user engagement in social interactions, especially in a scenario where users' attention should be drawn by how the robot looks and behaves.

3.2 Technical overview

The main body dimensions are shown in Figure 3.1. The weight of the robot, approximately 28 *kg*, is concentrated in the lower part to optimize movements of the arms and torso.

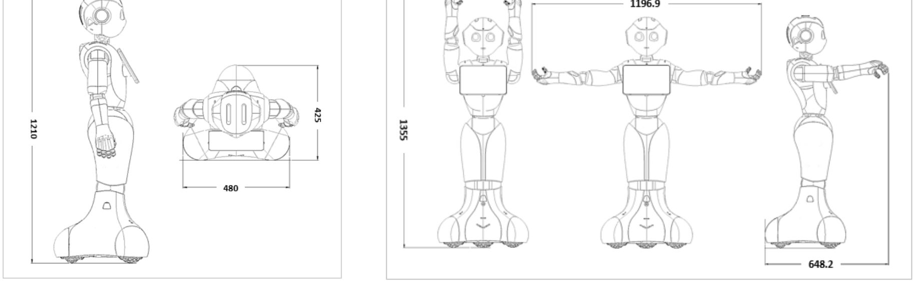


Figure 3.1: Physical dimensions of Pepper [22].

To activate its degrees of freedom the robot uses 14 motors located as shown in Figure 3.2. For the scope of this project, the 3 motors of the wheels are not going to be used, as the robot will be involved in a stationary interactive scenario. For the same reason, also its 6 laser sensors situated in the lower body, employed for front ground and surroundings evaluation during navigation, are not necessary. *Pepper* has two sonars that lie below the torso, and the frontal one is going to be fundamental for people detection in this task. They are located in the back and front of the *KneePitch* and used to estimate the proximity of objects (or interlocutors) within a range of $5m$. The field of detection of the sonars is shown in Figure 3.2.

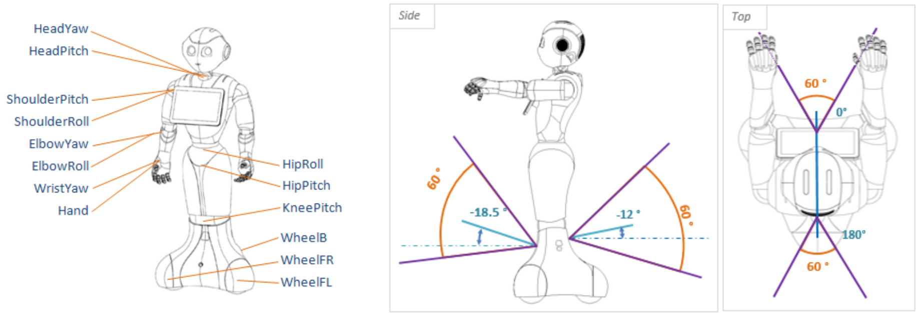


Figure 3.2: Joints location (left) and field of detection of sonars (center and right) [22].

Most of the sensory input needed to fulfill the task of this project is going to come form *Pepper*'s cameras. Two identical 2D cameras are located on the forehead and in the mouth, while behind the right eye of the robot it is placed a One ASUS Xtion 3D depth camera. The fields of view provided by these sensors are shown in Figure 3.3. A set of four microphones is positioned on top of the head, which optimizes understanding of human speech as interlocutors are usually taller than *Pepper*. Instead, loudspeakers are contained in the "ears" of the robot, on the sides of the head.

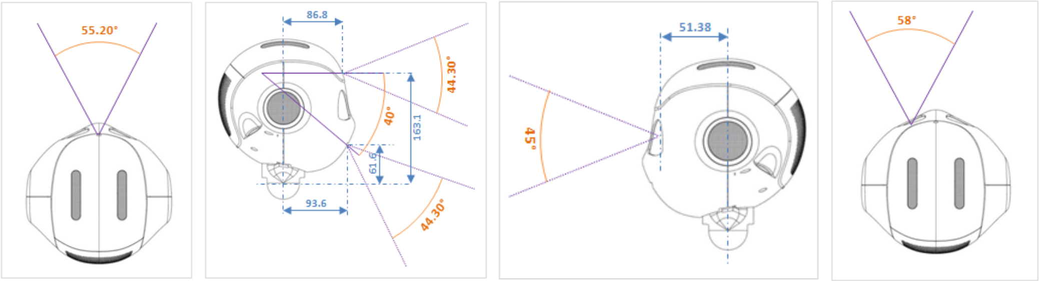


Figure 3.3: Field of view of 2D cameras (left) and 3D sensor (right) [22].

The robot can also interact through the tablet on its chest, which can be used for both visual representation (e.g. displaying images) and tactile feedback (i.e. touching the tablet). *Pepper* has additional tactile sensors on its body as shown in Figure 3.4. Three of them are on its head, in the front (A), middle (B), and rear (C), both hands have one in the back and three others are in the bumpers at the bottom.

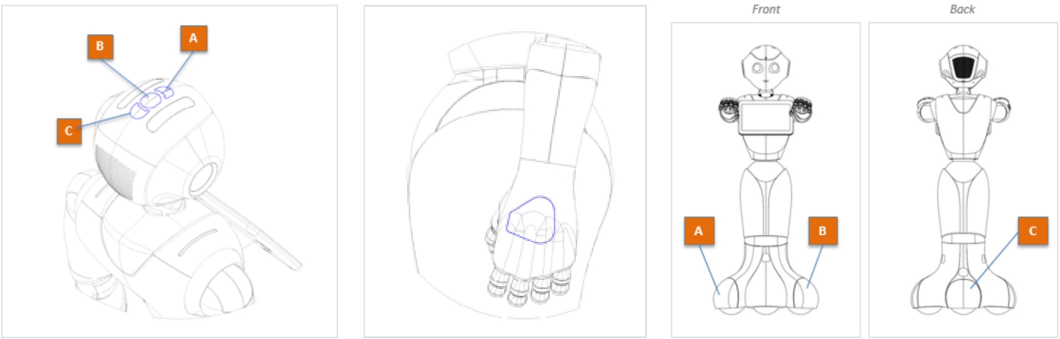


Figure 3.4: Tactile sensors in the head (left), hands (center) and bumpers (right) [22].

Pepper uses an embedded PC with an Intel Atom card. The card has 2 CPU cores running at 1.6 GHz, and 1 MB of RAM. There is no GPU. The Operating System is OpenNAO and on top of it, NaoqiOS, the robotics OS solution created by Aldebaran, is running all the Software used for hardware control and AI decision making.

3.3 NAOqi Framework

NAOqi is the name of the main software that runs on the robot and controls it. The NAOqi Framework is the programming framework used to program Aldebaran robots [23]. Within the NAOqi structure it is possible to create several applications with *Pepper* using the modules already embedded in it. These modules have default methods that are programmed for specific actions and are all comprised in the API (Application Programming Interface), so that users can easily retrieve what are the commands that the robot can execute. The main programming languages available in the framework are C++ and Python.

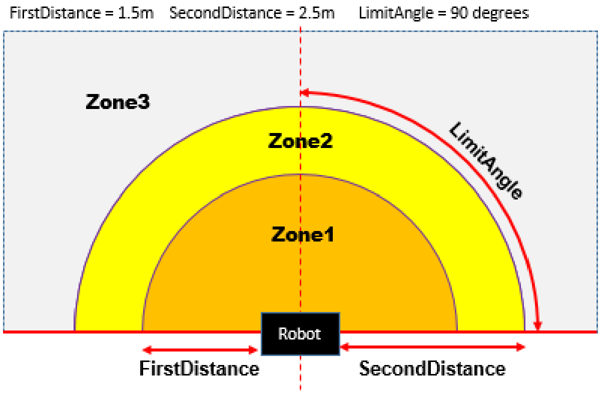

FirstDistance = 1.5m SecondDistance = 2.5m LimitAngle = 90 degrees
Zone3
Zone2
Zone1
LimitAngle
Robot
FirstDistance
SecondDistance

Figure 3.5: Default engagement zones [22].

There are several modules available within the NAOqi framework. For instance, *ALEngagementZones* gives access to the configuration of the engagement zones that *Pepper* uses to classify people according to their relative distance form it. Depending on the zone in which a person is, the robot can change its behavior, and this can be adjusted using the parameters *FirstDistance*, *SecondDistance* and *LimitAngle*, which are illustrated in Figure 3.5. Other modules relevant for the purpose of this project are further described in Appendix C.

3.4 Preliminary experiments

Before starting to work on *Pepper*, some preliminary tests have been conducted to investigate capabilities and limitations of its sensors. Even if this information is already partially available on the technical manual of the robot, it is necessary to validate it within a context similar to the one used for this project. The experiments took place in office 12.260 of EWI in TU Delft campus. The room has a bright artificial lighting and a broad window that provides further external illumination, creating good visibility conditions for the robot. Pepper has been placed in front of the window looking at the entrance door, 4*m* away from it. The applications used to run the experiments have been implemented in *Choregraphe*, a graphical programming tool to control Softbank robots and create basic behaviors for them.

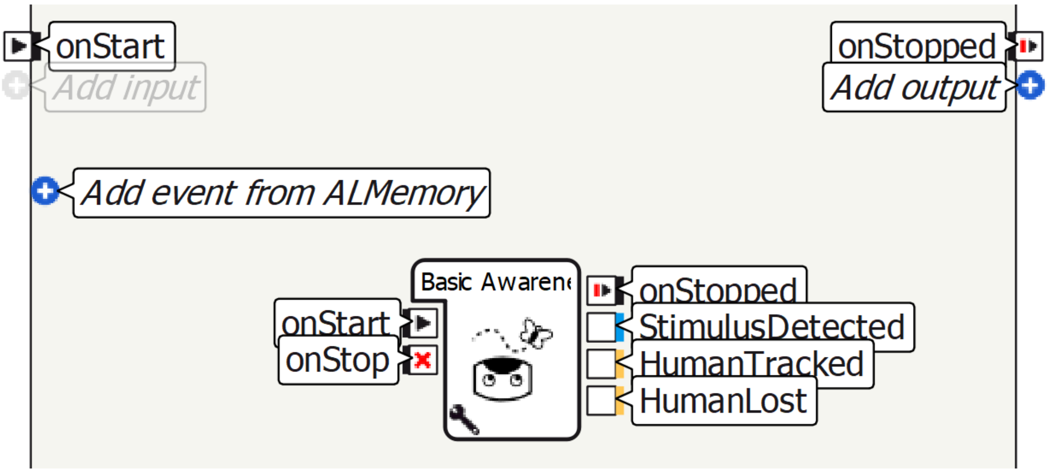


Figure 3.6: An example of the working environment in *Choregraphe* with description of inputs and outputs.

Figure 3.6 shows an example of the working environment in *Choregraphe*. On the left side there are the inputs used to start the behavior, which can be triggered either manually (i.e. *onStart*) or automatically when an event is raised (i.e. *Add event from ALMemory*), while on the right side there is the output that stops the behavior (i.e. *onStopped*). In the center of the working space there is an example of a *Choreographe* block, with inputs on the left and outputs on the right. In order to start a behavior, one input of the working environment has to be connected to one from the block. To stop the behavior one of the outputs from the block needs to be linked to the output of the working environment. There are different kinds of inputs and outputs, which can be distinguished by their color: black (sends or receives a generic signal when activated), blue (sends or receives a string), yellow (sends or receives a number). Multiple blocks can be used and connected to each other to generate more complex behaviors for *Pepper*.

3.4.1 People detection

The goal of this experiment is to understand the maximum distance at which *Pepper* can detect a user. A simple two blocks application has been implemented. Once started the behavior, an input is sent to a *Basic awareness* block, which makes the robot establish and keep eye contact with people. Its *Human Tracked* output, which is triggered when a stimulus that is confirmed to be a human is detected, is connected to a *Say* block that makes the robot say ”There you are!”. The results of this first experiment showed that *Pepper* can successfully detect people up to 3*m*. The robot may still detect users up to 4*m*, but only if it is already looking in the direction where they are going to appear. If the experimenter was right outside the door of the office (from 4*m* to 5*m*), no detection occurred even if the robot was already looking in that direction.

3.4.2 People tracking

The goal of this experiment is to understand the maximum distance at which *Pepper* can track a person once detected. A simple two blocks application has been implemented. Once started the behavior, an input is sent to a *Basic Awareness* block, and its output *Human Lost*, which is triggered when the human currently tracked is lost, is connected to a *Say* block that makes the robot say ”I lost you!”. The experimenter made sure to be tracked by the robot once eye contact had been established, then moved away waiting for the robot to speak (i.e. human tracked is lost). The robot was able to keep track of a person up to 3*m*, but sometimes had difficulties in coping with bright light and reflections, which could make it loose track of the target even at close distances. Also the walking speed of a person can considerably affect the performance of tracking, which sometimes fails even if the user keeps still and looks at the robot.

3.4.3 People distance

The goal of this experiment is to understand whether *Pepper* can correctly detect how far a person is. In this case, only three *Say* blocks have been used, each of them connected to a different event input: *PersonEnteredZone1*, *PersonEnteredZone2* and *PersonEnteredZone3*. These inputs are triggered when the related event is raised. In this case, each event should be raised when a person enters a different engagement zone, and their values have been kept as default as shown in Figure 3.5. The three blocks simply make the robot say ”One”, ”Two” or ”Three” when the experimenter enters the respective engagement zone. Once started the behavior, the experimenter approached the robot from 5*m* away and waited for it to say the messages. The events were not raised in the correct order (i.e. ”Three”-”Two”-”One”) as in most of the trials the event *PersonEnteredZone3* was not raised. Moreover, the two other events were sometimes raised in the wrong order or one of them was not raised. Also in this case, the walking speed of the user can considerably affect the accuracy of the results.

3.4.4 Gaze control

The goal of this experiment is to understand whether *Pepper* can successfully detect when a person is looking at it. In this case, only two *Say* blocks were used, one of them connected to an event input *PersonStartsLookingAtRobot* and the other connected to the *PersonStopsLookingAtRobot* event input. In this case, the first event is raised when a user starts looking at *Pepper* and the other when it stops. The two blocks simply make the robot say ”Start” and ”Stop”. Once started the behavior, the experimenter looked at the robot to make it pronounce its first message, then gazed away waiting for the second. These events were not always raised when the experimenter gazed at or away from the robot. It is not clear why this happened, because in different trials under the same conditions they were raised correctly. Moreover, while the first event was raised as soon as the experimenter looked at the robot, the other one was always raised late. This delay, which has been timed using a *Time* block in *Choregraphe*, is on average around $4s$. Finally, these events could be raised only when the experimenter was within $3m$ from the robot, which corresponds to the maximum people detection threshold.

3.5 Conclusion

These experiments have been performed within a similar context to the one used for the final experiment with *Pepper*. This is why they have been structured according to the ideal conditions for the encounter, with the human looking in the direction of the robot, which is stationary in an indoor environment with bright natural lighting. People detection can be generally performed effectively up to $3m$ from the robot. However, bad lighting conditions such as too bright lights or reflecting surfaces in the surroundings should be avoided, to prevent the detection module to miss the targets. Similar conclusions can be drawn for people tracking, which works effectively up to $3m$ but its performance is highly sensitive to lighting conditions. Moreover, the events from the NAOqi framework are not reliable to detect users' distance or users' gaze, because they are either not raised or raised late. These results already suggest that some of *Pepper*'s on-board functionalities need to be extended, as they have a limited perception range and lack of robustness. However, a more thorough analysis of the interactive skills of the robot will be conducted with a user study, to further define which features need to be improved.

Chapter 4

Baseline application

As described in Section 1.3, the on board functionalities of *Pepper* need to be tested with a user study to understand whether they are already satisfactory for the goal of this project. The approach chosen to answer this question is to create a simple application in *Choregraphe*, the default programming tool to control Softbank robots, to deploy baseline interactive skills and verify their effectiveness. This application has been tested during a quick pilot experiment aimed at understanding how users responded to the behaviors of *Pepper*. In Section 3.4 the sensing capabilities of the robot have already been investigated with preliminary experiments, to validate the information from the specification manual within our context. In this section, a more global analysis is conducted to provide a basic comprehension of: the interactive capacities of the robotic platform, the role of the human factor, and the appropriateness of the tools used. The building of Computer Science in TU Delft campus at Van Mourik Broekmanweg 6 (VBM building) has been the site where the experiment took place.

4.1 Baseline application

4.1.1 Motivation

This baseline application provides a basic interactive activity for *Pepper* that can be tested with users before implementing any off board functionality. Its purpose is to highlight whether there are some capabilities of the robotic platform that need to be improved and, in case there are, which of them have to be prioritized to successfully initiate engagement. As this analysis is mainly qualitative, the application scenario is not similar to the one of the final experiment, where *Pepper* will not actually interact with users. The application has been structured as a dialog in which the robot uses its *autonomous life* mode to display the social cues already embedded in its architecture, such as simple gestures, blinking and vocal tone variations according to sentence format. The whole application is going to be stationary to avoid variations of context and the necessity of socially aware navigation. This choice has been made also considering that some functionalities of the robot are not available once in motion (as explained in section 3.4).

Originally, two possible contexts were considered for the baseline application: the break area with the coffee machine and the space in front of the elevator, at the 6*th* floor of the building of Computer Science. These spaces have been chosen under the assumption that they can easily be the setting of a short interaction, because in both contexts users will be waiting either for the coffee to be ready or for the elevator to arrive. It is therefore expected that they will have time to talk to *Pepper* without perceiving it as annoying. However, when deploying the robot in the field, some issues were found with the elevator scenario. In fact, the hallway in front of the elevator is not wide enough to allow for both a robot and a person to talk comfortably without obstructing the passage to other people. Moreover, the waiting time for the elevator cannot be known a priori and could also be too short for the whole interaction, inducing the user to interrupt it abruptly. For such reasons this scenario has been discarded, while the break area with the coffee machine was selected for further development. This context allows for a more flexible duration of the interaction and provides a more comfortable setting for the conversation. Moreover, it is less likely that users will not have enough time to interact with *Pepper* (as in the elevator scenario).

4.1.2 Description

The application has been developed in *Choregraphe* as a behavior, which is uploaded on *Pepper* and is given some trigger conditions. The first condition activates the application when a person is detected in *EngagementZone2*, i.e. between 1.5m and 2.5m distant from the robot (Chapter 3). However, due to sensory limitations, it is possible that if a user moved quickly, it could go from *EngagementZone2* to *EngagementZone1* (closer to the robot) without being detected. This is why a second analogous condition is introduced, which makes the behavior activate also if a user is detected in *EngagementZone1*. A third condition prevents *Pepper* from restarting the application immediately after it has ended, because the interlocutor may still be in the engagement zones after the conversation. If the interacting user is only one, it is likely that he or she will leave soon after finishing to talk with *Pepper*, so a short time span before restarting the application could be enough. However, if two or more people go to take coffee together, it is possible that they will stay longer at the coffee machine. Then it would not feel natural if the robot restarted the same conversation while they are still around. In order to find a waiting time suitable for both these cases, the robot has been set to wait 2 minutes before repeating the same behavior. Nevertheless, sometimes two or more people can spend longer time in the break area to chat. In such unpredictable situations the behavior of the robot is deactivated (and later on reactivated) by the experimenter using the *Choregraphe* interface, so that *Pepper* will not restart talking to the same users.

Once the behavior has been started, an input signal is sent to a sequence of *Say-Delay-Say* blocks, used to respectively get user's attention, wait 1s, then repeat the initial utterance accompanied by a greeting. This ”pause and restart” method to draw attention has been experimented by Pitsch *et al.* [48], to ensure attention of distracted or not attending users. Subsequently, a *Python Script* block is used to lower the threshold of speech recognition accuracy from 50% to 40%. When the robot was using the highest threshold, users would often need to repeat themselves more clearly to obtain a response from the robot. However, it has been tested that utterances can often be understood correctly also with the lower threshold, which benefits to the rhythm of the interaction. A *Dialog* block follows, which contains the main body of the application with questions and answers that *Pepper* will give according to users' response. To make the conversation more involving, the tablet is used to display images of some objects mentioned. This is done using the block *Show Image*. Figure 4.1 shows how the baseline application has been implemented on *Choregraphe*.

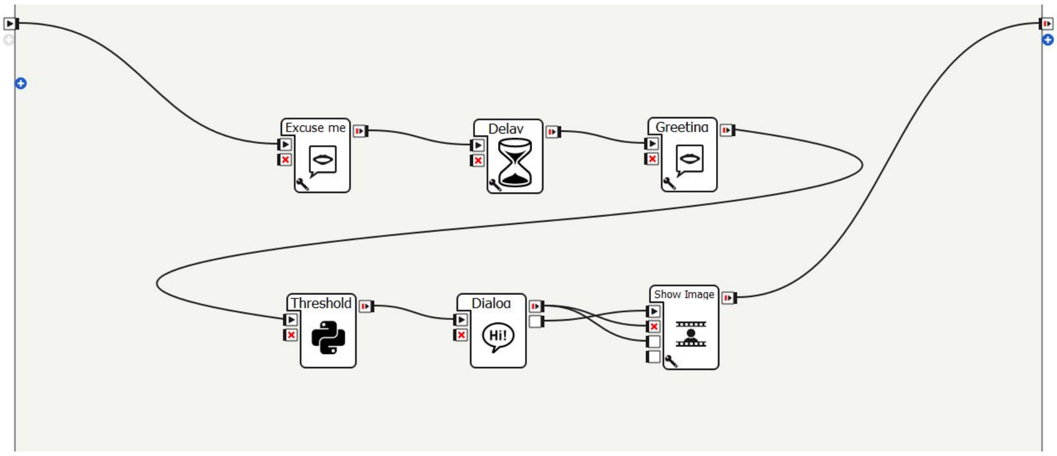


Figure 4.1: Baseline application in *Choregraphe*

This baseline application exploits several modules of the NAOqi framework in order to deploy *Pepper*'s interactive skills. To begin with, the *ALEngagementZones* monitors the presence of people in the engagement zones in front of the robot and triggers the application when at least one person is detected. This module works together with the *ALPeoplePerception* module, which keeps track of the users detected, and with the *ALFaceDetection* module, which recognizes the faces of users and make the robot look at them. These modules use the information from *Pepper*'s cameras and 3D depth sensor, they are activated by the *autonomous life* mode so they keep working throughout the whole application. The *Say* and *Dialog* blocks use the module *ALAnimatedSpeech* to make the robot speak and perform contextual movements during the conversation. Within the *Dialog* block, this module coexists with the *ALSpeechRecognition* module, which is responsible for the interpretation of what users say and associates it to a confidence value. These modules use *Pepper*'s speakers and microphones to communicate with users, and they are active only if the correspondent blocks have been started.

