UNIVERSITA’ DEGLI STUDI DI ROMA

“LA SAPIENZA”

DIPARTIMENTO DI INGEGNERIA INFORMATICA, AUTOMATICA E GESTIONALE “ANTONIO RUBERTI”

Master Degree Thesis in

ARTIFICIAL INTELLIGENCE AND ROBOTICS

Personalized Interactions for an Assistive Robot

Supervisor

Prof. Luca Iocchi

Student

Giorgia Piernoli 1648511

Academic Year 2019/2020

Contents

Chapter 1

Introduction

1. Context and definition of the problem

During the past years, the advancement in robotics facilitated the deployment of robots not only in laboratories and industrial environments but also in people's daily lives. As technology advanced, it became possible to extend the Robot working space to public environments inhabited by people. In these environments, robots perform tasks and services autonomously or in cooperation with the "occupants" with whom robots share the working space. Generally, robots' deployment ranges from "entertainment" to "supportive" tasks with the objective on helping humans performing difficult or tedious tasks. A typical example of these applications comes from robotic systems able to act as home companions or assistants in health-care / elderly-care or rehabilitation. Nowadays, automated deployments are mainstream in education, entertainment, industrial and homeland security sectors, as well as safety and research. As a result of this multi-sector and multi-purpose adoption, robots are nowadays ideated and designed to become part of people everyday's life. The role of robots in society keeps expanding and diversifying, bringing with it a host of issues surrounding the relationship between robots and humans. To be able to work in an environment shared with people, the first requirement of a robotic system is for its automated operations to be safe for humans. “Human-Robot Interaction is the capability of enabling robots to successfully interact with humans”. Establishing Interaction with humans around it allows the Robot asking them for aid and, therefore, receiving the help needed.

With my thesis submission, I extended the boundaries of the current TERESA project (Therapeutic Educational Robot Enhancing Social InterActions) whose objective is to develop a project for human-robot Interaction through Pepper.

Pepper is a humanoid developed by SoftBank Robotics. In this project, Pepper is named TERESA, and the objective is to prove that the presence of the Robot could be supporting medical therapies. The evaluation of TERESA involvement benefits is observed and tested through the organization of a series of therapeutic, educational meetings with obese patients. During these meetings, doctors leveraged already available and tested didactic contents. The only difference in the execution of these sessions is the presence of TERESA; whose role is to assist the doctor. These therapeutic sessions are led by a physician whose responsibility is to provide educational content to patients. The physician is also responsible for the discussion and interactions between patients and doctors. During the therapy sessions, another member of the medical staff is responsible for monitoring and controlling the Robot. Being TERESA not able to understand when is the right moment to intervene in the didactic conversation, an assistant doctor has the role to guide and determine the right timing for TERESA to interact, using a simple web interface. As a guiding principle, the assistant physician determines and decides when it is the right time for TERESA to contribute, intervene and deliver the educational content. At this point, after the activation by the medical assistant, TERESA will autonomously carry out the predefined interactions. The fact that the medical assistant executes the control of the Robot is communicated with the patients. Although this control of the robot via the medical assistant is not explicitly hidden to the patients (as the medical assistant is present in the room where the therapy session takes place and does not hide the fact that he interacts with a PC), patients tend to identify the robot as an "intelligent" assistant able to support the primary physician. On the other hand, the didactic contents included in the robot have a lower level of details than the ones used by the main doctor, because the aim is to not create confusion in patients about the role of the main doctor and the role of the robot assistant. The interactions implemented in TERESA support the topics described by the principal physician and are typically provided in the form of summaries of what has been said previously, examples of applications and questions. After a few sessions, the doctor agrees that the presence of TERESA is helping the interaction with the patients. The doctor also addresses the fact that this interaction should be personalized. For example, when TERESA communicates verbally, it does not address the conversation with a specific person's name. The Robot does not look to the person while it's talking to him/her. Pepper uses generic wording and is looking in no direction. Doctors believe that in the interaction in which TERESA needs to address to a specific person, it should be able to call this person by his/her name and turn its head towards him/her. All these would make patients more engaged and supported.

In Chapter 4 of my thesis I will evolve these concepts further on one of the critical parts of my work.

The second step of my work starts from this TERESA's project to be extended to an autonomous work regarding the movement of the robot through tags, through visual recognition techniques. Without going into details, the primary purpose of my thesis is to evolve and personalize the Interaction between robot-patient. I will extend the TERESA project focusing on the movements of the Robot. These tag-images will allow the Robot to understand how to move, and when to rotate towards one person or the other. Above will be explored in Chapter 4, where I will summarize my assumptions, research and document my solutions. At the end I studied the robustness of the detection system, I did a lot of experiments to show how the performance of this system increases or decreases in presence of occlusions and light changes.

1.1 Motivation

During my Master program in Artificial Intelligence and Robotics, and as part of the "Human-Robot Interaction" course, I worked on a project "Pepper as a controller for COVID-19 rules in supermarkets". The goal of this project was to guarantee the application and respect of COVID-19 rules by users entering in a supermarket. I found this project very interesting: I was fascinated about how a robot can support humans in different use cases and therefore I decided to continue this research as a case of my thesis.

What excited and inspired me in choosing Human-Robot Interaction as the subject of my thesis is the belief that human-robot collaboration can significantly support different use cases and how it can improve the human condition by removing frictions from people's lives.

The fact that the patients of the TERESA project felt more comfortable responding to addressed questions by the Robot rather than the ones addressed by the doctor makes me reflect on the importance and potential of robot-human integration in different real-life scenarios.

What struck me the most in the TERESA project, is the fact that somehow the scientific world, i.e Robot’s one,can influence human psychology and human way of feeling.

I believe that in today's world, the best results can be obtained through the fusion and collaboration between different disciplines, working together towards a common goal.

1.2 Chapter organization

- *Chapter One* gives a brief introduction of the context of the project, the opening of the thesis' objectives and my motivation for my work.
- *Chapter Two* talks about related work. Papers that inspired me and guided me through the TERESA's project.
- *Chapter Three* gives an overview of the software and tools used in the thesis.
- *Chapter Four* explains the project in detail, showcasing how the personalization of the interaction and of Robot’s moves have been ideated and executed.
- *Chapter