The dialog between *Pepper* and the user is expected to last less than 1 minute. The structure of the conversation aims at simulating the interaction between two people, one familiar with the settings (the user) and one that is new to the environment (the robot) and asks question about it. In fact, the people interacting with *Pepper* will most likely be workers in the building. Hence, the robot asks simple routine questions about the coffee machine and how it works, as illustrated in Figure 4.2. The sequence of actions that *Pepper* follows based on this behavior are the following:

1. *Pepper* waits for a person to approach in *autonomous life* mode;
2. Once a user approaches and enters *EngagementZone2* or *EngagementZone1* the application is triggered;
3. *Pepper* tries to attract the attention by saying ”Excuse me”, then waiting for one second before repeating the same utterance accompanied by a greeting, using a ”pause and restart” technique;
4. *Pepper* asks information about the context it is set within, to act like someone not familiar with the place;
5. According to user's answer *Pepper* asks a second question and displays an image on its tablet, otherwise disengages;
6. If it has not disengaged yet, after the answer to the second question *Pepper* disengages greeting the user.

The questions asked by the robot are designed to (1) be easily answered by any person familiar with the working place; and (2) allow for closed answers. This is particularly important when programming in *Choregraphe*, because the *Dialog* block that controls the conversation flow has to receive a known input from users after every question. In fact, if the users does not use an expected answer, the robot just keeps silent and does not pronounce the next utterance.

4.2 Pilot experiment

4.2.1 Procedure

The baseline application has been tested with a pilot experiment, aimed at revealing whether the on board functionalities of *Pepper* allow to successfully initiate engagement. In case they do not, the experiment will highlight which aspects of the interaction require major refinement. The experiment has been observed from a distance to avoid any influence on users' performance. The robot has been deployed in the break area next to the coffee machine at the 6*th* floor of the building of Computer Science. The procedure of the experiment is as follows:

1. The experimenter places *Pepper* in the break area next to the coffee machine. The robot is switched on but not in *autonomous life* mode.
2. The experimenter leaves the robot and observes the setting from distance.
3. The experimenter enables the *autonomous life* mode of *Pepper* so that the robot ”wakes up” and is ready to begin the interactive activity.

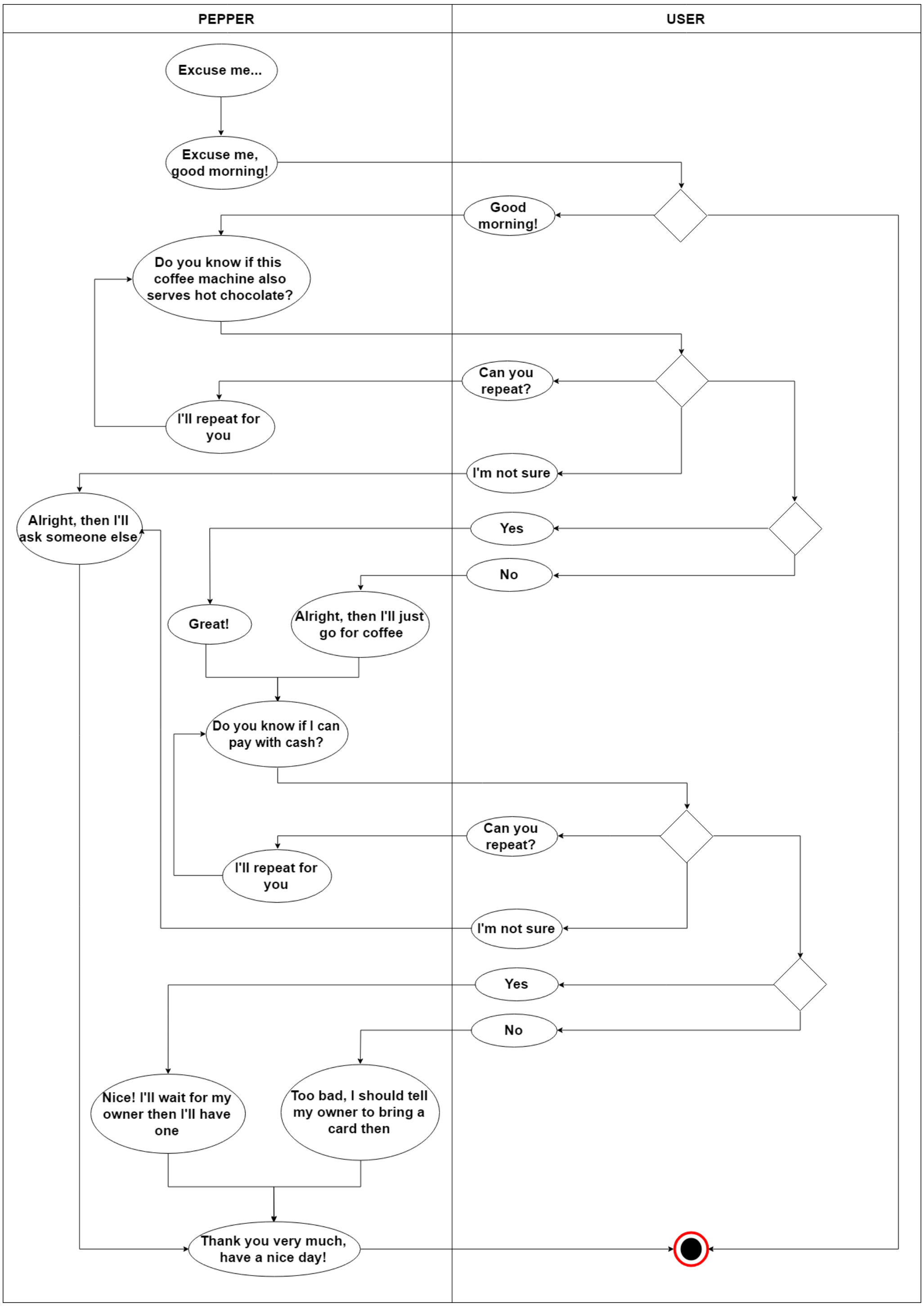

PEPPER
USER
Excuse me...
Excuse me, good morning!
Good morning!
Do you know if this coffee machine also serves hot chocolate?
Can you repeat?
I'll repeat for you
I'm not sure
Alright, then I'll ask someone else
Yes
No
Great!
Alright, then I'll just go for coffee
Do you know if I can pay with cash?
Can you repeat?
I'll repeat for you
I'm not sure
Yes
No
Nice! I'll wait for my owner then I'll have one
Too bad, I should tell my owner to bring a card then
Thank you very much, have a nice day!

Figure 4.2: Dialog flow of the baseline application.

4. When a participant approaches, the trigger conditions of the interactive activity make *Pepper* begin the interaction.
5. *Pepper* will try to execute the script of the conversation flow of the *Choregraphe* behavior.
6. If the end of the script is reached, *Pepper* disengages from the participant.
7. The participant leaves the break area.
8. The experimenter approaches the participant and hands out a questionnaire about the experiment.

The 7th step of this procedure will probably not happen all the times, as some participants may want to drink their coffee in the break area or, if there is more than one, stay there to have a chat. In such cases, the experimenter will disable the trigger conditions of the baseline application to avoid that *Pepper* restart the conversation after 2 minutes. Then, participants will be approached, debriefed and interviewed as explained in step number 8.

4.2.2 Method

The pilot experiment has the purpose to provide a qualitative evaluation of the on board functionalities of *Pepper*. The main tools that are going to be used to conduct this evaluation are the observations of the experimenter, noted during the experiment, and the post experimental questionnaires filled in by participants.

4.2.2.1 Objective metrics

During the experiment, the researcher has the tasks of observing the interaction and recording the limitations of the application. The focus is to test the performance, of people detection and face tracking as discussed in Section 3.4. These can be summarized in the following set of metrics:

- Track lost: indicates if the robot has lost track of the participant's face during the conversation.
- Late detection: indicates if the robot detected the participant while approaching or once already very close.
- Missed answer: indicates if the robot did not understand what the participant said and did not continue with the next utterance. This moment is really important for the interaction because it may either induce the participant to repeat its answer or to leave.
- Completion of ”happy flow”: indicates if a participant got to the end of the whole script or abandoned it earlier.
- Time: the duration of the whole interaction is also taken into consideration.

4.2.2.2 Subjective metrics

The paper based questionnaire that participants will be asked to fill in has the purpose to obtain participants' feedback on *Pepper*'s capabilities. Questions have been chosen in relation to the objective metrics, in the format of a *Likert Scale* of 5 points:

Q1 `The robot noticed my presence`: this question aims at understanding if the participant feels that the robot detected him/her on time.

Q2 `The robot was attentive to me during the conversation`: indicates whether the robot looked distracted and not attending.

Q3 `The robot interpreted correctly what I said`: focuses on the performance of *Pepper*'s speech recognition, and if users felt understood when talking.

Q4 `The verbal and bodily expressions used by this robot felt natural`: tests whether the robot's behavior seems natural and understandable to participants during the conversation.

Q5 `I find it easy to interact with a robot.`

Q6 `I find it intuitive to interact with a robot.`

Q7 `I appreciated seeing a robot near the coffee machine`: aims at understanding whether the presence of a robot in a public space (e.g. near the coffee machine) is accepted by users.

Finally, any suggestion or recommendation for future developments is asked, to collect a more personal feedback from users. The questionnaire is attached in Appendix A.

4.2.3 Results

The baseline application was tested during the pilot experiment with 11 participants in total, 9 males and 2 females, all of them were master students of TU Delft. Only one among them had previously interacted with *Pepper*, while the others had never interacted with a robot before. The observations of the experimenter are summarized in Figure 7.3.3, which shows the metrics indicative of *Pepper*'s performance, and Figure 4.4, which presents the duration of the interaction for every user.

As shown in Figure 7.3.3, *Pepper* lost track of 7 participants throughout the trials without being able to track their faces again, at different times of the interaction. For what concerns late detections, 5 participants were noticed only when they were in front of the robot (less than 1m from it), while the others could be seen when they entered *EngagementZone1* or *EngagementZone2*. As for misunderstood answers (missed input from user), the robot could not interpret participants correctly in 7 interactions, sometimes not even after they repeated the answer twice or three times. In 4 of these cases participants abandoned the application before completion, but were approached and asked to complete the questionnaire anyway.

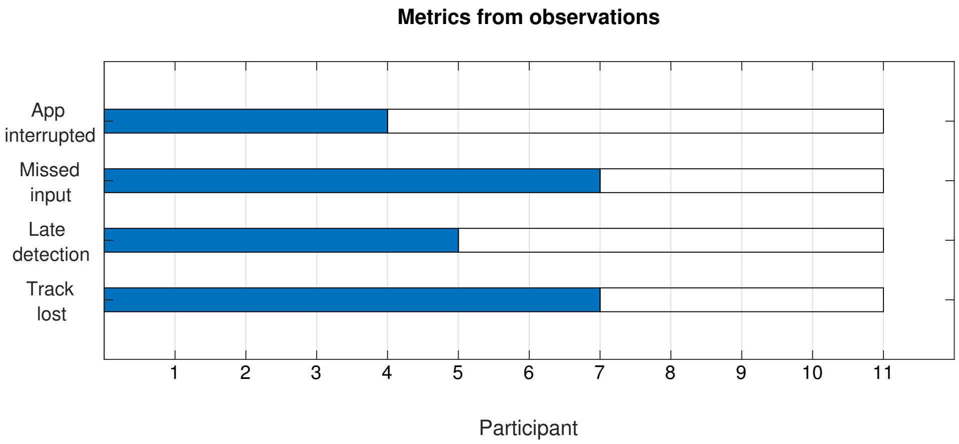


Figure 4.3: Analysis of the metrics observed during the experiment.

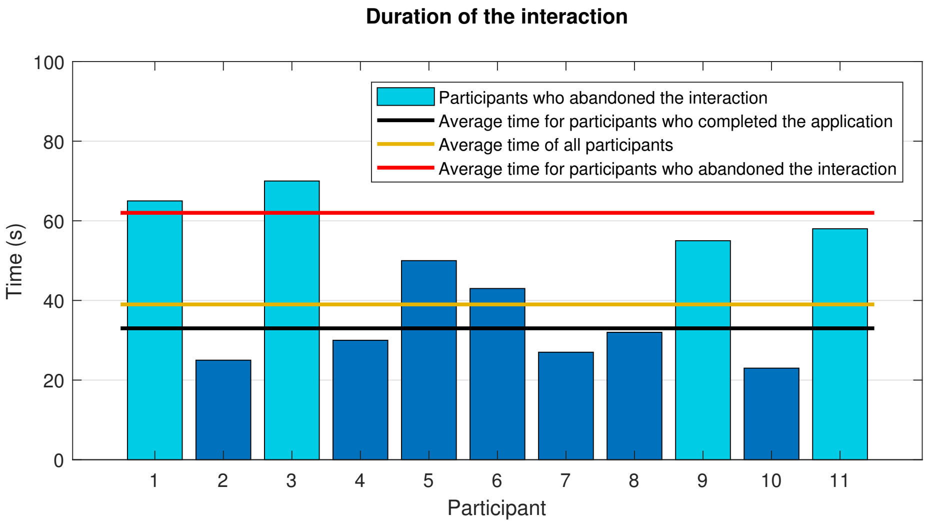


Figure 4.4: Duration of the interaction with *Pepper*.

The average duration of all the interactions with *Pepper* was around 40*s*, but this value drops to 33*s* if only those who completed the ”happy flow” script are considered. The average raises to 52*s* if we take into account only the users who had problems being understood by *Pepper*. In these cases, users waited some time before repeating themselves, probably assuming the robot understood them and was about to reply. If these pauses were too long and the robot could not yet understand participants once they repeated themselves, then the application was abandoned. Considering only the users who left before the application was over, the average time raises to 62*s*. Nevertheless, the abandonment of the application may not be caused only by issues with speech recognition, because in these 4 cases *Pepper* lost track of users 3 times and did one late detection. These factors may also have contributed to make the participant leave the interaction.

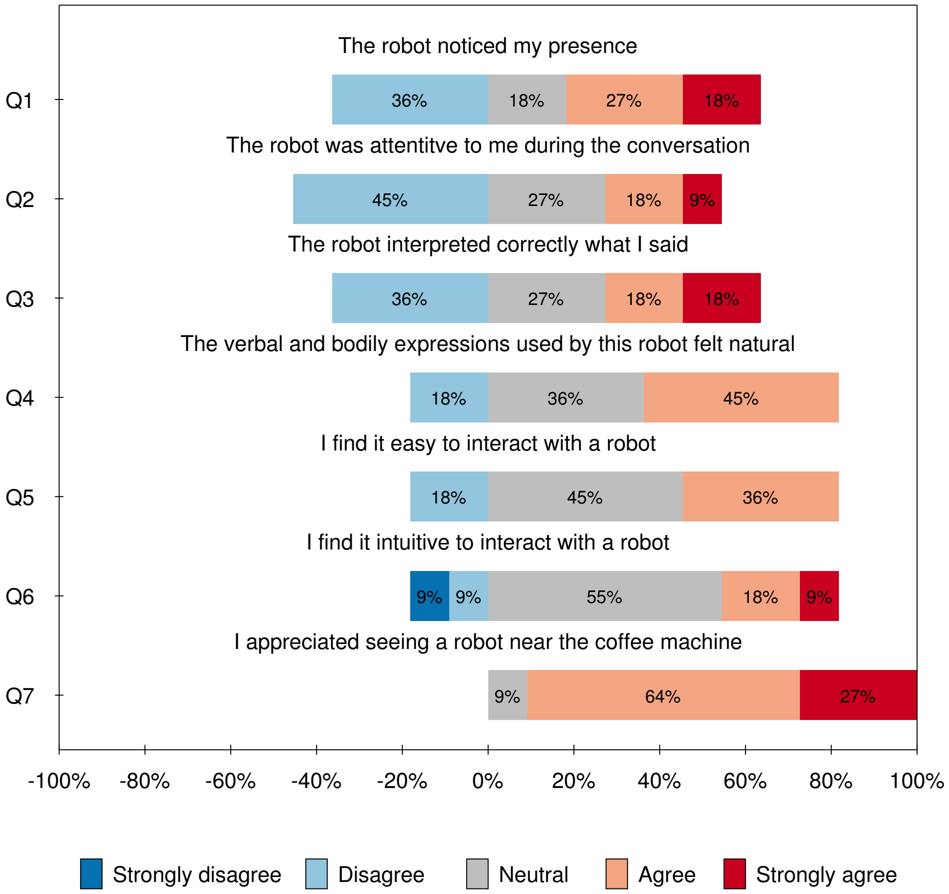


Figure 4.5: Results from questionnaires presented as a Likert Scale Plot.

The answers of the questionnaires have been collected and summarized in Figure 4.5. The outcome of the experiment confirmed the limitations of on board functionalities emerged during the preliminary analysis presented in Section 3.4. Question 1 (Q1) corroborates this idea, as only 5 participants felt that *Pepper* noticed their presence while 4 experienced that the robot was not aware of them. Question 2 (Q2) is in line with these findings, because only 3 participants perceived *Pepper* as attentive during the conversation, while 5 thought it was distracted. Question 3 (Q3) highlights the same trend, showing that *Pepper* could interpret correctly only 4 participants throughout the whole trial. On the contrary, question

4 reveals that verbal and gestural expressions used by the robot were mostly perceived as natural, suggesting that the on board *autonomous life* mode is probably already satisfactory to conduct short interactions.

As for the overall impression about the interaction with *Pepper*, analyzed in questions 5 (Q5) and 6 (Q6). Only 4 users found the interaction easy and 3 intuitive, while one participant did not find it intuitive at all. Nevertheless, question 7 (Q7) shows that the idea of having *Pepper* next to the coffee machine was highly appreciated by almost the totality of users (10), regardless of the performance of the robot during the experiment. The main suggestions and recommendations from users were related to speech recognition and face tracking. One user wrote ”the robot was not looking at me and moving too much”, while the comment ”the robot could not understand me” was more frequent. Some suggested that, in order to make the application more intuitive, the robot should say when it did not understand, so that people would know they have to repeat themselves, while others expected the robot to look more at them.

4.3 Discussion

The results obtained from the observations and the questionnaires helped to evaluate the performance of *Pepper* during the interactions and participants’ impression about it. The main aspects that can be discussed with these findings are: people detection and tracking, dialog flow and initial engagement.

4.3.1 People detection and tracking

Pepper’s sensitivity to lighting had a major influence on its capability to detect people. In 5 of the trials (out of 11), it could not detect users until they were closer than $1m$ because it was distracted by the lights. This means that the robot was staring at the artificial lights from the ceiling, most likely mistaking them for faces. The late detection of participants is also due to the fact that distance is estimated using only the 3D depth sensor in the right eye of the robot. Therefore, if the robot is distracted and does not look in the direction from which a person is approaching, neither of the conditions of having people in *EngagementZone1* or *EngagementZone2* are triggered. As a consequence, participants assessed *Pepper* as aware of their presence only in 5 cases. If the robot was using also its sonars to estimate distance, it would probably be able to detect users nearby without necessarily looking at them, then turn toward them and perform detection. However, even in those cases where the robot was looking toward a person approaching, the detection algorithm would not work unless the person was closer than $3m$. Also people tracking was hindered by the sensitivity to lighting, because *Pepper* often mistook lights for faces. In 7 trials, this occurred even when the interaction had already started, when the robot lost track of the user and continued talking while not looking at it. This issue had been already explained in the Softbank robotics manual and observed also during some preliminary experiments of Section 3.4. Users seemed particularly sensitive to this aspect, as only 3 of them felt the robot was attentive to what they said. Moreover, similarly to the people detection algorithm, also people tracking has a maximum range of perception up to 3*m* from the robot. This limitation is not due to the cameras of the robot, which can see people even from more than 5*m* away, but to the on-board algorithms themselves.

4.3.2 Dialog flow

The rhythm of the interaction was considerably affected by the limitations of speech recognition, as demonstrated by the 7 trials in which users had to repeat themselves. If *Pepper* could understand the input after one repetition, participants continued the interaction because the robot moved on with the script. If they had to repeat themselves twice or more and still did not receive any feedback, they eventually abandoned the application. This was the case of the 4 users who did not reach the end of the script, because they could not be understood by the robot even after two repeated answers to either the first or the second question. When the robot did not hear what a user said with a speech recognition accuracy of at least 40%, it kept waiting in silence until the input was repeated more clearly. The performance of speech recognition could have also been affected by the background noise of the setting chosen, but it seems clear that this idle behavior of the robot discouraged participants to continue the interaction. In fact, less than half of users felt *Pepper* was understanding them correctly. These elements contributed to slow down the conversation and make feel unnatural the rhythm of the interaction.

4.3.3 Initial user engagement

As anticipated, the way *Pepper* drew users' attention was based on a ”pause and restart” technique inspired to the research conducted by Pitsch *et al.* [48]. This method was not always successful to engage users in a conversation, because it relied on a specific reaction from the addressee. People were expected to turn to the robot after the pronunciation of the first utterance, then take initiative in starting the conversation. If users did not reply to this initial approach, *Pepper* would just stay silent staring at them, waiting for verbal inputs. On the other hand, if a user tried to approach *Pepper* too early, it would have been interrupted by the robot starting the behavior of the application. This explains why most users felt neutral about the simplicity and intuition of the application, while some of them even disagreed with these statements. In spite of the ”pause and restart” strategy chosen, the limitations of the programming environment would always require users' input to start a conversation, which does not make the interaction flexible to the context. During the pilot experiment *Pepper* was in *autonomous life* mode, which makes the robot simulate breathing and use gestures coherent with the dialog flow. Participants appreciated the expressions generated by this mode during the conversation, but outside the interaction it made *Pepper* behave awkwardly or look distracted (e.g. when it was staring at lights mistaking them for faces). In these situations, if the robot had looked more attentive, it would have probably aroused users' interest to approach. Nevertheless, the fact that the robot ”looked alive” triggered the curiosity of users and induced them to talk to it anyway. However, this interest in interacting with the robot could also be related to the novelty effect rather than to its *autonomous life* mode.

4.4 Conclusion

The experiment highlighted the main limitations of *Pepper*'s behavior and demonstrated that its embedded functionalities are not sufficient to ensure the naturalness of the interaction. Among the aspects discussed above, people detection and tracking as well as initial engagement have been selected for further improvement, while dialog flow is left for future work. The reason of this choice is that in the establishment of initial engagement dialog plays a minor role, while better people detection and tracking functionalities can generate more successful strategies of initial engagement. In fact, even if most of users experienced issues related to the conversation with *Pepper*, this is important mostly for the maintenance of engagement. In fact, those who abandoned the interaction did it because they could not keep talking with *Pepper*, but they approached it because the robot noticed them and drew their attention. Therefore, issues related to lighting have to be solved firstly by choosing a more appropriate setting of the experiment and implementing more robust detection and tracking algorithms on the robot. Staring at bright lights or at reflecting surfaces hinders people detection, compromising the initiation of an interaction. The robot needs off board functionalities to extend its detection range beyond $3m$ and ensure real-time accuracy. Finally, the robot has to display specific social cues to draw attention more effectively. In particular, these behaviors need to be deployed when users are not close to the robot yet, to convey the impression that *Pepper* is aware of them even when they are approaching it. A combination of social cues, maybe even in synchrony with speech (e.g. waving and greeting) could result in more successful techniques to elicit initial engagement.

Chapter 5

Off-board feature detection

When people enter a building, they usually spend some time looking around and exploring the surroundings, especially if they are not familiar with the place. While doing so, they can either stop or continue walking, but in both cases their attention is more likely to be drawn by something that is in the new setting they entered. If a social robot like *Pepper* is present in the entrance hall of this building and welcomes people when they enter, it is probable that it will successfully draw people's attention and maybe even make them approach. However, this is possible only if *Pepper* is able to correctly detect people and their social cues on time, which depends on its sensors and capabilities. Therefore, the purpose of this chapter is to identify which off-board functionalities can allow *Pepper* to successfully draw users' attention.

One of the main objectives of this project is to exploit exclusively *Pepper*'s on board algorithms to monitor interactions with users. The motivation of this choice lies in the intention of making the application more compact, so that only the robot itself would be needed to reproduce the experiment in another scenario. However, *Pepper*'s CPU does not have enough power to run different state-of-the-art techniques simultaneously. Therefore, the robot has to be complemented with the necessary hardware (e.g. GPU) to process the data obtained from the sensors. This will improve the speed and performance of people detection and enable *Pepper* to detect major features of users entering the building. The information collected can then be exploited to elicit initial engagement and correlate it to specific attention drawing behaviors. To do so, three main requirements need to be taken into consideration.

The first requirement is real-time data processing. The robot is going to be deployed in a real-life scenario, so it has to respond to people in real-time. This will make *Pepper*'s behaviors look more natural and smooth, but also prompt them at the right time after the person has entered (i.e. not too early when the person might still be too far to hear or see, or too late when the person might have already gone). Moreover, the robot also needs to record data with real-time precision, so that people's reactions or social cues can be correctly associated to the behavior that generated them. Therefore, the processing time of each algorithm is going to be taken into account.

The second requirement is detecting features that indicate if users are paying attention to *Pepper*. The robot has to be able to detect relevant social cues such as proxemics and gaze direction, while others are going to be recorded on the observation sheet (i.e. gestures). It is important to understand how people respond to *Pepper*'s attention drawing behaviors and study how their social cues change when initial engagement is elicited. Also in this case, if the robot is able to record this information autonomously and accurately, it will be possible to study the relationship between its behaviors and participants' reactions.

The third requirement is extending the range of detection within the experiment setting. The robot is going to be stationary in the entrance hall of the building of Computer Science (VMB building), and several users are going to enter and leave its field of view. It has to detect salient features in real-time, but it also has to do it within a certain range. Only in this way it will be possible to study the evolution of users' reactions in response to what the robot does. A minimum detection range of $8m$ would be optimal for the algorithms investigated, in order to ensure that the robot will start collecting data as soon as participants step into the building, regardless of its location.

The selection and implementation of the off board functionalities described below has been done in collaboration with Elie Saad, PhD student at the department of Intelligent Systems - Interactive Intelligence group.

5.1 People detection

As anticipated in Section 3.4, *Pepper*'s built in functionalities for people detection are limited within a range of $3m$ from the robot. However, detecting people from further away can provide additional information about their engagement and willingness to interact. For instance, interrupting or deviating from a walking trajectory to approach the robot could be a signal of interest or curiosity [41]. In order to analyze these behaviors and their correlation with initial engagement, we complemented Pepper's capabilities with a people detection technique. For this purpose, two new state-of-the-art vision techniques are tested and compared by taking into account their computation time and hardware requirements. The first technique, *Detectron* [19], is a novel approach which uses object masking by drawing complex polygons around objects rather than bounding boxes. This feature can be useful for tracking moving objects (in this case, walking people). The second technique, *YOLOv3* [50], is a real-time object detection system which is expected to be fast and accurate.

5.1.1 Detectron

The *Detectron* project started in July 2016 by Facebook AI Research and released in January 2018. It has the aim of developing a fast and flexible object detection algorithm written in Python and powered by the Caffe2 deep learning framework [19]. *Detectron* includes implementations of multiple object detection algorithms, which include Mask R-CNN [27]; RetinaNet [36]; Fast R-CNN [18]; Faster R-CNN [51]; and R FCN [10].

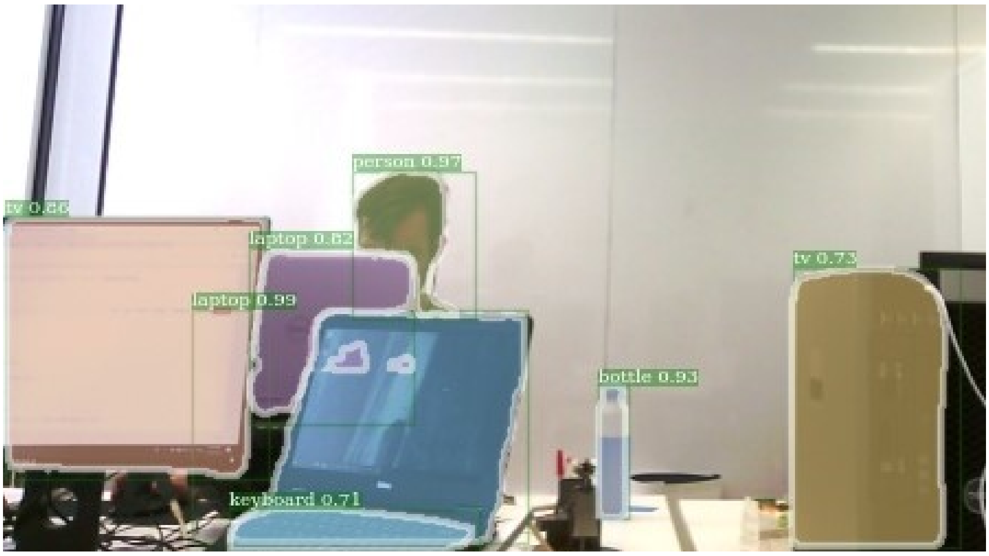


Figure 5.1: An example of object masking done with *Detectron*.

One of the main novelties of this algorithm is object masking: instead of just drawing a bounding box around the image of an object, as it happens in the current state-of-the-art detection algorithms, *Detectron* actually draws a complex polygon around the object, as shown in Figure 5.1.

5.1.2 YOLO v3

You Only Look Once v3 (*YOLOv3*) [50] is a system for object detection that uses a single convolutional neural network (CNN) for both classification and localization of the object. *YOLOv3* uses a variant of Darknet [49] which originally has a 53 layer network trained on Imagenet. For detection, 53 additional layers are stacked onto it, giving a final 106 layer fully convolutional architecture. As the name says, its approach allows to look at the image just once and extract the information necessary for object detection. The architecture of this technique provides three main advantages [50]:

- Speed, as the image is run only on a single CNN at test time to obtain detections.
- Reasoning globally about the image while making predictions, thus making less background errors than a Fast R-CNN.
- Learning generalizable representation of objects. The algorithm is less likely to break when applied to new domains.

YOLOv3 takes an input image and makes predictions at three different scales, which are given by down sampling the dimensions of the input image by 32, 16 and 8. At every scale, the algorithm divides the image into an $S \times S$ grid. Each cell of the grid predicts B bounding boxes and associates to everyone of them a confidence score. Such score indicates how sure the model is that a specific box contains an object and also how accurate it is about the object class predicted for that box. If no object is present in the box then its score should be zero, otherwise it has to be equal to the intersection over union (IOU) and is defined as $Pr(Object) * IOU$. The IOU compares how the bounding box given during prediction overlaps with the ground truth (from training and test data) bounding box. Its value is obtained dividing the overlap area between the boxes by the union of those areas, therefore ranges between 0 and 1.

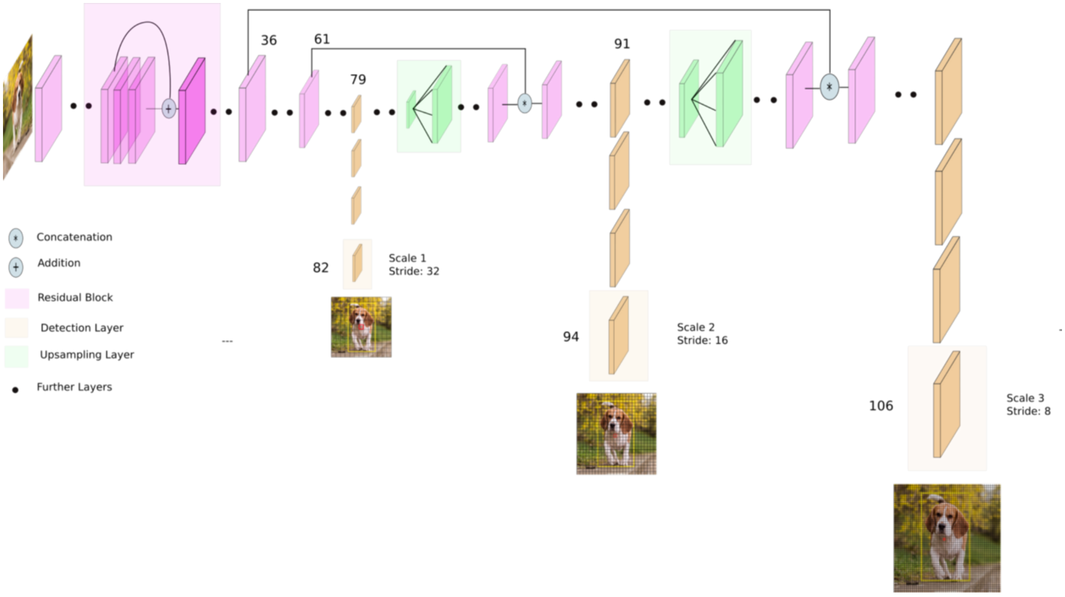


Figure 5.2: *YOLO v3* network architecture [50].

Every bounding box has 5 predicted values: *x*, *y*, *w*, *h* and a *confidence*. The (*x,y*) values are the coordinates of the center of the box with respect to the boundaries of the grid cell. The (*w,h*) values stand for width and height of the object, and are computed relatively to the whole image. Lastly, the *confidence* value represents the IOU between the box and any ground truth box. Detection occurs three times at three different scales along the network. The image is initially down sampled, then undergoes few convolutional layers before the first detection. Then the image is up sampled and the process is repeated until three detections are performed, as illustrated in Figure 5.3.

The detections at three different scales make the algorithm more precise in detecting small objects. *YOLO v3* claims to be as accurate as state-of-the-art algorithms while being considerably faster. This performance is achieved through training in the COCO dataset [37] at mean average precision (mAP) 50 benchmark, which corresponds to an accuracy of 0.5. However, in higher benchmarks (i.e. COCO 75) where the boxes need to be aligned better not to reject the prediction, *YOLO v3* is less accurate than other algorithms [50].

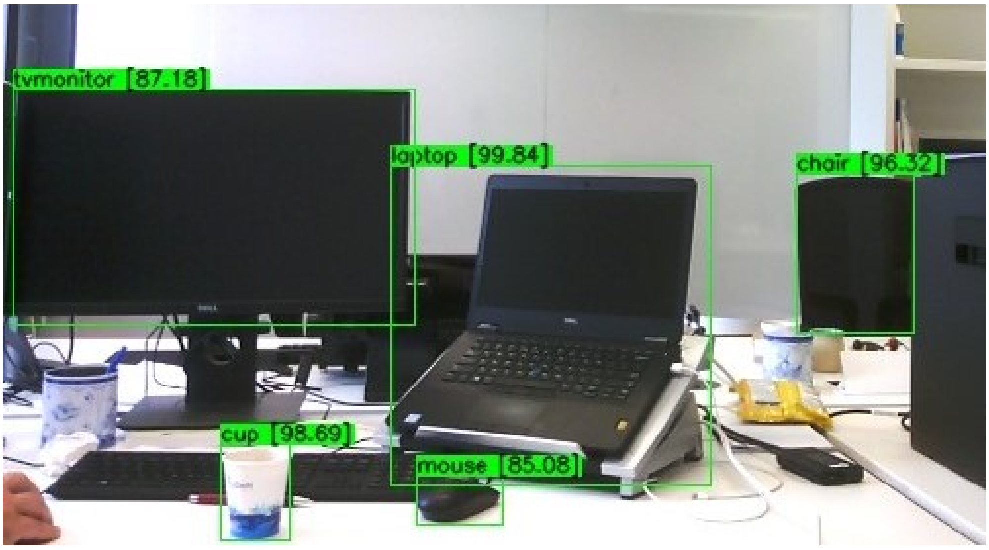


Figure 5.3: An example of object recognition done with *YOLO v3*.

5.1.3 Results

Both *Detectron* and *YOLOv3* have been tested on a GeForce GTX TITAN X GPU, using the real-time video stream from the upper camera of the robot. With both algorithms the robot could detect objects much farther away than needed for the experiment. In fact, they were able to recognize cars parked outside the building around 20*m* away form the main entrance. Therefore, both the algorithms are suitable for the experiment under this point of view: they can successfully detect people within a range of 8*m* from the robot. Moreover, none of them showed any limitations related to the lighting conditions of the setting, as it occurred for the on-board functionalities. The new algorithms have been tested in the setting of the final experiment, described in Chapter 7, and they performed well in spite of the bright light coming through the glass doors of the entrance. Therefore, a tangible improvement for people detection and tracking can be achieved. For what concerns real-time data processing, while *Detectron* took 0.16*s* to process a single frame (results confirmed by the authors as well [19]), only 0.03*s* was needed by *YOLOv3*. This aspect is crucial to ensure real-time performance of the robot and make its behaviors as smooth and natural and possible. Therefore, these results show that *YOLOv3* is more suited for our experiment, being able to process the frames faster than *Detectron*.

5.2 Depth estimation

Pepper has a built-in depth estimation system that uses a 3D sensor placed behind the right eye of the robot. However, also in this case the on board sensors have a range of perception that is reliable only within 3*m* from the robot. Multiple depth estimation techniques are currently available [16] [38] [61]. However, these techniques use stereo vision to estimate depth, which cannot be performed with the currently available Pepper. This is the reason why we attempted to infer depth using monocular vision.

5.2.1 Unsupervised Monocular Depth Estimation

For this project, we selected the Unsupervised Monocular Depth Estimation with Left-Right Consistency developed by Godard *et al.* [21]. Monocular depth estimation refers to the problem where only a single image is available at test time. The authors' approach the task as an image reconstruction problem during training. The model is fully convolutional and does not require any depth data. It learns to estimate depth predicting correspondences at a pixel-level between pairs of stereo images that have a known camera baseline. Results are then refined with a loss module that takes into account smoothness, reconstruction and left-right consistency terms. At training time the model has access to two images I^l and I^r, which are the left and right color images from a calibrated stereo pair captured at the same time. It learns how to find a dense correspondence field d^r that allows to reconstruct an image (e.g. the right one) given the other one of the pair (the left one). The reconstructed images obtained from the original left and right images respectively, are indicated as $\tilde{I}^r = I^l(d^r)$ and $\tilde{I}^l = I^r(d^l)$, as shown in Figure 5.4.

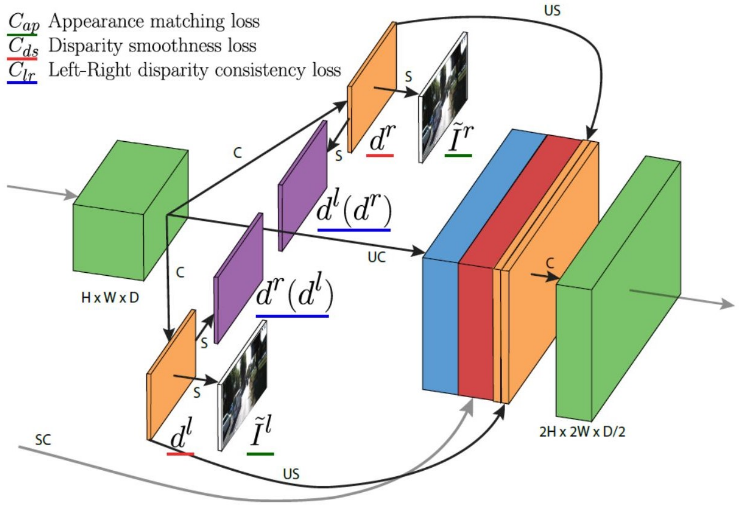

C_{ap} Appearance matching loss
C_{ds} Disparity smoothness loss
C_{lr} Left-Right disparity consistency loss
US
S
C
S
d^r
$\tilde{I}^r$
$d^l(d^r)$
UC
C
$d^r(d^l)$
H x W x D
C
S
S
d^l
$\tilde{I}^l$
SC
US
2H x 2W x D/2

Figure 5.4: The loss module outputs left and right disparity maps d^l and d^r [21].

Under the assumption that the images are rectified, they have a known disparity d, a given distance between the cameras b and a fixed focal length f. The depth $\hat{d}$ can be trivially predicted as:

$$\hat{d} = \frac{bf}{d} \tag{5.1}$$

The network infers depth from both the disparities that warp the left image to match the right one and vice versa. It learns to predict the disparity maps for both views by sampling from the opposite input images. This process still requires a single image as input for the convolutional layers, while the other one of the pair is used only during training. Enforcing the maps to be consistent with each other ensures more accurate results. The network is inspired by DispNet [40], and has been adjusted not to need supervision in terms of ground truth depth. It outputs disparity maps at four different scales, with doubled resolution from one scale to the subsequent one. The model defines a loss C_s at every scale, which is computed as:

$$C_s = \alpha_{ap}(C^l_{ap} + C^r_{ap}) + \alpha_{ds}(C^l_{ds} + C^r_{ds}) + \alpha_{lr}(C^l_{lr} + C^r_{lr}) \tag{5.2}$$

As illustrated in Figure 5.4, the value C_{ap} makes the reconstructed image appear similar to the respective training input by comparing them at a pixel level. C_{ds} makes disparities locally smooth and C_{lr} ensures disparity coherence. The values α_{ap}, α_{ds} and α_{lr} are the weights of the different loss components. The model is trained on rectified stereo images from the KITTI 2015 dataset [17]. The model is not yet precise in depth estimation of occluded regions. In this case, pixels are not visible in both images so the disparity maps are incomplete. Another limitation is the impossibility to use single-view datasets during training: the method requires rectified stereo pairs captured at the same moment in time. Finally, the model relies on the image reconstruction term, so specular and transparent surfaces will produce inconsistent depths.

5.2.2 Results

The algorithm was tested on the GeForce GTX TITAN X GPU of the experimental setup, using *Pepper*'s cameras for the video feed. Depth estimation results were more precise for objects far from the robot than for those closer than $3m$ from it, but the overall performance did not meet the accuracy requirement of the experiment. Depth estimates were on average inaccurate of ± $1m$, resulting even larger if the target was very close to the robot (i.e. a person $1m$ distant from *Pepper* was estimated to be $5m$ away). Initially we thought that averaging the depths estimated for neighboring pixels would have improved the accuracy, but even with this modification the algorithm still generated imprecise results. Although it has a detection range of more than $8m$ and can process data in real-time, the Unsupervised Monocular Depth Estimation is not accurate enough for this experiment. Due to time constraints it was not possible to improve the performance of depth estimation, in particular for depth estimation in the short range. However, it is still going to be implemented on the robot and its refinement will be part of future work.

5.3 Head pose estimation

An important metric of initial engagement is users' head pose, because it approximates well the direction of gaze indicating where visual attention is directed. *Pepper* has a built-in functionality to estimate gaze direction, however it was already proven to lack of accuracy in Section 3.4. Therefore, three recent state-of-the-art algorithms have been tested on the robotic platform to improve head pose and gaze estimation. *Deepgaze* [47] has been selected because it proposes method for head pose estimation with convolutional neural networks (CNNs), it is based on OpenCV and Tensorflow for computer vision and machine learning. Also *OpenFace* [2] is based on OpenCV and relies on deep neural networks, but has been tested because it reportedly can train with little data achieving high accuracy and detecting not only faces, but also gaze direction and major facial features. Finally, *OpenPose* [8] has been taken into consideration because it can detect head pose, facial features but also body pose, all of them with high accuracy. Even if body pose is not used at this stage of the research with *Pepper*, it is still an important metric of user engagement which may turn useful for future implementations.

5.3.1 Deepgaze

Patacchiola *et al.* [47] propose a novel approach to head pose estimation using CNNs. Every convolutional layer has a certain number of kernels w, which are used to generate feature maps of an input image. During training, kernels are updated through back-propagation, an algorithm that minimizes a loss function using gradient information available at the current time. The training of this deep architecture is improved by dropout and adaptive gradient methods. Dropout addresses the problem of generalization on new sets of data, which can be compromised by overfitting in the memorization of the training set. This technique randomly drops units and connections during the training phase, but introduces noise in the gradients. To deal with this drawback, a higher learning rate and momentum are recommended. As for adaptive gradient methods, they are used to choose the best learning rate value to prevent the optimization process to be very slow (low value of α) or diverge (high value of α). The technique selected by the authors is Adagrad [12], an optimizer that associates low learning rates with frequently occurring features and high learning rates with infrequent ones, thus facilitating the identification of the most predictive features. Using this method, kernel weights are updated as follow:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{E[g^2]_t + \varepsilon}} \odot g_t \tag{5.3}$$

Where $E[g^2]_t$ is the moving average that represents the past squared gradients, which depends only on the previous average and current gradient, ε is a small value to avoid divisions by zero and $\odot$ indicates the matrix-vector multiplication. The algorithm uses the Viola-Jones object detection framework as face detector and 2 CNNs trained on the AFLW dataset as head pose estimator. When the image is acquired faces are isolated by the Viola-Jones algorithm and sent to the CNNs for head pose estimation.

5.3.2 OpenFace

OpenFace [2] is a Python and Torch implementation of face recognition with deep neural networks based on the paper by Schroff *et al.* [54]. The model focuses on real-time face recognition on mobile devices, so that it can be trained with high accuracy using little data. Moreover, thanks to the scientific computing framework Torch, *OpenFace* can be trained off line. The trained neural network is subsequently used in Python when new images are run through the face detection model. The faces are then normalized by an affine transformation in OpenCV so that they all point at the same direction when they are passed to the trained network. This results in 128 facial embeddings used for classification and matching. During the training phase of *OpenFace*, 500 thousand images are passed through the neural net. In this way, it is not necessary to perform such an intense training on mobile devices and it is possible to retrieve facial embeddings in real-time. In fact, the training produces the 128 facial embeddings of a generic face that are used later on to match with images, hence making the model fast. This process is illustrated in Figure 5.5.

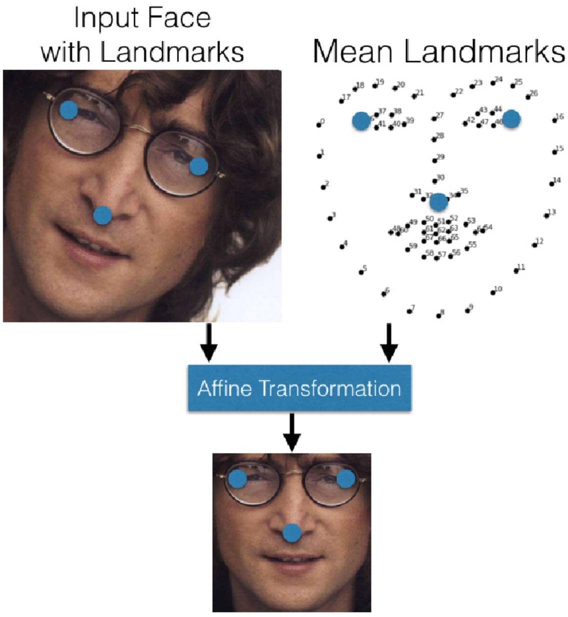


Figure 5.5: Face normalization using facial landmarks [32].

The images from the training datasets are passed to Google's *FaceNet* model for feature extraction, which uses a triplet loss function to estimate the accuracy of the network that classify faces. This is done by training on three different images: a known anchor image, another image with the same person with positive embeddings and an image of a different person with negative embeddings. In order to perform facial recognition, faces need to be isolated from the background of the image and from each others. *Dlib* together with *OpenCV* make sure that this process is robust to lighting and positions variations of faces, finding fiducial points on the face and handling the normalization of face orientation. In fact, *OpenCV*'s affine transformation makes the position of eyes, mouth and nose consistent using 68 facial landmarks detected on the image and comparing them with the values obtained from training. The normalized image is then cropped to a 96×96 pixels and input in the trained network. Once the image has been processed and passed to the trained neural network, only one forward pass on the neural network is necessary to obtain the 128 facial features that are used for prediction. The use of these low-dimensional embeddings allows to perform classification in only few millisecods.

5.3.3 OpenPose

OpenPose is a library for real-time multi-person key-point detection written in C++ using *OpenCV* and Caffe [8]. It represents the first real-time system to jointly detect human body, hands and face on a single image for a total of 130 points, and its performance is invariant to the number of people detected. *OpenPose* takes an RGB image as input and generates an output that contains all the 2D locations of key-points for all the people in the image. Firstly, a feed-forward network simultaneously predicts a set of body parts 2D confidence maps S and one of part affinities 2D vector fields L. Then, this information is combined to output the 2D key-points locations (Figure 5.6).

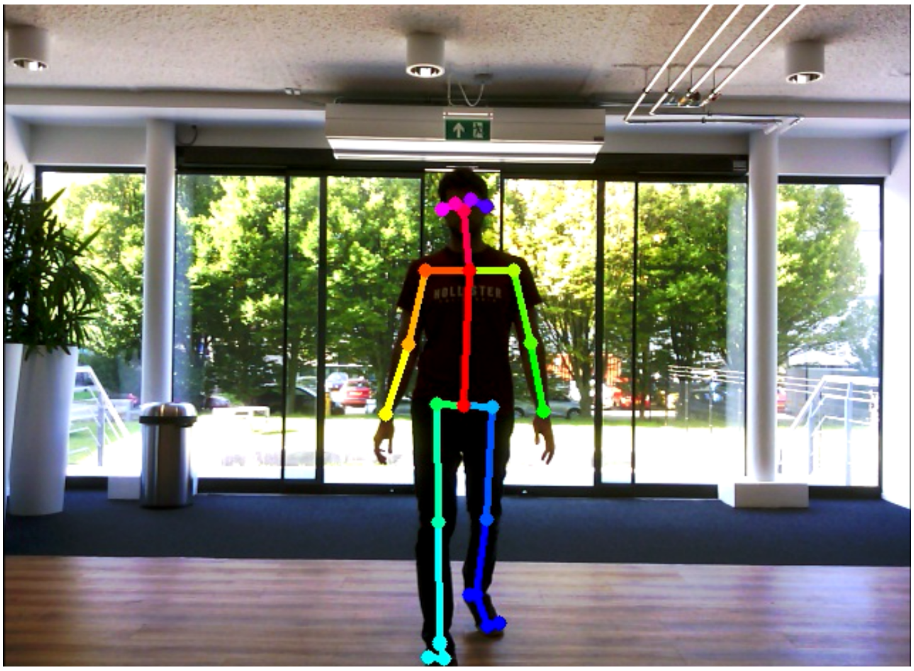


Figure 5.6: *OpenPose* skeleton data as seen from *Pepper*'s camera.

OpenPose relies on a neural network divided in two branches: one of them (ρ) predicts the confidence maps, the other (ϕ) the affinity fields, and both of them are refined over successive steps of predictions. The image is first analyzed to produce a feature map F, which is used to generate the confidence map $S^1 = \rho^1(F)$ and the affinity field $L^1 = \phi^1(F)$ of the first stage. In subsequent stages, predictions are made taking into account previous predictions and the original feature map, becoming $S^t = \rho^t(F, S^{t-1}, L^{t-1})$ and $L^t = \phi^t(F, S^{t-1}, L^{t-1})$.

To guide the network through predictions, the authors apply two loss functions at the end of each stage, one per branch respectively. To evaluate the confidence map loss function, a set of ground truth confidence maps S^* is generated from the 2D key-points. Each of them is a bi-dimensional representation of the belief that a particular body part occurs at each pixel location. So, whenever a person appears in an image, there should be a peak in all those confidence maps that correspond to the visible body parts. Once detected all the body parts, it is necessary to check if they belong to the same person. To do so, part affinity fields (PAF) are used, as they preserve both location and orientation of the identified limbs. The part affinity is a 2D vector field that pixel-wise encodes the direction of points from one body part to another. Also in this case, to evaluate the loss function a ground truth part affinity vector field L^* is defined, which averages the affinity fields of all people in the image. This PAF is compared with the vector that would be formed by connecting the candidate body parts and estimates the confidence of the association between them.

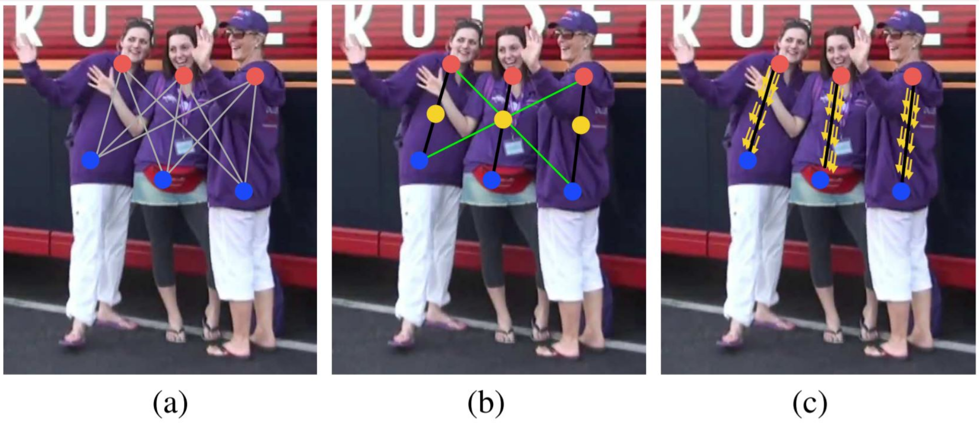


Figure 5.7: Part association strategies. (a) The body part detection candidates for two body part types and all connection candidates (gray lines). (b) The connection results using the midpoint representation: correct connections (black lines) and incorrect connections (green lines) that also satisfy the incidence constraint. (c) The results using PAFs [8].

Non-maximum suppression is performed on the confidence maps to obtain a discrete set of candidate part locations. Due to false positives or the presence of multiple people in the image, it is likely to have several candidates for each part. Every candidate limb is scored comparing its connections with other body parts with the ground truth PAF, then an optimum is found using a greedy relaxation method. They reduce the problem to a maximum weight bipartite graph matching as shown in Figure 5.7. In this case, nodes are the body parts candidates and the edges the possible connections between pairs. A matching in bipartite graph is a subset of edges chosen in such a way that no two edges have a node in common. The goal is to find a matching with maximum weight. The same approach is used when it comes to finding the full body pose of multiple people. First, a minimal number of edges is chosen to obtain a spanning tree skeleton of human pose. Second, the matching problem is is decomposed in bipartite matching subproblems to determine the matching between adjacent tree nodes independently.

5.3.4 Results

All the head pose algorithms have been tested on a GeForce GTX TITAN X GPU, using the real-time video stream from the frontal-upper camera of the robot. This GPU is the same that is going to be employed during the experiments with *Pepper*. *Deepgaze* and *OpenFace* did not fulfill the requirements for the experiment because none of them could work accurately until 8*m* from the robot, yet providing precise results for shorter distances. In particular, *Deepgaze* could not detect faces beyond 3.5*m* while *OpenFace* not even beyond 2.5*m*. On the contrary, *OpenPose* could ensure both real-time performance and an extended range of perception, even beyond 8*m*. One of the most useful characteristics of *OpenPose* is that it provides key-points for the main body and head features. In particular, the key-points of ears, eyes and nose (16, 14, 0, 15 and 17 in Figure 5.6) are going to be used in this experiment to approximate gaze direction. For instance, if *Pepper*'s camera can see all of them in a user, we consider the user is looking at the robot. Unfortunately, the opposite is not always true: if one of the ears key-points (16 and 17 in Figure 5.6) is not visible, a user could still be looking at the robot using eye movements. However, this eventuality was not very frequent while testing the algorithm, because users would need to intentionally look at the robot without moving their head to make it happen. Therefore, if a person reacts spontaneously to the robot's behaviors and looks at it, we expect to see all the five key-points of the face through *OpenPose*. Moreover, this algorithm also provides the whole body pose of a person, which can be used for future work on engagement with *Pepper*. Hence, *OpenPose* has been chosen to extend *Pepper*'s functionalities in the final experiment.

5.4 Distance estimation

Due to the limited accuracy of the unsupervised monocular depth estimation algorithm described in Section 5.2.1, an alternative method to measure the distance of people from the robot is necessary. To this purpose, the work of Taha and Jizat [60] has been analyzed, as it presents a method for collision avoidance using monocular vision. They estimate the distance of obstacles appearing on a single image finding the correspondent vertical and horizontal angles of each pixel. The estimation can be done only knowing the resolution, the height of the camera, its field of view and its tilt angle. All these value are accessible in *Pepper* and it is interesting to notice that the tilt angle of the camera is considered in the computation. In fact, *Pepper* will be moving its head while making animations to draw user's attention, so the height and tilt angle of its camera need to be updated in real-time. Nevertheless, the authors claim that the method proposed is really sensitive to ambient lighting, which may cause bright spots in the image to be mistaken for obstacles. In our scenario this is very likely to happen because of the bright lighting of the experiment setting. Therefore, this method has not been further investigated for implementation on *Pepper*.

To overcome this limitation we decided to use Pepper's front sonar, which has a detection range from 0.3*m* to 5*m* (every distance shorter than 0.3*m* is approximated to 0.3*m*). The measurements of this sensor can be accessed in real-time to provide an approximation of where an obstacle is. As explained in Chapter 3, the sonar has a field of detection limited within 60°, so it is not able to detect obstacles (i.e. people) coming from the side of the robot. However, as in this experiment people are approaching the robot from the front, this limitation will not affect the results.

5.4.1 Results

While testing the performance of *Pepper*'s front sonar, we noticed that it can only detect the distance of the closest object to the robot, hence providing only a single measurement. This prevents the estimation of multiple distances in case several people are in front of the robot, which is why a depth map would be preferable. Moreover, the distance estimated by the sonar is not associable to any object in the field of view of *Pepper*, and the sonar does not distinguish between people and any other obstacle. However, considering that *Pepper* will ideally draw the attention of only one person at a time and that no object (i.e. a potential obstacle for the sonar) will be around it during the experiment, the measurement of the sensor will indicate the distance of the closest person to the robot. The measurements from the sonar have been tested to indicate correctly the real distance of the person from the robot and can be accessed in real-time. Therefore, if only one person is in front of the robot, it is possible to determine its distance from it. In spite of its limitations in the detection range and when dealing with multiple users, this approach has been chosen for distance estimation because of its simplicity and real-time accurate.

5.5 Face tracking

Face tracking is a fundamental requirement to draw attention and elicit initial engagement [7]. However, *Pepper*'s built-in face tracking system, tested during the pilot experiment of Chapter 4, has been found to have limited perception range and robustness to lighting. Due to time constraints, it was not possible to examine the current state-of.the-art techniques for face tracking and choose the best fit for our requirements. Nevertheless, we implemented a basic face tracking method that exploits information from *OpenPose* and *YOLOv3* to extract the position of a person. Comparing the variation of positions in subsequent frames it is possible to compute the head yaw angle that *Pepper* needs to turn to keep track of the person (i.e. to keep it in the center of the field of view). The new face tracking method has been tested on the GeForce GTX TITAN X GPU of the final experiment. With respect to the correspondent on-board functionality, our approach turned out to be robust to lighting and have a perception range larger than $8m$. In fact, in spite of the bright light of the experiment setting, *Pepper* was able to track faces even before participants entered the building.

5.6 Tactile stimuli

In order to exploit the full potential of *Pepper*, also its tactile sensors described in Chapter 3 have been used to implement specific features. In particular, during some tests with the robot at the entrance of the building, we observed that several people touched its head and hands, or tried to do a handshake. Considering that this behavior is not common during a human-human interaction and that we want the robot to be perceived as natural as possible in its manners, some reactions to these events have been developed. In particular, we want *Pepper* to simulate how a human would behave in this situation, e.g. complain and ask not to be touched again. *Pepper* has three tactile sensors on its head, which trigger an event whenever they are touched. To warn the participants who touch the head, we introduced the following behavior for the robot: the LEDs in its eyes blink and the robot politely asks not to be touched (”*Don’t touch my head please*”). We implemented similar behaviors for the tactile sensors in the hands and in the robot bumpers, which make the shoulder LEDs blink while the robot asks not to be touched (”*Don’t touch my hands please*” and ”*Don’t kick me please*”).

5.7 Conclusion

This chapter illustrated the main algorithms and techniques that have been implemented on *Pepper* to extend its functionalities. In this process three main requirements have been followed: real-time data processing, detection of engagement features and extension of the perception range of the robot. These choices were imposed by the necessity of responding in real-time to people and accurately detect relevant social cues when they encountered *Pepper*. Thanks to *YOLOv3* and *OpenPose*, the robot is now able to autonomously detect people and extract their body pose in real-time, even before they enter the building. Both these algorithms are not affected by the lighting conditions of the experiment setting, which represents a considerable improvement with respect to the performance of on-board functionalities. *OpenPose* is used to extract users’ head pose (by tracking ears, eyes and nose key-points), because it approximates well gaze direction (i.e. if all the key-points are visible the person is looking at the robot).

We attempted to keep track of the number of users entering the building using *YOLOv3*. A person was counted as soon as it appeared in the field of view of the robot, then its bounding box was compared to the one of the subsequent frame. If the boxes were similar within a certain threshold, the person was considered the same, otherwise it was counted as a new user. However, the autonomous count of people entering the building is left for future work, because of inaccuracies in determining if a person was entering or exiting the building. In fact, bounding boxes from *YOLOv3* are also used to measure the direction of motion of a person, checking their displacement in subsequent frames. While this approach correctly identifies whether a user is going left or right, it is not always accurate in determining if it is entering or exiting the building. This is due to the variations in width and height of the bounding boxes in different frames, which are not constantly increasing or decreasing if a person approaches or leaves, due to variations in their body pose. Hence, also the refinement of this measurement is left for future work.

Proxemic behavior is another social cue that we tried to record autonomously. Unfortunately, the Unsupervised Monocular Depth Estimation did not provide accurate results, especially in the short range. Moreover, the method proposed by Taha and Jizat [60] for distance estimation with a tilting camera is too sensitive to ambient lighting to successfully work in the experiment setting. Therefore, proximity is recorded using *Pepper*’s front sonar, which can only detect the distance of the closest obstacle to the robot. With this limitation it will not be possible to reconstruct the trajectory of users in front of the robot, to see if they changed or slowed down their walking path after the display of a behavior. Nevertheless, *Pepper* will be able to record in real-time when a user approached it. This information, together with data on gaze direction and from the observation sheets, will still be enough to determine whether users' attention has been drawn, which is why a better recording of proxemic behavior is left for future development. The same happened for users' speed of motion, as these two social cues are related. In fact, the lack of precise information on proximity, combined with the difficulties in correctly estimating the direction of motion, prevented to estimate this measure correctly. On the other hand, the face tracking method developed for *Pepper*, in spite of its simplicity, allows the robot to successfully track participant's faces as they pass by.

Participants' reactions to the behaviors are not going to be autonomously recorded by *Pepper*. The main reason for this choice is that there are several ways in which a person could react, such as waving, replying to the robot's greetings, taking a picture etc. Moreover, the same reaction can be displayed differently by distinct users: a waving gesture could be more or less visible, or a verbal utterance more or less audible by the robot. For these reasons, this information is going to be noted down on the observation sheets during the experiment. However, the robot is going to collect other data on the behaviors of passersby and store them in a database, as explained in Section 7.3. In spite of the limitations and challenges encountered, we managed to successfully extend some crucial functionalities that *Pepper* needs to draw attention and to detect relevant social cues in participants.

Chapter 6

Behavior design for a robot in an entrance hall

It is important to consider *where* the robot is going to be positioned and *which* behaviors it is going to display to draw people's attention. The location of the robot needs to comply with requirements of light robustness and detection range, also allowing the robot to be clearly visible to participants and providing enough space for them to possibly approach. Behavior design depends on the engagement features we want to detect. This chapter analyzes these aspects and provides a motivation of the design choices made.

6.1 Location

The experiment is carried out at the rear entrance of the VMB building in TU Delft campus. As explained in section 1.3, the choice of an entrance hall is motivated by the influence that context has on user expectations. The underlying assumption is that the attention of people is easier to be drawn when they are entering a building, because they are entering a different context and are expecting to make new encounters or even short interactions (e.g. greeting people, asking for information etc.).

Figure 6.1 shows the possible locations of the robot in the entrance hall: close to the wall in the corner between the sliding doors on the right and the emergency door (BR); close to the wall in front of the main doors and below the TV screen (BC); in the right top corner between the entrance and the sliding doors on the right (TR). *Pepper* has been tested in these locations to make sure that users could approach it comfortably (i.e. without being interrupted by other users passing by). Also the light conditions have been taken into consideration: the main entrance doors are made of glass and make the main hall very bright, so it was necessary to check whether the off board functionalities implemented on *Pepper* were robust enough to meet the requirements for the experiment. From preliminary observations we estimated that more than 90% of participants per day go through the right sliding doors (i.e. left sliding doors for a visitor entering the building), because they lead to the cafeteria and to the elevators while those on the left lead to a study room.

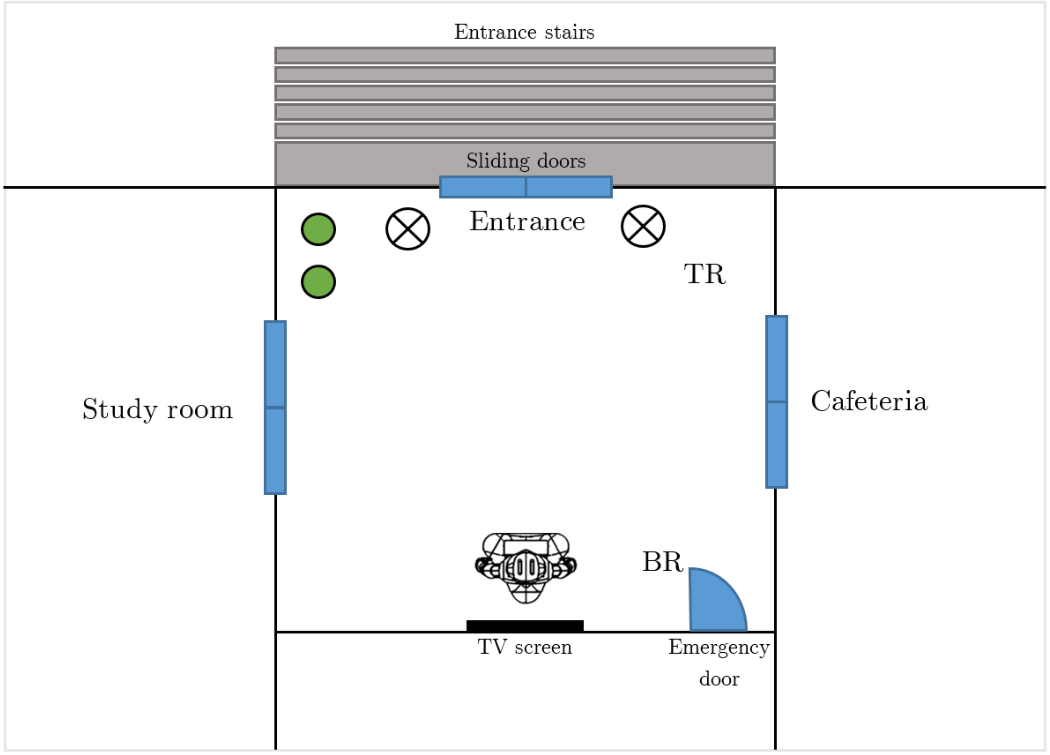


Figure 6.1: Top view model of the entrance hall with *Pepper* in location BC.

In the locations illustrated in Figure 6.1 the robot is clearly visible to participants and has space for comfortable interactions in its proximity. Location BR was discarded because the emergency door behind the robot is frequently used. Users attracted by *Pepper* in this location, could be distracted or discouraged to approach by the passage of other people going through that door. Location TR is closer to the entrance in the corner, so there is enough space around it to avoid these disturbances. Initially, when participants' head pose was estimated with *DeepGaze*, this location allowed to see users within the accuracy boundaries of the algorithm (i.e. 3.5 m from the robot). However, after the implementation of *OpenPose*, which has a broader perception range, also location BC was taken into consideration. It has the advantage of being central, so that *Pepper* can attract people going both left and right after entering the main hall. From this location the robot is able to detect people and their faces, it is far from other passage doors so it allows a comfortable encounter and is clearly visible to the experimenter (who is going to observe from the study room behind the left sliding doors). For these reasons, *Pepper* is going to be placed in location BC during the experiment, as illustrated in Figure 6.2.

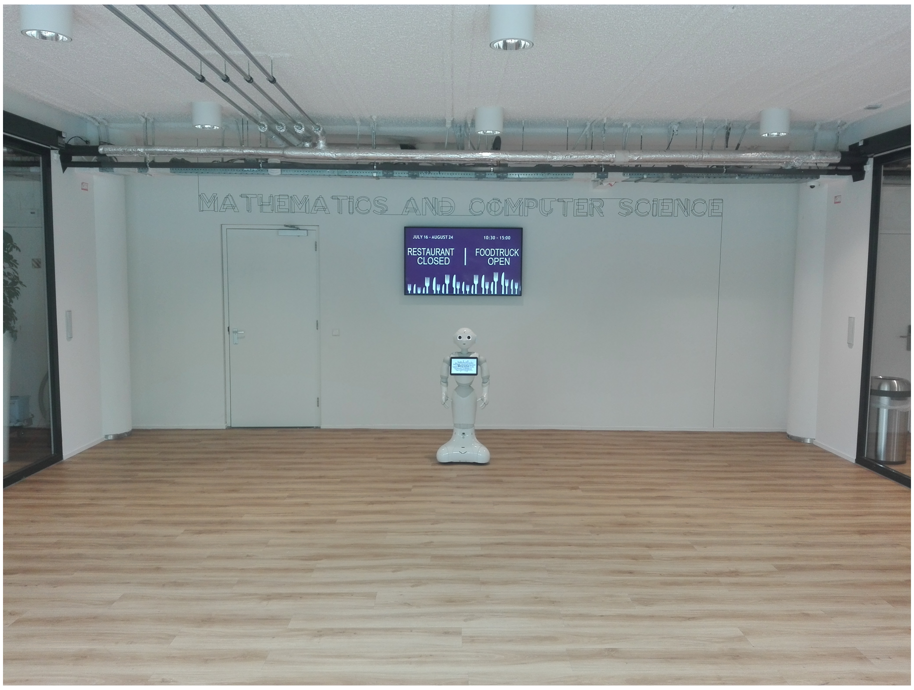


Figure 6.2: *Pepper*'s location during the experiment.

6.2 Behaviors

People entering the building will be welcomed by *Pepper*, who will display one of three behaviors designed to draw their attention. Specific social cues have been selected to achieve this purpose. In particular, face tracking is going to be present in all the behaviors for two main design reasons. First of all, because previous research studies claim it is fundamental for initial engagement and makes social robots look more aware and present [7] [46] [57]. Second, because it is functional to monitor users' gaze while passing in front of the robot. In fact, if *Pepper* did not turn its head toward users when they passed by, it would loose track of them as soon as they moved to the side of the entrance hall (i.e. toward the left or right sliding doors, outside of the field of view). However, with face tracking it is possible to observe their reactions longer and check if they keep looking at the robot or ignore it. Other social cues are going to be combined with face tracking, and the designed behaviors are going to be executed randomly. In order to further strengthen the attention drawing techniques, in all three conditions *Pepper*'s LEDs on eyes and shoulders are going to be flashing when a behavior is displayed.

The three behaviors that *Pepper* displays are triggered as soon as a participant enters through the main doors. This timing is achieved by monitoring the height of a person's bounding box in *YOLOv3*: if the bounding box exceeds the height of 140 pixels the behavior is displayed. The value of 140 pixels has been chosen after conducting a preliminary observation with *Pepper* in location BC, in which the heights of participants' bounding boxes have been recorded as they passed through the main doors. Results showed that in the moment right before entering the building participants had an average height of 135 pixels, while immediately after entering the entrance hall this value exceeded 140 pixels. Hence, the value of 140 pixels has been chosen as reference threshold to trigger *Pepper*'s behaviors as soon as a person steps into the building. The main disadvantage of this method is that it depends on the height of participants. If a person is taller than $1.95m$ and, after climbing the entrance steps, stops outside the building in front of the main doors (maybe to check his/her phone or to smoke) its bounding box will be taller than 140 pixels and the behavior will be triggered. However, such an event hardly ever occurred, therefore does not represent a significant limitation for this approach. In fact, it has been observed only four times out of 26 days of observations with on average 50 participants per day, i.e. in only 0.3% of all cases. Moreover, these episodes occurred in the days of the pilot experiment, so they will not be considered during data analysis.

We briefly discuss why we did not use two alternatives for triggering behaviors. One alternative would have been applying a simple proximity condition: knowing the distance between the robot and the doors, if a person is closer than that distance then the behaviors is triggered. This method would have also avoided any limitation related to the height of participants, but did not represent a viable option due to the reduced perception range for distance estimation (up to $5m$, as explained in Section 5.4). In fact, if the sonars were used to trigger a behavior, this would happen when the target person has already taken a few steps inside the hall, as *Pepper* is more than $5m$ away from the entrance. This will further reduce the available time for behavior display and might increase the risk that *Pepper* is not noticed. Another alternative would have been to use external sensors to detect when a person is passing through the main doors. However, in this project we aim at expanding *Pepper*'s functionalities only using on-board sensing capabilities, therefore such a solution has not been implemented.

Users entering the building spend different times walking from the main entrance to the sliding doors either on the left or on the right. A person that is in a hurry and walks really fast can cover that distance in $4s$, but someone that is looking at his/her phone and walks slowly might take up to $8s$. It has been observed that the average time that a user spends in the entrance hall is around $6s$, so we will take this value as reference to design *Pepper*'s behaviors. This detail is important because it sets a crucial requirement for the duration of a behavior, which has to begin and end within this time lapse. If it takes too long to be displayed, users will not be able to see it completely (i.e. they will already be beyond the sliding doors) and their reaction will not be recorded properly. Hence, behavior duration has been taken into account during the design process. The behaviors implemented on *Pepper* are going to be described in the following sections, together with a motivation that justifies their design.

6.2.1 Gestures

Previous HRI research has shown the effectiveness of gestures in eliciting initial engagement [57] [62]. In particular, waving is known to be a behavior that people frequently use to draw attention or to initiate an interaction [55], hence also *Pepper* needs to exploit such a technique. The behavior designed after this analysis consists in making engagement gestures to the people entering the building. Initially, *Pepper* is in its standard position, standing upright as shown in Figure 6.2. When a person that satisfies the bounding box condition is detected, the robot is going to flash its LEDs on eyes and shoulders and wave with the right hand, as shown in Figure 6.3. The robot then returns to standard position. This gesture has been chosen among the animations already available in the NAOqi framework, its full name is *animations/Stand/Gestures/Hey_1*.

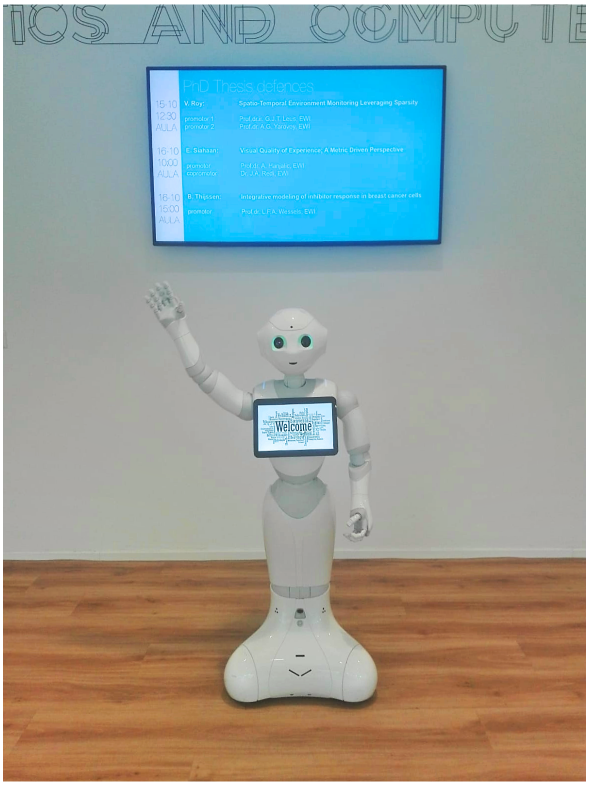

ICS AND COMPUTE
PhD Thesis defences
15-10 12:30 AULA
V. Roy:
Spatio-Temporal Environment Monitoring Leveraging Sparsity
promotor 1
Prof.dr.ir. G.J.T. Leus, EWI
promotor 2
Prof.dr. A.G. Yarovoy, EWI
16-10 10:00 AULA
E. Siahaan:
Visual Quality of Experience; A Metric Driven Perspective
promotor
Prof.dr. A. Hanjalic, EWI
copromotor
Dr. J.A. Redi, EWI
16-10 15:00 AULA
B. Thijssen:
Integrative modeling of inhibitor response in breast cancer cells
promotor
Prof.dr. L.F.A. Wessels, EWI
Welcome

Figure 6.3: *Pepper*'s waving gesture.

Other animations of the framework could have been used as engagement gestures. *Pepper* could have waved with both hands (*animations/Stand/Gestures/Hey_2*) or raised the right hand as if it was trying to do a ”high five” (*animations/Stand/Gestures/ShowSky_5*), but we decided to focus only on the one that was more likely to be successful according to the findings of previous research. Originally, we wanted *Pepper* to combine two gestures: waving at first, then showing the way to the right sliding door. However, the combination of these animations takes 7.5*s* for the whole behavior to complete (3.5*s* per animation plus the transition time from one to the other), which is too long with respect to the average time a participant spends in the entrance hall. Even speeding up the animations would not reduce the duration to less than 6s and will make the movements too unnatural. Leaving both gestures regardless of the overall duration of the behavior would probably bias the results, as some participant could be exposed to two different behaviors: those who walk fast will only see waving, while the others will see also the second gesture or at least part of it. In fact, it has been observed that most of the users were already beyond the sliding doors when *Pepper* showed the way to the right. Moreover, it was noticed that if a participant reacted to the behavior by either waving back or saying something, this reaction occurred in response to waving, before the second animation started. This probably happened because participants did not expect any other gesture after the first one. Therefore, displaying only waving was considered the best option to obtain more reliable results.

6.2.2 Gestures and speech

In literature, the combination of gestures and utterances has been proven to play a major role in initial engagement [29]. One of the reasons that make this technique successful is the combination of both audible and visible cues, which is more likely to attract the attention of users [62]. This behavior builds on the one previously described, combining the waving gesture with a verbal greeting. Also in this case, the robot is in its standard position, standing upright as shown in Figure 6.2. When a person that satisfies the bounding box condition is detected, the robot is going to flash its LEDs on eyes and shoulders and wave with the right hand while pronouncing a greeting. The robot then returns to standard position.

The speaking volume of the robot was chosen during a preliminary test with 12 participants, who were asked to assess how clearly they could hear *Pepper* say ”Good morning” as they entered through the main doors. Participants stood inside the building in front of the entrance and listened to *Pepper* repeating its greeting at 5 different volumes presented in a random order. The volume of 50% was unanimously classified as too low, while those of 80%, 90% and 100% as too loud, as the voice of the robot was echoing in the entrance hall. The volumes of 60% and 70% were both considered as acceptable by users, with 4 preferring the lowest and 8 expressing their preference for the highest. Taking also into consideration that people might be chatting when entering the building (if they are in groups) and that some background noise could be present (e.g. wind noise from outside, chatting noise from the cafeteria etc.) the threshold of 70% has been selected as most suitable.

The following utterances have been tested for this behavior: ”Good morning!” or ”Good afternoon!” (depending on the day time),”Good morning, how are you?”, ”Hi, welcome to building 28!” and ”Hello, have a nice day!”. The goal of phrase testing was to identify which greeting could attract the attention of participants most effectively. During one day of observations *Pepper* only performed the speech and gesture behavior whenever a participant passed by, alternating among the different greetings. The reactions of the 63 participants have been recorded and summarized in the graph presented in Figure 6.4. The utterances that turned out to have the lowest success were ”Hello, have a nice day!” and ”Hi, welcome to building 28!", as they were ignored by most of the users. Speculating on these results, it is possible that the former greeting did not attract attention because in the moment it engages the person (i.e. "Hello,...") it immediately disengages (i.e. "...,have a nice day!"). As for the latter, it may be uncommon or unexpected for a real scenario, especially if most of the users are people who come to the building every day. Moreover, the behavior with this sentence needed 6s to be completed, and many times users were already about to leave the entrance hall when *Pepper* finished it. The simple "Good morning" drew the attention of more users, especially if followed by a question as in "Good morning, how are you?". The latter phrase also elicited stronger reactions from users: 6 of them not only looked at the robot when greeted, but also smiled, replied or thanked the robot for asking. This may be due to the presence of a question, which might engage people more. This last greeting is also significantly more effective in drawing attention than "Hi, welcome to building 28!" ($t(30) = 3.423, p = 0.0021$) and "Hello, have a nice day!" ($t(31) = 4.034, p = 0.0004$).

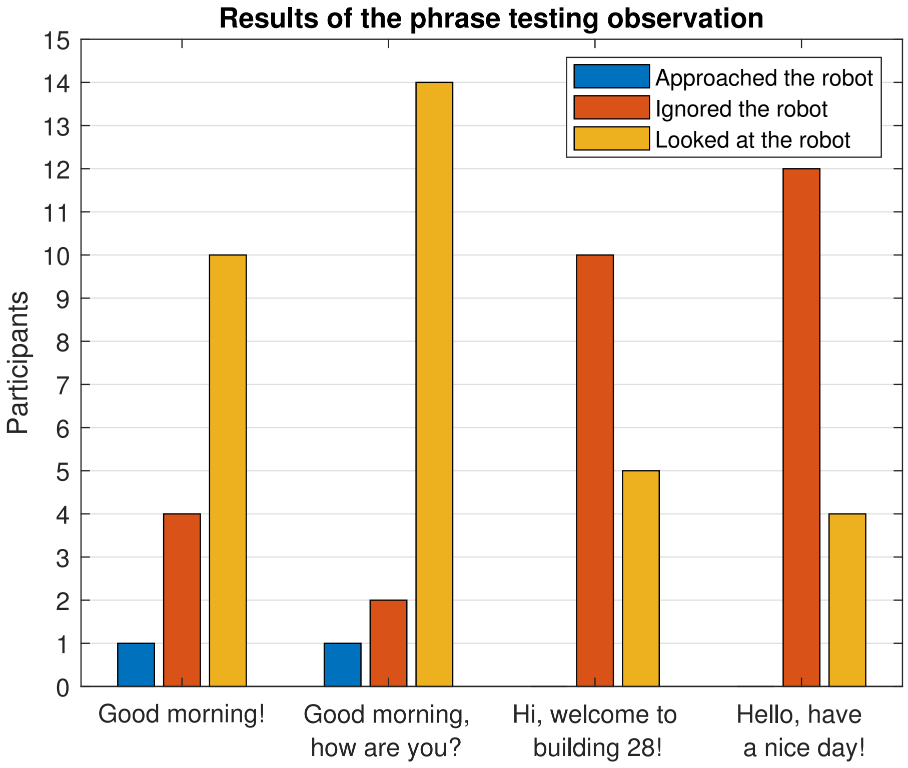


Figure 6.4: Graph illustrating the results from the phrase testing observation.

6.2.3 Movement and gestures

Previous HRI research has investigated the role of proximity in eliciting initial engagement [41] [4]. The experiment of Satake *et al.* [53], where a robot approached participants according to the social cues they displayed, showed that getting closer to target users can successfully draw their attention. In our experiment we are not going to actually approach users, but simply make *Pepper* move forward to investigate if combining movement with waving can attract participants' attention better. Therefore, speech is not going to be used for this behavior. Also in this case, the robot is in its standard position, standing upright as shown in Figure 6.2. When the behavior is triggered, the robot will move forward, then wave with the right hand, return to standard position and finally move backward.

Originally, the behavior was tested with *Pepper* moving forward (and backward) $0.5m$. However, this behavior was taking $7.5s$ to complete, and when the robot started waving users were about to leave the entrance hall, often missing the gesture. The most straightforward solution to this problem would be to make the robot wave while moving forward. Nevertheless, this is not possible when using *Choregraphe*: once in motion the robot has to be ready to perform any safety action to ensure the non collision with the environment (such as look with the head or stop to re-plan a new path), thus no animation can be run during navigation. Therefore, there was no other choice than reducing the distance covered during motion. Eventually, the value of $0.3m$ was identified as optimal, because allows people to see both the motion and the gesture of *Pepper*, overall taking $5.5s$.

This behavior is the only one that is going to be triggered manually by the experimenter, because it was the last to be implemented and still needs to be tested throughout the experiment. The behavior is not available in the experiment GUI and will not be recorded automatically. In fact, the experimenter is going to deploy this behavior through *Choregraphe* and pause the experiment GUI in this time lapse, otherwise the other behaviors could be automatically triggered and overlap with it. However, the database will still record people entering and exiting, as well as if they looked at the robot or approached it. Therefore, it will still be possible to retrieve information about this behavior once analyzed the data collected.

6.2.4 Tactile warnings

If a participant approaches *Pepper* it may happen that it touches its head or hands (or even hit its bumpers in the foot). In response to such tactile stimuli, *Pepper* complains asking not to be touched again, accompanying speech with contextual gestures and flashing its LEDs. If the user touches the robot again after the same behavior is triggered, as explained in Section 5.6.

6.2.5 Distance warnings

This behavior is triggered when a participant approaches *Pepper* and stops standing closer than $0.5m$ to its frontal sonar. When this happens, the robot asks to keep distance accompanying speech with contextual gestures: ”Hi, can you please step back so I can see you better? Thank you!”. The behavior is repeated if the user approaches again.

6.2.6 Wizarding

Other behaviors can be deployed on the robot through the experiment GUI created for behavior control. However, this tool is only meant to remotely handle those situations which would require the intervention of the experimenter. For instance, if a person stops in front of the robot making a phone call, the experimenter might make the robot ask to move somewhere else to free its field of view. In another scenario, some users might insist for a long time trying to interact with *Pepper*. Then, the experimenter might trigger an apologetic behavior that makes the robot say ”I’m sorry, I cannot interact yet” to make them disengage.

6.3 Conclusion

This chapter analyzed the settings of the experiment and the design of the behaviors to be implemented on *Pepper*. Among the three possible locations identified in the entrance hall of the building, the central one has been chosen as most suitable (Figure 6.2). From this place *Pepper* will be able to attract the attention of people going both to the right and to the left sliding doors, and will see them before they enter the building. The off-board functionalities that extend *Pepper*’s on-board capabilities can perform within the requirements in this location.

Three main behaviors have been selected to be deployed on the robot and have been motivated with findings from previous studies. All of them are going to display face tracking, because it is both fundamental for initial engagement and functional to track participant’s gaze. Therefore, it is necessary that off-board functionalities allow to detect relevant social cues such as gaze. On top of face tracking, *Pepper* is going to show (1) waving, (2) speech with waving and (3) movement with waving. Thanks to this approach the purely gestural behavior can be used as term of comparison for the other two, to investigate if combining different social cues represents an effective strategy and understand which of them provides the major contribution in drawing participants’ attention.

These behaviors have been gradually refined to achieve the highest attention drawing effect, choosing the waving gesture and the verbal greeting among a set of different options and adjusting *Pepper*’s movements to be well timed when participants enter the building. The experimenter will trigger the movement and gesture behavior remotely through *Choregraphe*, because it is going to be tested for the first time during the experiment and is not yet automated on the experiment GUI. Other minor behaviors have been developed to ask participants not to touch the robot or to stay farther away if they get too close.

Chapter 7

Experiment

A user study will be performed to assess how people's behavior is affected by *Pepper*'s drawing attention techniques. In order to guarantee the validity of the experiments, the setup needs to be defined and followed in every trial, including those situations that might interrupt or alter them. For this reason, a pilot experiment has been conducted to identify the major aspects of the experimental design that need to be refined before the experiment starts. This chapter describes the *Objectives*, the *Procedure* and the *Data* of the experiment.

7.1 Objectives

The main purpose of the pilot is to test the experimental setup and behavior design, in order to make the necessary refinements for the experiment. It will also be used to improve the observation sheet and identify appropriate metrics to measure initial user engagement and the novelty effect. The objective of the experiment is to investigate which of the behaviors designed for *Pepper* can draw attention more, analyzing the spontaneous reaction of users to the robot. Therefore, participants are not introduced to the nature of the trials to avoid biasing the results. This study is going to provide an evaluation of *Pepper*'s drawing attention techniques and test the robot's off board functionalities in a typical experimental set up. Moreover, observation sheets are going to be filled in to record users' reactions to *Pepper* and compare the results with those obtained from MongoDB. These are going to be used to evaluate initial user engagement and novelty effect.

7.1.1 Initial user engagement

Initial user engagement is evaluated by measuring the influence of the verbal and nonverbal behaviors of *Pepper* on participants. In particular, the robot's drawing attention techniques were assessed analyzing the reactions they induced in people. The objective is to identify which technique is most successful in drawing people's attention depending on participant's behaviors. In this way it is possible to understand if certain social cues are more effective than others in eliciting initial engagement. In this project, we hypothesize that behaviors integrating more social cues (i.e. speech with gesture and movement with gesture) will be more effective in drawing attention (H1). We speculate that participants will be more responsive toward a robot that shows multiple behaviors rather than a singular one. We also hypothesize that the behavior combining speech with gesture is going to be the most effective (H2). Torta *et al.* [62] have shown how auditive cues attract attention more effectively than visual cues. On the basis of these findings, we argue that auditive cues may also have a higher drawing attention potential than proximity cues, and will be more easily noted.

7.1.2 Novelty effect

As discussed in Section 2.3, in HRI studies the novelty effect is observable in long-term interactions and wears off over time, e.g. as participants get used to the new technology. Because of the nature of this phenomenon, doing observations before the final experiment can bring two main advantages. On one side, this strategy allows to study the changes of users' reactions over a longer period of time, starting from the first appearance of *Pepper* in the entrance hall. On the other, it exploits the pilot to wear off the novelty effect, so that data are less affected by this phenomenon when the experiment starts. At this stage, the objective is to understand if users genuinely respond to *Pepper*'s behaviors or if they do it just because it is novel. Also in this case we made some hypotheses, motivated by the results achieved by previous HRI studies on long-term interactions. We hypothesize that there will be a significant drop of participants affected by the novelty effect through the course of the pilot and experiment (H3). Moreover, we also expect this decrease to occur between two and five weeks after the deployment of *Pepper* in the entrance hall (H4). This is suggested by previous studies on novelty effect, even though the context of the experiment or the robot used were different from the present one. The novelty effect is expected to be minimally present during the experiment and the results more reliable.

7.2 Data

The experiment is going to be evaluated using (1) the observation sheets filled in by the experimenter and (2) the data stored in the MongoDB database during the trials. The research of Finke *et al.* [14] has been taken as reference to identify which participants' reactions to observe. Even if their study focuses on using sonars to recognize human movements, they too conducted some observations and classified people's reactions in seven categories as they encountered a service robot. However, as in our case the experimenter is going to fill in the observation sheet manually, we decided to focus only on the reactions most easily identifiable, to avoid possible misinterpretations. Therefore, only four of their categories have been used, organized from lowest to highest interest that people show for the robot:

1. the participant ignores the robot.
2. the participant looks at the robot while walking;
3. the participant stops and looks at the robot;
4. the participant approaches the robot;

These reactions can be generated when *Pepper* displays gestures, speech with gestures or movement with gestures. Therefore, every reaction is going to be subdivided in three categories to keep track of which behavior of the robot triggered the reaction in the participant. Appropriate metrics have to be defined to interpret our data correctly and draw meaningful results. These metrics are hereby presented, subdivided according to the phenomenon they are related to.

7.2.1 Initial user engagement

Initial user engagement manifests through verbal and nonverbal behaviors that users display in response to those of *Pepper*. When a participant ignores the robot and does not show any response to its attention drawing techniques, it means that the strategy used was not effective in engaging the person. On the other hand, if a participant looks at the robot after it displayed a specific behavior, we consider that attention has been successfully drawn thanks to this behavior. A participant may engage in more explicit behaviors which indicate the robot has his or her attention, such as waving or speaking to the robot. For every behavior that the robot displays (which are alternated in a random order) the experimenter records the reaction it induced in participants. Different metrics are going to be presented to evaluate this phenomenon. The metrics collected are going to be averaged through different days of observations, in order to obtain a final comprehensive estimate of which technique was more successful in drawing attention.

As the behaviors are triggered randomly, it may happen that they are not displayed evenly throughout the day. Therefore, considering the percentage of users who looked at the robot (after the display of a specific behavior) among the total number of people who entered the building would not be meaningful. A more accurate metric is the percentage of users who looked at the robot (after the display of a specific behavior) among the total number of people who were displayed the same behavior. This metric can provide a more objective point of view on user engagement. Another insightful metric is represented by the percentage of people (among those who triggered the same behavior) who not only looked at the robot, but also waved or replied to it after the display of a specific behavior. Both these metrics are also going to be analyzed sequentially, to study the reactions of users day by day and investigate the presence of possible learning effects.

In order to understand if a person is looking at the robot from the analysis of data from MongoDB, different conditions must be met. First, the person has to be visible for at least $4s$, which correspond to the shortest time a person would usually spend walking in the entrance hall. If there are less, that can be a person exiting or that was only partially visible when passing by (e.g. if someone enters with a group and becomes visible to the camera only for a short time), so the data related to this person are removed because they may not be reliable. Second, the value of the distance from the robot does not have to become shorter than $2m$, because in such case the person would have approached, and not only looked at *Pepper*. Usually, a person only passing by keeps an average distance that never drops below $4m$, therefore the $2m$ threshold has been chosen to distinguish between ”approaching” and other behaviors. Then, we have to determine if participants' reaction has been triggered by the gestural behavior, by the speech with gesture behavior or by the movement with gesture behavior. Finally, a dedicated variable will measure if a person was ”looking at the robot”.

7.2.2 Novelty effect

When it comes to the analysis of the novelty effect, the metrics chosen should take into consideration the temporal nature of this phenomenon and be able to capture its evolution in time. Therefore, the results obtained from the observations are going to be analyzed through the different days of observations. Moreover, as initial engagement depends on the context of the encounter, the novelty effect related to it does as well. Therefore, it is necessary to define which are the expected reactions from participants according to the context so that we can classify the others as due to the novelty effect. If a person ignores the robot after the display of a specific behavior, we argue that such participant is not affected by novelty because it does not show neither interest nor curiosity for the robot. In the previous section, we classified ”looking at the robot while walking” as indicative of engagement, as well as waving back or replying to the robot. These are the reactions that we assume to be independent from the novelty effect.

On the contrary, all those reactions that are unexpected are considered as influenced by novelty. For instance, taking pictures of the robot is a behavior associable to novelty, as usually it is induced when something new is encountered. We consider the behavior of stopping to look at the robot or approaching it as most representative of the novelty effect. In fact, both these behaviors make participants spend more time in the entrance hall than what they would if they simply passed by. This means that they have been engaged to a point that they interrupt their current task to dedicate more attention to what *Pepper* is doing. This extra attention may be reasonably associated to a novelty factor, as the participants are curious about the robot and spend more time examining it. People may approach the robot to read what is written on its tablet, touch it or try a hand shake, and all of these reactions are considered as caused by the novelty effect. Even these behaviors of users can be partially predicted, which is why the observation sheet has the appropriate sections for their classification. However, for all the other reactions that may occur, the ”Notes” section will be used to record them. Some examples of such behaviors could be: a participant inviting friends to interact with *Pepper*, or a participant triggering *Pepper*'s tactile warnings multiple times on purpose etc. Theoretically, as the novelty effect wears off, the number of users reacting in such a way to *Pepper*'s behaviors is expected to significantly drop over time.

An indicator of novelty effect is represented by those who ”stopped and looked at” or ”approached” the robot in response to a specific behavior among the people exposed to the same behavior. This metric can highlight which of the three attention drawing techniques generates the strongest novelty effect. However, a more global view can be provided considering the percentage of people who ”stopped and looked” or ”approached” the robot among the total users who entered the building, regardless of the behavior that triggered that reaction in them. This last metric can show how novelty as a generalized phenomenon wears off over time. Data from MongoDB can also be used to determine when a person approaches the robot. The same requirements of the ”looking at” behavior must hold, but in this case distance becomes more important. If it drops below $2m$ for more than $3s$, then the person has ”approached the robot”. This threshold has been chosen because some people go through the emergency door behind *Pepper* and may pass very close to it. In this case, their distance may be less than $2m$ from the robot even if their intention is not to approach it but only reach the door. Users who approach the robot usually spend longer time in front of it, so a threshold of $3s$ has been selected.

7.2.3 Summary of metrics

After defining how the data of this project is going to be analyzed, the metrics for initial engagement collected form literature and classified in Table 2.1 need to be updated. In particular, the metrics *Eye contact*, *Gaze direction* and *Face tracking* have been substituted by the more general *Looking at*. In fact, we developed a method to determine whether a person is looking at *Pepper* or not (both for the observations and the MongoDB database) but we are not specifically measuring any of the three metrics individually. Therefore, the table from Chapter 2 is revised as follows.

Metrics	**Measured with *Pepper***	**Measured during observations**
Looking at	Y	Y
Speed of motion	N	N
Proxemic behavior	Partially (sonar)	Partially (people who approached)
Waving	N	Y
Speech and gestures	N	Y

Table 7.1: A summary of the relevant metrics for user engagement that have been considered for the experiment.

7.3 Procedure

We now describe the different steps that are followed in every trial. *Pepper* is positioned in the entrance hall of the building and turned on. The experimenter waits next to the robot until it has completed its initialization phase and makes sure that it is connected to the network before leaving. Once the robot is ready, the experimenter goes to the study room accessible from the left sliding doors of the entrance hall and sits at the workstation that allows to control *Pepper* remotely. The experiment GUI is started and the robot is ”woken up" (i.e. stand upright, because when it turns on it is in a crouched down pose), a welcome message is displayed on the tablet and its people detection and head pose algorithms are started. The camera feed of the robot is visible from the experiment GUI and shows the bounding boxes around people (from *YOLOv3*) and the links that indicate their body pose (from *OpenPose*) as shown in Figure 7.1.

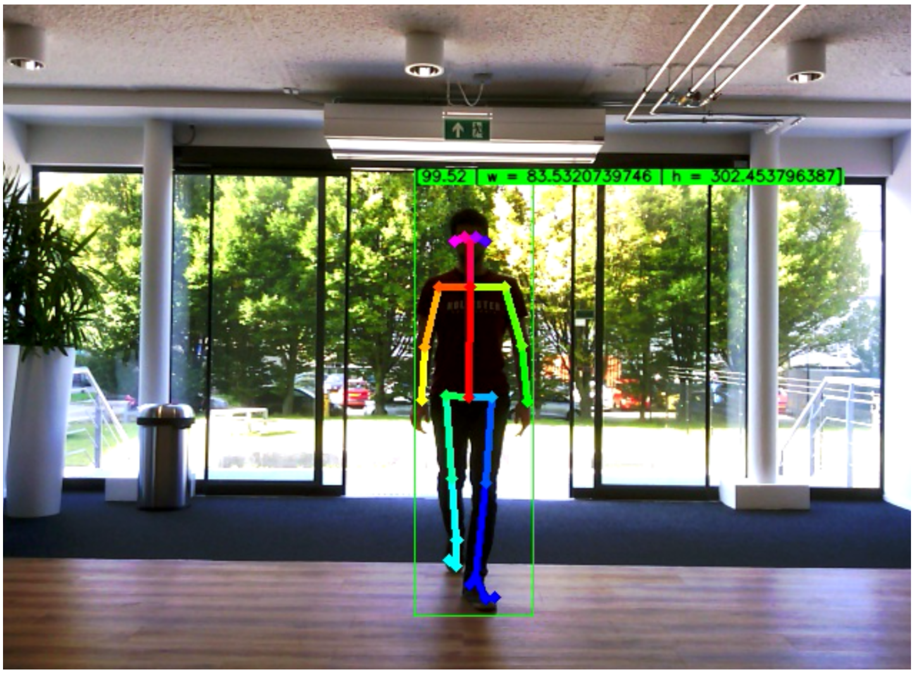


Figure 7.1: Example of people detection during the experiment.

Before the experiment, *Pepper* has been deployed in the entrance hall for a series of preliminary observations, 2 hours every morning for 16 days. In this period, the experimenter investigated how to improve the observation sheet and the behaviors of the robot. Thanks to these trials it was possible to incorporate the most frequent reactions of users (i.e. waving, speaking, touching the robot, taking pictures) and their behaviors (i.e. ignore, look at, stop and look, approach) in the observation sheet, so to simplify the job of the experimenter. The sheet itself went through a series of revisions, which made it more comprehensive of all possible scenarios as the flaws of previous versions emerged. During this period, also the gesture and speech with gesture behaviors have been tested. In fact, the available engagement gestures of the robot have been inspected and eventually "waving" has been chosen as definitive. Also phrase testing was performed in these days, showing that the sentence "Good morning, how are you?" had the highest attention drawing potential among those tested. The movement with gesture behavior was shortly tested only in the last few days of pilot experiment, to find the optimal displacement that could allow passersby to also see the robot waving. Therefore, the behavior has not been included in the experiment GUI neither classified in the MongoDB database due to time constraints. Nevertheless, it has been deployed during the experiment, which has been conducted for 10 days for 2 hours.

The observations took place in the morning, because more participants are expected to enter the building than in the afternoon. During the experiment, a protocol needs to define which participants are going to be considered and what data is going to be recorded. Every time the experiment GUI application is started data is autonomously collected by *Pepper*, which stores it using a MongoDB database [42], a free open source NoSQL database program. The experimenter is going to write its observations on paper observation sheets, provided in Appendix B.

7.3.1 Participants

Other people can be present when a user approaches the robot or can enter the building immediately after, in a sequential order. They do not trigger any of *Pepper*'s behaviors, but they may also see them when they are displayed to another person. These people and their reactions are not going to be recorded, because they are not our focus on one to one interactions. For instance, participants who stay in the entrance hall to have a phone call while other users pass by *Pepper*. Such people are also going to be recorded on the ”Notes” section of the observation sheet with the time they arrived and left the entrance hall, in order to identify them in the MongoDB database. The interpretation of individuals' reactions is simpler because they can be clearly associated to a specific behavior of the robot. As for groups, data analysis is more complicated because participants' reactions can be influenced by those of other group members, which is why they are going to be recored on a separate observation sheet. On the other hand, people who enter more or less at the same time are hard to classify, especially because they might have seen the behavior of the robot only partially. Therefore, in order to simplify our analysis, only individual passersby and multiple passersby in groups who have triggered one of *Pepper*'s behaviors will be considered during the observations.

7.3.2 MongoDB

When participants enter the building *Pepper* is going to store relevant data about them in the MongoDB database. This information is recorded for every frame in which a person appears, in case more people are present at the same time they are going to be assigned a different *personID* but they are going to share the same *frameID*. Among the several features, we store in the database if a person *is_new*, if it *is_looking_at_camera* (i.e. if we can see the 5 head key-points from *OpenPose*) and its *distance* from the robot measured with sonars. When the robot displays a behavior, the boolean variable *is_engaged* is recorded as ”true”, and the database specifies if that behavior was an animation or a verbal greeting (respectively *has_animation* and *has_message* values). If a person approaches and triggers one of the tactile or distance warning of *Pepper*, the boolean variable *is_warned* is set to ”true”. Finally, all the frames are associated with a time stamp that indicates the *last_encounter* with a person.

7.3.3 Observation protocol

During the experiment, the experimenter is going to write observations on three different observation sheets: one for male participants, one for female participants and one for groups of people entering together the building. After noting date and starting time, the experimenter is going to record the number of participants entering the building and their reactions to *Pepper*'s behaviors. The observation process needs to be regulated by a specific protocol that defines how participants have been classified. Their reactions are going to be observed after the robot has displayed one of its behaviors, because in this scenario it is reasonable to associate human reactions with the robot's behaviors. As anticipated, only the gesture and gesture with speech behaviors are going to be automatically displayed by the robot. When both these behaviors have been triggered a set of ten times, the experimenter will pause the experiment GUI and trigger for a set of ten times the movement with gesture behavior. Then, the experiment GUI will be resumed and the automated behaviors will be displayed again for another set of ten times.

After *Pepper* has displayed one of its behaviors, participants will be classified in four categories and recorded the appropriate observation sheet. This procedure is done observing specific reactions they elaborate as a response according to the following protocol:

- Ignored the robot: the participant does not look at the robot after behavior display, i.e. it does not do any movement to orient its head pose toward the robot.

- Looked at the robot while walking: the participant glimpses or keeps gazing toward the robot after behavior display, i.e. it makes a visible head movement to orient its head pose toward the robot. This category also includes people who glance multiple times toward the robot while walking.

- Stopped and looked at the robot: the participant interrupts their walking path to look at the robot, i.e. it orients its head pose toward the robot, then stops and eventually changes also its body pose to orient it toward the robot. After a short pause, the participant resumes its walking path without deviating from the original trajectory.

- Approached the robot: the participant looks toward the robot (i.e. orients its head pose toward it) and deviates from its original trajectory to approach *Pepper* after it displayed one of its behaviors. If a participant immediately aims for the robot without changing its walking trajectory, it is recorded in the "Notes" section as a person who approached spontaneously and not in response to any behavior.

When participants do not ignore the robot it is important to record if they display any other reaction, such as waving, speaking, taking pictures or touching the robot. These are clear indicators of engagement, because participants are not only paying attention to the robot but also trying to respond to its behaviors. This protocol has been followed to collect data during the experiment, and any behavior that was unexpected (i.e different from those described above) has been recorded in the "Notes" section of the observation sheet.

7.4 Conclusion

This chapter presented the objectives and the protocol for the experiment. The experimenter will record whether participants ignored the robot, stopped to look at it, looked at it while walking or approached it. Every one of these reactions is going to be distinguished depending on the behavior of the robot that generated it. Reactions of single users and groups (two or more people) are going to be recorded on different observation sheets. Data is also going to be autonomously collected by *Pepper* and stored in a MongoDB database. Eventually, it will be reorganized into meaningful metrics and compared to those derived from the observation sheets, so it will be possible to validate the accuracy of data collection.

For what concerns initial engagement, the results of the observations are going to be considered both cumulatively and sequentially, to identify the behavior that attracted attention most successfully and check if there was any learning trend of participants throughout the observations. The reaction that is most representative of user engagement is looking at the robot after the display of a specific behavior. If an additional reaction (such as waving or speaking) is displayed, engagement is considered to be even stronger. Observing the number of users who looked at the robot over the participants who triggered the same behavior will provide a comprehensive view over the effectiveness of a specific technique. We hypothesize that the behaviors combining more social cues will be more effective in drawing attention (H1), and that the speech with gesture behavior will be the most engaging, as we assume that auditive cues will elicit a stronger response in participants than proximity cues (H2).

For what concerns the novelty effect, the results of the observations are going to be considered sequentially, to analyze how users' reactions varied through time. In this case, approaching the robot and stopping to look at it are the reactions that are identified as most clearly related to novelty. These values are going to be presented divided per behavior and as a whole, to have also a general idea of the global novelty effect caused by *Pepper*. The experimenter also notes about any other reaction that does not correspond to those defined, which may still be related to novelty. These reactions are going to be used as subjective metrics, their occurrence through the observations period will indicate the strength of the novelty effect. We hypothesize that there will be a significant decrease of this kind of reactions in participant through the course of the pilot and the experiment (H3), and that this drop is going to occur between two and five weeks from the first deployment of *Pepper* in the entrance hall (H4).

Chapter 8

Results and conclusions

8.1 Results

Results are presented for the two main topics of interest of this project. For initial user engagement, only the results of the experiment are presented, because only in this period all three behaviors were tested together. For novelty effect, also the results of people who approached the robot during the pilot are provided. This choice is motivated by the long-term nature of this phenomenon, which can be better analyzed over a longer period of time. Before presenting the results, the data collected in MongoDB need to be cleaned in order to be comparable to those of the observations. The number of participants entering the building was not homogeneous through the 26 days of pilot and experiment. On average, 50 people passed by *Pepper* every day during the hours of observations, with a maximum of 74 participants and a minimum of 33, for this reason the results are presented as percentages. Out of the total participants, 27% were females.

8.1.1 Data cleaning

The MongoDB database records data of all people entering and exiting the building during a day, but our analysis is focused only on those who enter and are welcomed by *Pepper*. However, exiting people are easy to identify in the database because they enter into *Pepper*'s field of view from the side, so they are visible to the robot for much shorter time (depending on the walking speed of the user, this could be between $1s$ and $2s$). On the other hand, people entering are visible longer, because they enter in *Pepper*'s field of view from the front and are tracked as long as they stay in the entrance hall (on average around $6s$). Another issue with people classification may occur when participants approach the robot. In general, when a person is identified as new, it is attributed a new *person_id* and its actions are associated with that specific value. However, if a person gets too close to the robot it is not visible entirely anymore, so *Pepper* will usually see only the torso and sometimes the head. In this scenario, the person is not recognized as the same who was identified in distance, but is counted as a new one. In this case, data can be cleaned looking at the *last_encounter* and *distance* variables, which indicate if people that have been classified as different should actually have the same *person_id*.

Another category of people that can be identified in the data is represented by persons who stay in the entrance hall to have a phone call. These people may hang out in front of the robot for a long time. They can be identified by the large number of frames in which they appear and thanks to the notes taken by the experimenter on the observation sheet. As for groups, *Pepper* records every person individually but from the *last_encounter* variable it is possible to see if they were seen at the same time. Nevertheless, the groups are still hard to identify because some users may have not been visible by the robot, whose vision was probably occluded by the members closer to it. This data cleaning has been performed in Excel for all the 10 MongoDB databases, one for each day of the experiment. Even so, automatically collected data do not always match with the observations due to some technical limitations discussed in Section 8.3. Therefore, as it is not feasible to use the information from all databases, we considered only those that had a difference of 3 participants or less with the corresponding observation sheets. In particular, the results collected on the 10^{th}, 12^{th}, and the 14^{th} of September are going to be presented. The reason of this better correspondence with respect to other databases is due to the fact that few groups approached the robot in these days, and that no participants spent time in entrance hall to have a phone call. These favorable circumstances helped keeping data more readable and reducing the cleaning effort. The data was analyzed on Matlab as explained in Section 7.2, and compared with that of the observation sheets for validation.

8.1.2 Initial user engagement

Initial user engagement was evaluated by measuring the influence of *Pepper*'s behaviors on participants entering the building. In particular, the behavior that has been identified as indicative of initial engagement was ”looking at the robot while walking” (i.e. making a visible head movement to orient the head pose toward the robot), in response to the robot waving, waving and greeting or moving and waving. Figure 8.1 shows the cumulative results obtained from the observations conducted during the 10 days of the experiment.

When *Pepper* only displayed a gestural behavior, it drew the attention of 38.3% of the participants. No significant differences emerged comparing the behaviors of male and female participants, even if females seem slightly less engaged than the average (34.3%) while males slightly more (39.8%). Similar results are obtained for the movement with gesture behavior, which drew the attention of 37.1% of the participants who triggered it. Also in this case, no significant difference was found in the behaviors of male and female participants, even if the gap between their averages is larger (44.8% for females and 34.2% for males). On the other hand, whenever *Pepper* addressed users with a verbal greeting accompanied by a waving gesture it scored the highest attention drawing rate, with 51.6% of participants looking at the robot after being exposed to this behavior. Similar averages have been achieved by male (50.7%) and female (54.2%) participants. According to the data recorded on the observation sheets, the speech with gesture behavior is significantly more effective in drawing attention than the purely gestural behavior ($t(94) = 2.132, p = 0.0385$) and the movement with gesture behavior ($t(107) = 2.262, p = 0.0443$).

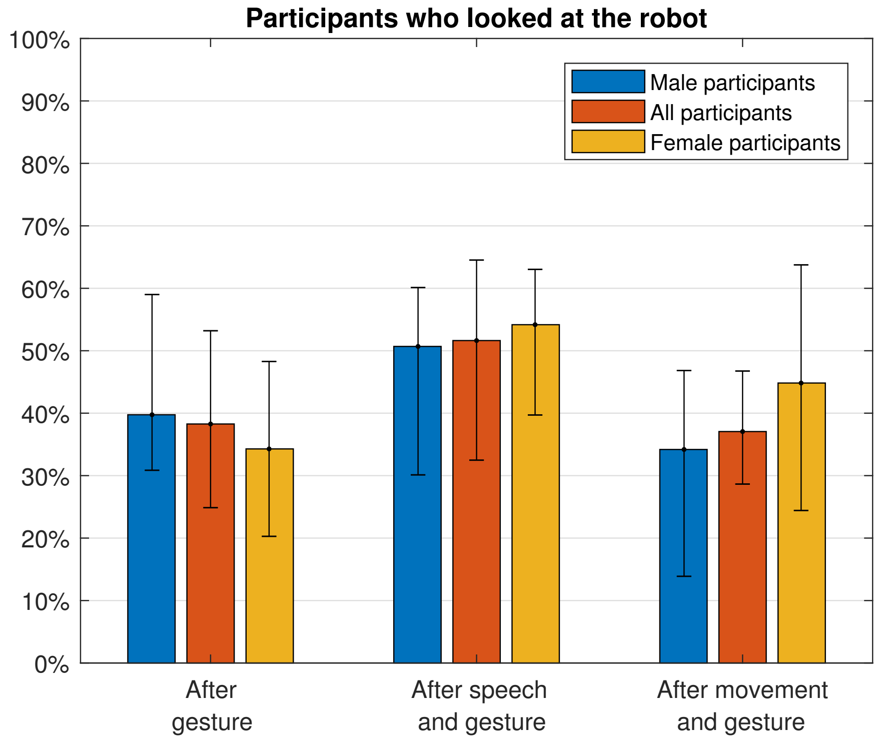


Figure 8.1: Graph showing the average percentage of individual participants who looked at the robot after the display of a specific behavior.

Figure 8.2 presents the behavior of participants throughout the 10 days of experiment, showing the percentages of their response to *Pepper*'s behaviors. The movement with gesture behavior shows a trend that is more or less stable, with no big deviations from the average. On the other hand, the gesture behavior has a larger variance: on day 3 participants looking at the robot were 12% less than the average, while on day 4 they were 16% above it. Variations are even larger for the speech with gesture behavior, which drew the attention of more than 70% of participants on day 3 but around 30% of them on day 7.

Figure 8.3 shows the users who, after looking at the robot, reacted waving or replying to it. Also in this case, the speech with gesture behavior scored the highest result, with 40% of users who either spoke or waved back to *Pepper* (40.6% for males and 38.5% for females). The purely gestural behavior obtained additional reactions from 24.2% of the participants (23.9% for males and 25% for females), while the movement with gesture one from the 27% (28.4% for males and 23.1% for females). In all these cases there is no significant difference between male and female participants, as both reacted after looking at the robot in a similar fashion. The speech with gesture behavior is significantly more likely to elicit an extra reaction from participants compared to the purely gestural behavior

$(t(31) = 2.520, p = 0.0178)$. However, it only shows a tendency to be more engaging than the movement with gesture behavior $(t(32) = 1.754, p = 0.0890)$.

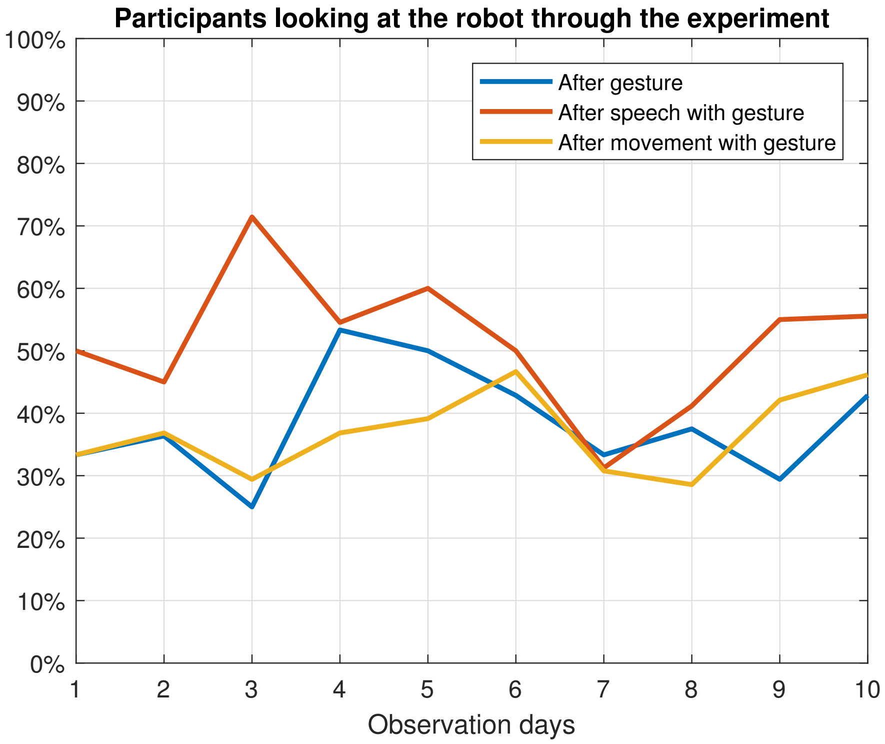


Figure 8.2: Graph showing the daily percentages of individual participants who looked at the robot after the display of a specific behavior.

Figure 8.4 shows how participants reacted after looking at the robot throughout the 10 days of experiment. The speech with gesture behavior shows the most stable trend, as it induced people to wave or reply to the robot during all days of observations. However, it shows a large variance on days 6 and 10, where the percentage of participants reacting is around 20% more than the average. The trends of the other two behaviors are less uniform, also due to the fact that on some days participants simply looked at *Pepper* with no additional reaction. This occurred twice for the movement with gesture behavior (days 8 and 10) and four times for the purely gestural behavior (days 1, 3, 7 and 8). The former elicited more reactions than speech with gesture on days 3 and 7, while the latter did so only on day 5.

The results form the MongoDB databases also reflected the trend highlighted by the observations, showing similar results. Data from the observations have been slightly amended to include also the reactions of participants who were in groups, but their behaviors have been considered individually. This correction was necessary because in the MongoDB database it is not possible to differentiate whether a participant was in a group or not. Integrating data from groups with that of individuals allows to make a more accurate comparison between the two data sets. In Table 8.1, the results from the three databases are compared to those obtained in the observations. Due to the limitations discussed in Section 8.3 the results are not perfectly matching, but confirm that speech with gesture draws attention the most.

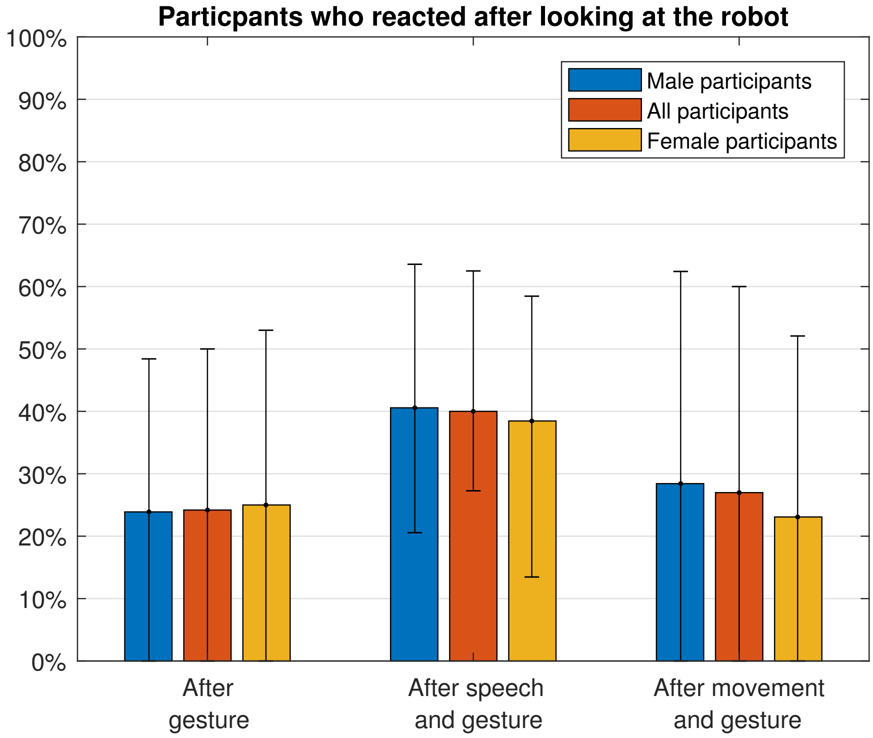


Figure 8.3: Graph showing the average percentage of individual participants who, when looking at the robot, reacted waving or speaking after the display of a specific behavior.

		Participants' behavior		
		Gesture	Speech and gesture	Movement and gesture
10^{th} September	Observations	5	8	6
	MongoDB	5	7	5
12^{th} September	Observations	5	10	5
	MongoDB	4	8	5
14^{th} September	Observations	10	15	9
	MongoDB	7	11	9

Table 8.1: Table comparing the results of observations and MongoDB for people looking at the robot, after the display of the different behaviors.

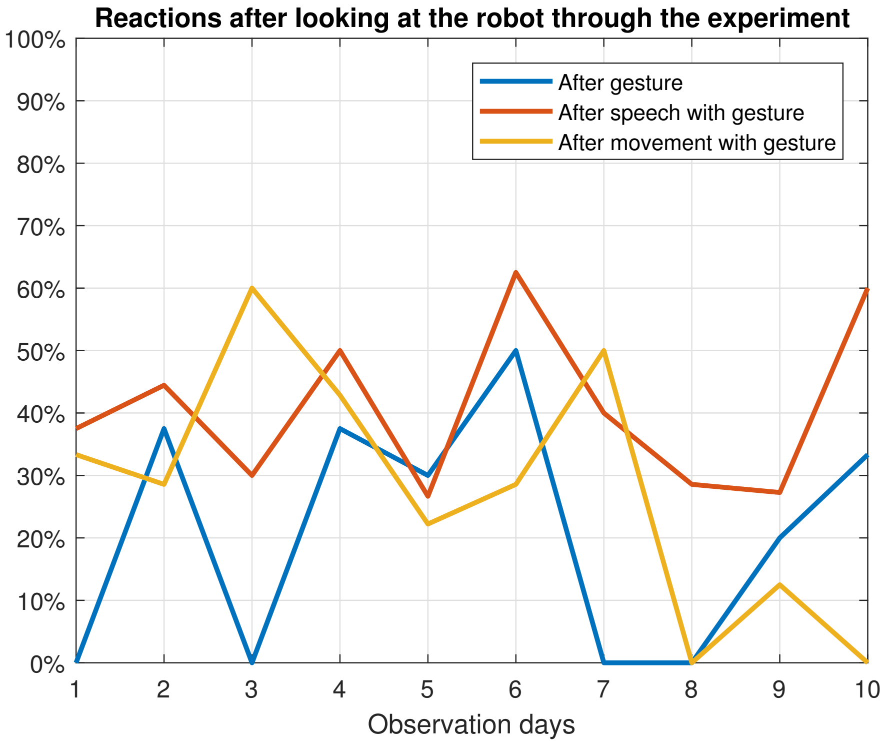


Figure 8.4: Graph showing the daily percentage of individual participants who, when looking at the robot, reacted waving or speaking after the display of a specific behavior.

8.1.3 Novelty effect

The novelty effect has been measured by observing how many people approached *Pepper* or stopped to look at it. Figure 8.5 shows a combination of participants' reactions regardless of the behaviors that triggered them, to provide a global overview of the phenomenon. These data are presented as dots in the graph, while a 2^{nd} degree polynomial function approximates the general trend of novelty.

Novelty appears to have a strong impact on the results in the first days of observations, then progressively wears off in time. The slope of the curve is not constant through the whole period, but becomes gradually more gentle after few days since the beginning of the experiment. In particular, it seems to reach a steady state below the 10% of total participants after three weeks of observations. This result shows that there is a significant difference between the users approaching the robot at the beginning of the observation process and those doing so in the last days ($t(115) = 2.176, p = 0.0334$). Data of male and female participants have been recorded only for the days of the experiment, therefore there are not enough data points to estimate their novelty effect trends separately.

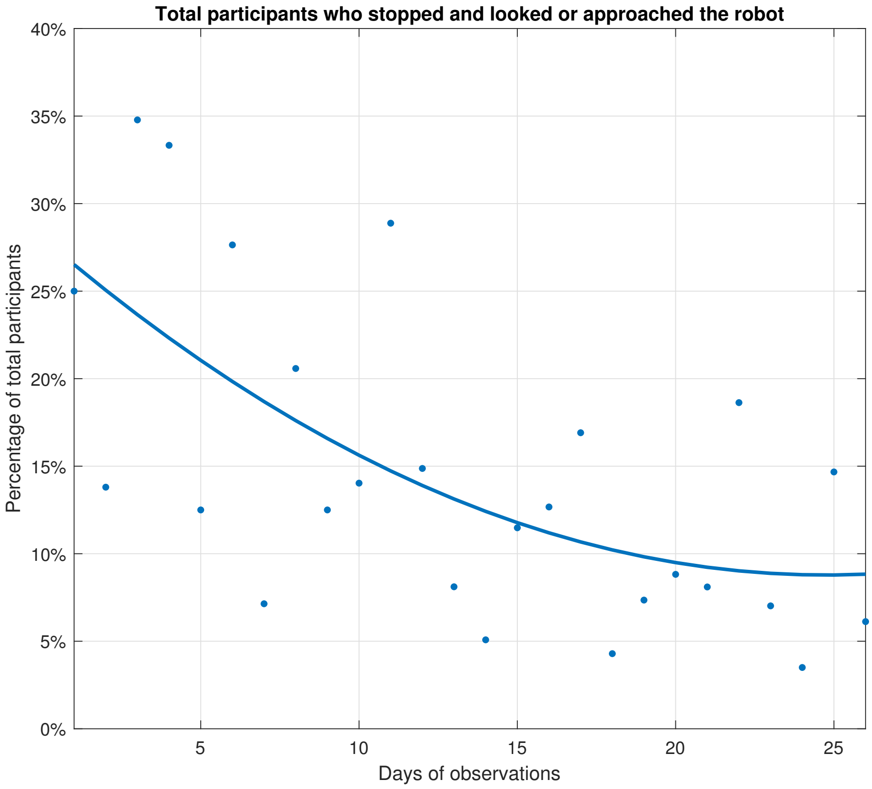


Figure 8.5: Graph showing the trend of the overall novelty effect.

Figure 8.6 shows the percentage of people who approached the robot at different days of observations after the display of a specific behavior. Also in this case, speech with gesture attracted more participants toward the robot. In fact, during some days at the beginning of the observation process, this behavior attracted more than 20% of total participants. Through the whole course of the observations there were some users who approached the robot spontaneously. These users spent more time than others trying to interact with the robot, making an effort to trigger its behaviors multiple times and taking pictures or videos of them. All the behaviors followed a similar trend, except for the movement with gesture behavior. In this case, the data of people who approached (or stopped to look at) the robot have only been collected during the 10 days of experiment, therefore there is not enough information to draw conclusions about its novelty effect as for the other behaviors. Results form the MongoDB databases reflect the trend highlighted by the observations, showing comparable results. Also in this case, the data presented include people who approached in groups or from a group, because the distinction from individuals is hard to conduct on the current MongoDB databases. However, results correspond more accurately to the observations if compared to those of initial engagement, probably because of the possibility to exploit the variable *distance* to determine if someone approached or not. In Table 8.2 the results from the three databases are compared to those obtained in the observations.

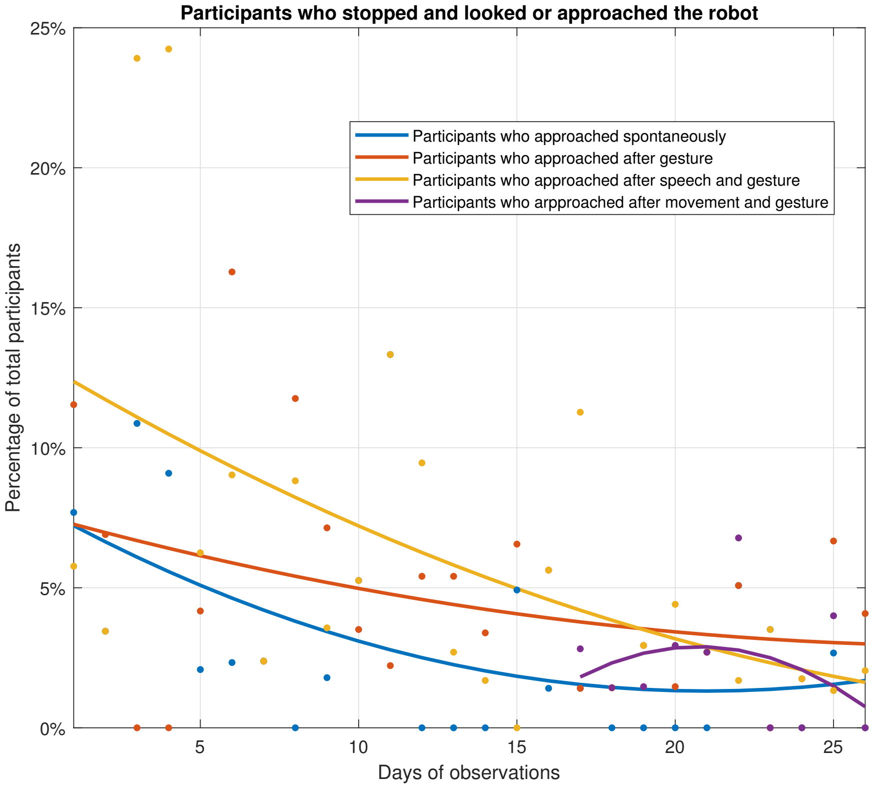


Figure 8.6: Graph showing the trend of the novelty effect for every behavior separately.

		Participants' behavior		
		Gesture	Speech and gesture	Movement and gesture
10^{th} September	Observations	1	6	3
	MongoDB	1	6	2
12^{th} September	Observations	1	1	1
	MongoDB	1	1	1
14^{th} September	Observations	2	2	2
	MongoDB	1	1	2

Table 8.2: Table comparing the results of observations and MongoDB for people approaching the robot, after the display of the different behaviors.

8.1.4 Groups

Groups of people have been recorded separately from individual participants, on dedicated observations sheets. During the 10 days of the experiment there were 47 groups that passed by *Pepper*: 35 couples, 10 groups of three people, 1 group of four people and 1 group of five people (109 participants in total). After triggering one of *Pepper*'s behavior, participants of the same group had the same reaction in 43 cases, while in the other 4 cases different reactions have been observed within the same group. The purely gestural behavior has been triggered by 20 groups (44 participants), the speech with gesture behavior by 16 groups (40 participants) and the movement with gesture behavior by 7 groups (17 participants). In 4 cases couples of people approached the robot spontaneously (8 participants). Data of groups have been recorded only during the experiment, therefore no novelty effect analysis can be conducted due to the limited data points available. Moreover, data is not homogeneous as different number of groups entered every day (e.g. 9 groups on day 7 and only 2 on day 10). A user engagement analysis of the behaviors is also hard to perform, because of the different amount of groups that triggered them. Only the gesture and speech with gesture behaviors have comparable amounts of groups (and participants), but too few triggered the movement with gesture to have a reliable insight into it. Therefore, data about this behavior are not analyzed. As shown in Figure 8.7, 40% of the groups looked at the robot after the purely gestural behavior, while 50% of them did so after the speech with gesture behavior.

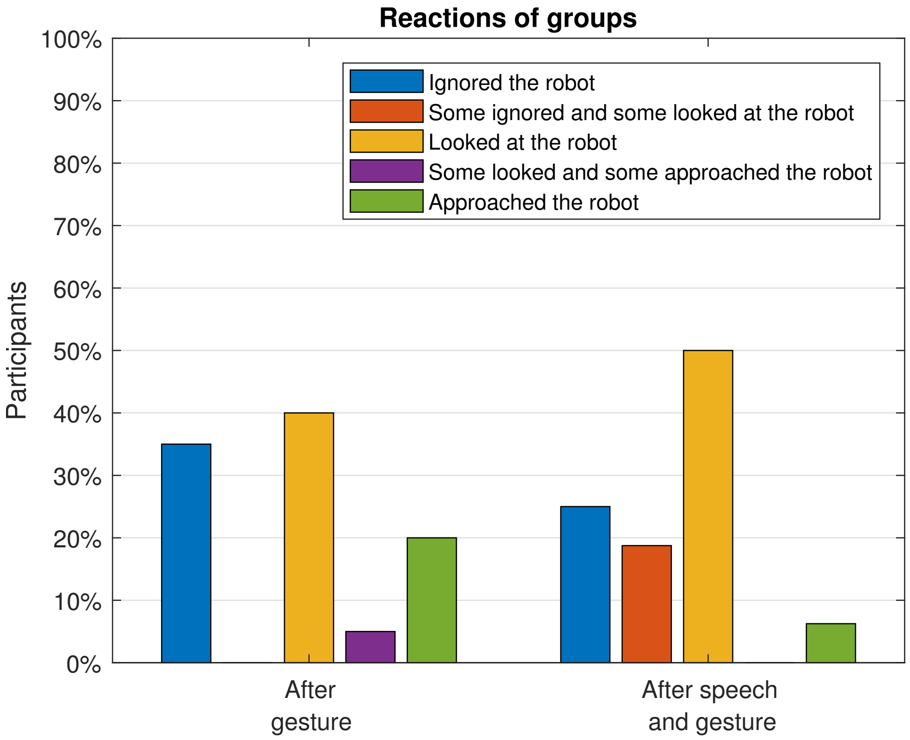


Figure 8.7: Graph showing groups' response to gesture and speech with gesture behaviors.

8.2 Discussion

Results about initial user engagement show how the speech and gesture behavior is the most effective in drawing attention, confirming what we initially hypothesized (H2: the behavior combining speech with gesture is going to be the most effective). Not only 51.6% of participants who were exposed to the behavior reacted looking at the robot, but 40% of them even waved or replied to *Pepper*. These reactions can be interpreted as an even stronger manifestation of initial engagement, because participants' attention is drawn to a point that they even generate a response for the robot, using the same social cues. If compared to the purely gestural behavior it is possible to appreciate how the addition of a verbal greeting made a significant difference for users, who replied or waved back to the robot almost twice as much. On the other hand, the movement with gesture behavior attracted significantly less attention than the speech with gesture one, but only shows a tendency to elicit significantly less reactions in users. Comparing the gestural and movement with gesture behavior, even if they seem to draw participants' attention in a comparable fashion, the latter appears to elicit a stronger engagement. Nevertheless, results are not significant in this case, so it is not possible to clearly determine which of the two has the strongest attention drawing power. Therefore, our first hypothesis is rejected (H1: behaviors integrating more social cues will be more effective in drawing attention), as we expected all behaviors combining multiple social cues to be significantly more engaging than the purely gestural one.

The comparison between male and female participants does not show any significant difference in their reactions, as their average response oscillates a few percentage points with respect to the overall mean. Analyzing the behavior of ”looking at the robot while walking” through the different days of observations, we observe a large variance for the gesture and speech with gesture behaviors. These deviations from the mean occur on the days that had the largest number of groups entering the building. For instance, on day 3 there were 7 groups of people, 3 of which triggered the gesture behavior. Their reactions are not included in the graph of Figure 8.2, which is possibly why it seems that less participants were engaged by such behavior. The same holds for day 7: 9 groups entered the building on that day and 5 of them triggered the speech with gesture behavior. This day is the only one in which this behavior scores slightly less than the gesture behavior. Overall, it is not possible to identify any learning effect from the trend of the plots. For what concerns the reactions of waving or replying to the robot while looking at it, the speech with gesture behavior had the most stable trend through the days, except for two days in which it scored a much higher responses than the average. For the two other behaviors there is a large difference in participants' response, with some days in which they score no reactions and others in which they score more than the speech with gesture one. Also in this case, the variance of the results does not allow to extrapolate any particular learning effect from the plots.

Results about the novelty effect showed that participants seemed much more curious about the robot in the first days of observations and progressively lost their interest. This trend is common to all behaviors, but the speech with gesture one seems to attract more participants at the beginning than the other, coherently with the findings about user engagement. The overall trend of the novelty effect has been obtained combining the results of the different behaviors in a single graph, which reflects what was expected according to literature. In fact, the curve of novelty significantly drops (H3: there will be a significant drop of participants affected by the novelty effect through the course of the pilot and experiment) in the first three weeks of observations (H4: the novelty effect will decrease between two and five weeks after the deployment of *Pepper* in the entrance hall) and seems to reach a steady state in correspondence of the first days of the experiment. This confirms both our hypotheses on the novelty effect and complies with findings of previous HRI research, where this phenomenon had been observed to wear off in a period between two and five weeks. After that, a steady state is reached, suggesting that results will be less affected by novelty so will be more reliable about initial engagement. This does not mean that no more users will approach the robot but they will do it much less than before.

The effect of novelty was also observed through the reactions of participants which deviated from what expected or that evolved in time. In fact, there were some users who came to the building every day, and their reaction to *Pepper* has been observed to change encounter after encounter. One subject, initially reacted very excitedly to the robot, approaching it, trying to handshake and trigger the different behaviors. However, after a couple of weeks his interest was gone, and he ignored the robot regardless of the behavior it displayed. Another subject was initially very excited about the robot as well, approached it and tried to understand what it could do. After a couple of weeks, he did not turn completely indifferent to it, but waved and replied to it every day, regardless of the behavior displayed. This suggests that novelty may wear off in different ways depending on the users, but additional metrics are needed to have such a fine grained view of this phenomenon.

Other reactions that are interesting to speculate on follow common patterns in different users. For instance, many users tested *Pepper*'s head tracking when they realized the robot was following them with the head. They moved from one side to the other of the robot for a couple of times to see if the robot kept looking at them. Another intriguing reaction happened after participants triggered the distance warning from *Pepper*. This happens when someone gets too close to the robot, which asks to step back so that it can see the participant better. Every time this warning was generated, all the participants executed what the robot told them. They stepped back, stood in front of it and waited. After realizing nothing else would have happened they either approached again or left. It would be interesting to investigate up to which point a robot can ask participants to do something and be obeyed. However this may also be purely due to novelty, because participants are curious to see what is going to happen and obey just to find out.

As for groups, data shows that in most of the cases participants react in the same way when they are together. Only the largest groups (of four and five people) and two other couples had mixed behaviors, which suggests that for small groups of 2-3 participants there is a general tendency to have the same reaction. The behaviors were not displayed evenly to groups, which is why no analysis of the movement with gesture behavior has been provided (only 7 groups triggered it versus the 16 of the speech and gesture behavior and the

20 of the gesture behavior). However, data for the two other behaviors reflects the trend already identified by the analysis of individuals, with 40% of groups looking at the robot after the gesture behavior, and 50% after the speech with gesture one. Data on groups that approached the robot shows too large variance and has been collected for a too short period to estimate a novelty effect trend.

8.3 Limitations

The main limitation of the experiment was the observation process itself. Even though there was a specific protocol to follow and the observation sheet had been refined during the days of the pilot, the observations were still recorded manually so the results may be affected by experimenter bias. Moreover, as the experimenter wrote the data on paper, it was not possible to precisely match a participant recorded on the observation sheet with a participant recorded by the MongoDB database (e.g. with a time stamp variable) and vice versa. However, this is also partially due to other technical limitations. In fact, while analyzing the data on MongoDB, we noticed that after a behavior was displayed there were from $2s$ to $3s$ that were not recorded on the database: the frames could not be collected because the behavior was running. Therefore, some of the participants who triggered these behaviors appeared in the database for less frames than what expected for entering users, so their data have been removed because they looked as exiting users. It is not possible to differentiate between entering users with few frames and exiting users, which explains the lower number of participants who looked at the robot in MongoDB than in the observations.

Moreover, the accuracy of the database may have also been hindered by processing speed limitations, as with all the off-board functionalities running on the GPU it was only possible to obtain a camera feed at $3fps$. Another limitation consisted in not having the behavior of movement with gesture integrated in the experiment GUI, which required the parallel use of *Choregraphe*. This choice did not allow to synchronize the movement with the waving gesture, which now occur in sequence, because of the safety constraints imposed by the programming tool. However, this limitation actually contributed to have more precise data for this behavior, because the use of *Choregraphe* allowed *Pepper* to correctly collect frames while the behavior was running. This explains why the results for the movement with gesture behavior most accurately correspond to those from the observations. A more contextual limitation is related to those users who hang out in the entrance hall to have a phone call. As they stay in *Pepper*'s field of view for several minutes they can alter the data from MongoDB, especially because they are not taken into account during the observations but they result as an extra person in the database.

Finally, another limitation was represented by the TV screen at the back of *Pepper*, as it was turned on during the experiment. In some situations the attention of participant might have been drawn by the TV rather than the robot, and it was hard to determine gaze orientation to discriminate between these cases. Sometimes this difference was noticeable as participant's head pose was different (i.e. slightly upward for the TV, slightly downward for the robot), but there might have been situations in which their reactions have been misinterpreted. However, this limitation might have affected our results only marginally, because we classified participants as ”looking at the robot while walking” if they turned their head toward the robot after behavior display. Therefore, it is unlikely that a significant number of them turned their head toward the TV after the robot displayed a behavior. This limitation affected both the observations recorded by the experimenter and the MongoDB data collection, as *OpenPose* does not estimate gaze orientation.

8.4 Conclusions

The results obtained from the experiment were used to evaluate *Pepper*’s drawing attention techniques and novelty effect. Data has been collected both manually by the experimenter and automatically by *Pepper* in a MongoDB database. Observations have been conducted for a total of 26 days: during the pilot, robot behaviors have been tested and refined, while in the last 10 days the experiment was conducted. On average 50 participants entered the building every day, providing a large data set to validate our research questions.

Throughout this project we identified metrics to use in data analysis for both initial engagement and novelty effect. These are related to the off-board algorithms that extended *Pepper*’s on-board functionalities. For instance, looking at the robot after the display of a specific behavior represents a feasible metric to measure if participants’ attention has been drawn, which is possible thanks to the key-points information from *OpenPose*. To measure the effect of novelty we considered people who approached or stopped and looked at the robot (as their reaction is regarded as more representative of this phenomenon), who were detected using *distance* information from the sonars in MongoDB. The limitations of these metrics reflect those of the off-board functionalities, which is why in future work they should be refined and extended. In fact, if with *OpenPose* we were able to detect also gaze direction, we would know more precisely if a person is looking at the robot even if not all the 5 key-points are visible. Moreover, using a depth map instead of on-board sonars will provide much clearer information about participants (both individuals and groups) who approached the robot. Nevertheless, the metrics identified and used for this experiment appropriately capture the information we wanted to collect, and provide a meaningful insight into both phenomena we wanted to analyze.

Initial user engagement was measured observing specific social cues that participants showed after the display of *Pepper*’s behaviors. We chose to monitor the ”looking at” behavior and the additional reactions of waving or replying to the robot, which denote a stronger level of engagement. Among the behaviors developed for *Pepper*, the one that accompanied a waving gesture with a verbal greeting appeared to be the most engaging. Results show that half of the participants exposed to this behavior reacted looking at the robot, while 40% of them waved or replied by saying something to the robot. This behavior is significantly more engaging than the others, which confirms what we had previously hypothesized (H2). With the results collected it is not possible to draw conclusions about which of the two other behaviors was the second most effective. Both of them obtained similar response in terms of ”looking at” but the movement with gesture seemed to elicit more reactions from participants. Therefore our first hypothesis (H1) is not confirmed by the results, as we expected the behaviors combining multiple social cues to be more engaging.

The novelty effect was examined using data from all the days of observations, in order to better appreciate the manifestation of this phenomenon. Also in this case, the speech with gesture behavior resulted to be the one attracting the most people toward the robot, so this finding further confirms that this behavior is the most effective in drawing attention. The results for all three behaviors and for the participants who approached the robot spontaneously show a decreasing trend (H3). This finding confirms that analyzing the effect of novelty considering participants who approached the robot was a reasonable choice. The global novelty effect appears to wear off in three weeks (H4), after which a steady state is reached. This does not mean that people will stop approaching the robot, but they will do it much less in comparison to the first days of observations. Throughout this period we observed a drop of almost 20% of the total participants approaching the robot, which suggests that novelty effect is less present.

8.5 Contribution

In this project we extended *Pepper*’s drawing attention techniques by making three main contributions:

- We tested the on-board capabilities of the robot and identified their main limitations using objective and subjective metrics. *Pepper* is a social robot that has been specifically created to interact with humans. However, its on-board capabilities were not sufficient to successfully elicit initial engagement in users. Through a series of preliminary experiments and a baseline application tested in a pilot experiment, its interactive skills have been examined to understand how they could have been refined to draw users’ attention at the beginning of an interaction. A technical analysis as well as a user centered research have been conducted, to understand the main sensory limitations and collect user feedback.
- We extended the robot’s functionalities to implement new techniques of initial engagement. Several state-of-the-art algorithms have been tested in order to extend *Pepper*’s functionalities. The first pilot experiment revealed that people detection and face tracking were the fields requiring major refinement. In addition, off-board head pose and depth estimation systems have been analyzed to improve *Pepper*’s initial engagement techniques. Within these fields, different algorithms have been compared, and those that were finally selected have been described in terms of their advantages and limitations. Thanks to these extended functionalities it was possible to design new behaviors to elicit initial engagement in participants with *Pepper*.
- We provided an evaluation of how these behaviors affected participants, both in terms of novelty effect and initial engagement. These phenomena have been measured dur-

ing 26 days of observations, split between 16 of pilot and 10 of experiment. The new behaviors have been analyzed on the basis of observations recorded by the experimenter and data collected on the MongoDB database. We showed that the behavior combining speech with gesture had the highest drawing attention power on participants, and that there was a novelty effect which wore off through the course of the experiment.

8.6 Future work

Throughout this project we encountered several limitations that need to be overcome for future research. First, *Pepper*'s functionalities need to be further extended to improve the detection and classification of participants. For instance, with an algorithm for depth estimation that measures participant's distance from the robot more accurately and on a broader range than the current on-board sonar. Another useful extension would be an algorithm for the detection of gaze orientation, to understand whether a participant is actually looking at *Pepper* and avoid biasing the results with misinterpretations. Second, the observation process needs to be digitalized in order to have a time-stamp correspondence between data collected on MongoDB and on the observation sheets. This will also reduce the data cleaning effort and provide more reliable information about participants. Third, the movement with gesture behavior should be integrated in the experiment GUI. Outside of the *Choregraphe* environment it will also be possible to synchronize the gesture with the movement rather than having them in sequence.

Once overcome current limitations, research with *Pepper* can progress with the identification of new behaviors that can draw people's attention. During our experiment we found that the combination of speech with gesture was the most engaging for participants, but that the addition of movement to waving did not score a significantly higher success than the purely gestural behavior. Therefore, it would be interesting to investigate how a behavior including all these social cues (gestural, proximity and verbal cues) would draw the attention if compared with those analyzed. A different design choice that is important to examine is the intensification of a single social cue rather than the combination of more. For instance, a behavior combining multiple gestures such as waving and another arm movement that invites participants to approach. This has already been partially explored with the speech with gesture behavior, where the greeting ”Good morning!” has been accompanied by the question ”How are you?” obtaining a higher drawing attention rate than the greeting alone.

The purpose of designing drawing attention behaviors for *Pepper* is to elicit initial engagement in participants. Future studies could exploit the results of this project to investigate how to further strengthen initial engagement and successfully attract people in front of the robot. This can be achieved by both combining and intensifying social cues in more complex and effective behaviors. The next natural step would be to smoothly bring participants to initiate an interaction, for instance by having *Pepper* ask them to do a simple interactive task, which can be made more engaging by using the tablet on the chest of the robot. From this point, the goal would be to maintain participant's engagement throughout the interaction, in particular through the use of gaze cues such as blinking, and eventually disengage. Finally, it is fundamental for future research to collect users' feedback on the encounter with *Pepper* with post-experimental interviews or questionnaires, to gain a deeper insight of the interaction with subjective metrics.

Appendix A

Questionnaire

Delft University of Technology, The Netherlands

Questionnaire on the interaction with a Pepper robot near a coffee machine

Your participation in this experiment is voluntary and your responses are strictly confidential. There are no known risks involved and you can withdraw at any moment.

Date:_________________

Gender : □ Male □ Female

Have you interacted with a robot before? □ Never □ Rarely □ Daily

Please answer the following questions.

	Scale				
Question	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The robot noticed my presence					
The robot was attentive to me during the conversation					
The robot interpreted correctly what I said					
The verbal and bodily expressions used by this robot felt natural					
I find it easy to interact with a robot					
I find it intuitive to interact with a robot					
I appreciated seeing a robot near the coffee machine					

Do you have any suggestion or recommendations for future developments?

Appendix B

Observation sheet

Human-Robot Initial Engagement Observation Sheet for Individuals

Total number of people passing by

Gender

□ □

M F

Date	____ /____/ 2018
Start time	
End time	

Passerby behavior	Robot Behavior														
	Gestures					Speech & Gestures					Movement & Gestures				
Ignore the robot															
	Passerby reaction														
	WAVE	TOUCH ROBOT	SPEAK	TAKE PICTURE	OTHER	WAVE	TOUCH ROBOT	SPEAK	TAKE PICTURE	OTHER	WAVE	TOUCH ROBOT	SPEAK	TAKE PICTURE	OTHER
Stop and look at the robot															
Look at the robot while walking															
Approach the robot															

Reflexive comments

Appendix C

NAOqi Framework

C.1 NAOqi Audio

This module contains a set of methods that *Pepper* uses in response to sensory information coming from its microphones to fulfill specific activities using the speakers. For the purpose of this project sound localization and detection, as well as speech management, are the main aspects of interest.

ALSpeechRecognition is the method used to recognize words or phrases in different languages, it stores them in the variables *WordRecognized* and *WordRecognizedAndGrammar*. Both of them are associated with a *confidence* parameter to estimate the likelihood that utterances understood were actually pronounced by the interlocutor, and the latter one is also used to gather information about the grammatical structure of a sentence.

ALAnimatedSpeech uses loudspeakers for communication and employs **ALTextToSpeech** to say text, so it inherits also its functionalities of voice customization. Furthermore, it supports the integration of tags within the text to add expressiveness to the robot's behavior. Being connected to **ALSpeakingMovement** it already displays animations when *Pepper* is talking, but allows to implement modifications and contribute with additional annotations.

C.2 NAOqi Vision

This module contains all those methods that *Pepper* uses to interpret the sensory input from the 2D cameras and the 3D depth sensor. Among the several functionalities, the one of major interest for this project was **ALMovementDetection**, which allows to detect movements that occur in the field of view of the robot. The method collects and compares successive frames to identify the clusters of pixels that changed during the time interval. Using its 3D sensor the robot can eventually compute the distance to the points that moved. This application is limited only to those scenarios in which the robot is not moving, like the one proposed for this study.

C.3 NAOqi People Perception

This module contains all the methods used by *Pepper* to analyze the people around it. They allow to keep track of users and their gazes while interacting with the robot, and also detect waving thanks to the 3D depth sensor. Those of major interest for this project provide information about proxemics and face detection.

ALEngagementZones gives access to the configuration of the engagement zones that *Pepper* uses to classify people according to their distance form it. These can be adjusted using the parameters *FirstDistance*, *SecondDistance* and *LimitAngle*, which are illustrated in Figure 3.5. Between the robot and the *FirstDistance* there is engagement zone 1, between *FirstDistance* and *SecondDistance* zone 2, while zone 3 is generally considered all the space beyond *SecondDistance*.

ALFaceDetection is the method used to detect and recognize faces of people in the field of view of the robot. It extracts location and orientation of faces as well as information about the main facial features (eyes, nose and mouth). Face recognition requires an initial training of the robot but is limited to specific lighting conditions and is less robust than detection, mainly because it does not have a 3D representation of the person.

ALPeoplePerception keeps track of the people around the robot and collects information about them. It uses the same tracking algorithm employed for face detection but with a different target dimension. It keeps in memory also people that exit the field of view of the robot in case they come back, and shows the same limitations related to lighting conditions already illustrated above.

C.4 NAOqi Event

The largest majority of the modules previously described exploits the events raised by the Memory module of the robot. Subscribing to a specific event, it is possible to adapt the behavior of *Pepper* to react accordingly. New events may also be created by developers, but a long list of them is already available [24].

Bibliography

[1] Amir Aly and Adriana Tapus. A model for synthesizing a combined verbal and nonverbal behavior based on personality traits in human-robot interaction. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 325–332. IEEE Press, 2013.

[2] 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.

[3] Salvatore M Anzalone, Sofiane Boucenna, Serena Ivaldi, and Mohamed Chetouani. Evaluating the engagement with social robots. *International Journal of Social Robotics*, 7(4):465–478, 2015.

[4] Niklas Bergstrom, Takayuki Kanda, Takahiro Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. Modeling of natural human-robot encounters. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2623–2629. IEEE, 2008.

[5] Cindy L Bethel and Robin R Murphy. Survey of non-facial/non-verbal affective expressions for appearance-constrained robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(1):83–92, 2008.

[6] BrainBotics. Brainbotics is now certified partner of softbank robotics, 2018.

[7] Allison Bruce, Illah Nourbakhsh, and Reid Simmons. The role of expressiveness and attention in human-robot interaction. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4138–4142. IEEE, 2002.

[8] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[9] Umberto Castiello. Understanding other people's actions: intention and attention. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):416, 2003.

[10] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[11] Dipankar Das, Yoshinori Kobayashi, and Yoshinori Kuno. Attracting attention and establishing a communication channel based on the level of visual focus of attention. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2194–2201. IEEE, 2013.

[12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[13] Beat Fasel and Juergen Luettin. Automatic facial expression analysis: a survey. *Pattern recognition*, 36(1):259–275, 2003.

[14] Markus Finke, Kheng Lee Koay, Kerstin Dautenhahn, Chrystopher L Nehaniv, Michael L Walters, and Joe Saunders. Hey, i'm over here-how can a robot attract people's attention. In *ROMAN*, 2005.

[15] Stephen M Fiore, Travis J Wiltshire, Emilio JC Lobato, Florian G Jentsch, Wesley H Huang, and Benjamin Axelrod. Toward understanding social cues and signals in human–robot interaction: effects of robot gaze and proxemic behavior. *Frontiers in psychology*, 4:859, 2013.

[16] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[18] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[19] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[20] Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, Alan C Schultz, et al. Designing robots for long-term social interaction. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1338–1343. IEEE, 2005.

[21] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[22] A. S. Group. Aldebaran documentation: Technical overview, 2018.

[23] A. S. Group. Former naoqi framework, 2018.

[24] A. S. Group. Naoqi event index, 2018.

[25] Edward T Hall. The hidden dimension: man's use of space in public and private the bodley head. *London, Sydney, Toronto*, 121, 1969.

[26] Joanna Hall, Terry Tritton, Angela Rowe, Anthony Pipe, Chris Melhuish, and Ute Leonards. Perception of own and robot engagement in human–robot interactions and their dependence on robotics knowledge. *Robotics and Autonomous Systems*, 62(3):392–399, 2014.

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[28] Kevin Anthony Hoff and Masooda Bashir. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors*, 57(3):407–434, 2015.

[29] Mohammed Moshiul Hoque, Tomomi Onuki, Dipankar Das, Yoshinori Kobayashi, and Yoshinori Kuno. Attracting and controlling human attention through robot's behaviors suited to the situation. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 149–150. ACM, 2012.

[30] Chien-Ming Huang and Bilge Mutlu. Robot behavior toolkit: generating effective social behaviors for robots. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 25–32. ACM, 2012.

[31] Chien-Ming Huang and Andrea Lockerd Thomaz. Joint attention in human-robot interaction. In *AAAI Fall Symposium: Dialog with Robots*, 2010.

[32] Stephanie Kim. Understanding facial recognition through openface, 2018.

[33] Christopher Lee, Neal Lesh, Candace L Sidner, Louis-Philippe Morency, Ashish Kapoor, and Trevor Darrell. Nodding in conversations with a robot. In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*, pages 785–786. ACM, 2004.

[34] Iolanda Leite, Carlos Martinho, and Ana Paiva. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013.

[35] Iolanda Leite, Carlos Martinho, Andre Pereira, and Ana Paiva. As time goes by: Long-term evaluation of social presence in robotic companions. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 669–674. IEEE, 2009.

[36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017.

[37] 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.

[38] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[39] Vicky A Lofthouse and Debra Lilley. What they really, really want: user centered research methods for design. In *DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia*, 2006.

[40] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[41] Marek P Michalowski, Selma Sabanovic, and Reid Simmons. A spatial model of engagement for a social robot. In *Advanced Motion Control, 2006. 9th IEEE International Workshop on*, pages 762–767. IEEE, 2006.

[42] MongoDB. Mongo db, 2018.

[43] Andrew Murphy. Commercial: Robotics outlook 2025, 2017.

[44] Bilge Mutlu, Toshiyuki Shiwa, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Footing in human-robot conversations: how robots might shape participant roles using gaze cues. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 61–68. ACM, 2009.

[45] Heather L O'Brien and Elaine G Toms. What is user engagement? a conceptual framework for defining user engagement with technology. *Journal of the Association for Information Science and Technology*, 59(6):938–955, 2008.

[46] Hiroshi G Okuno, Kazuhiro Nakadai, and Hiroaki Kitano. Social interaction of humanoid robot based on audio-visual tracking. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 725–735. Springer, 2002.

[47] Massimiliano Patacchiola and Angelo Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71:132–143, 2017.

[48] Karola Pitsch, Hideaki Kuzuoka, Yuya Suzuki, Luise Sussenbach, Paul Luff, and Christian Heath. the first five seconds: Contingent stepwise entry into an interaction as a means to secure sustained engagement in hri. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 985–991. IEEE, 2009.

[49] Joseph Redmon. Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`, 2013–2016.

[50] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017.

[51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[52] Claire Rivoire and Angelica Lim. Habit detection within a long-term interaction with a social robot: an exploratory study. In *Proceedings of the International Workshop on Social Learning and Multimodal Interaction for Designing Artificial Agents*, page 4. ACM, 2016.

[53] Satoru Satake, Takayuki Kanda, Dylan F Glas, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. How to approach humans?: strategies for social robots to initiate interaction. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 109–116. ACM, 2009.

[54] 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.

[55] Julia Schwarz, Charles Claudius Marais, Tommer Leyvand, Scott E Hudson, and Jennifer Mankoff. Combining body pose, gaze, and gesture to determine intention to interact in vision-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3443–3452. ACM, 2014.

[56] Candace L Sidner and Christopher Lee. Engagement rules for human-robot collaborative interactions. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 4, pages 3957–3962. IEEE, 2003.

[57] Candace L Sidner, Christopher Lee, Cory D Kidd, Neal Lesh, and Charles Rich. Explorations in engagement for humans and robots. *Artificial Intelligence*, 166(1-2):140–164, 2005.

[58] Maria Staudte and Matthew W Crocker. Visual attention in spoken human-robot interaction. In *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*, pages 77–84. IEEE, 2009.

[59] JaYoung Sung, Henrik I Christensen, and Rebecca E Grinter. Robots in the wild: understanding long-term use. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 45–52. ACM, 2009.

[60] Zahari Taha and Jessnor Arif Mat Jizat. A comparison of two approaches for collision avoidance of an automated guided vehicle using monocular vision. In *Applied Mechanics and Materials*, volume 145, pages 547–551. Trans Tech Publ, 2012.

[61] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Transactions on pattern analysis and machine intelligence*, 24(9):1226–1238, 2002.

[62] Elena Torta, Jim van Heumen, Raymond H Cuijpers, and James F Juola. How can a robot attract the attention of its human partner? a comparative study over different modalities for attracting attention. In *International Conference on Social Robotics*, pages 288–297. Springer, 2012.

[63] Wikipedia. Pepper (robot) - wikipedia, the free encyclopedia, 2018.

[64] Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. A model of proximity control for information-presenting robots. *IEEE Transactions on Robotics*, 26(1):187–195, 2010.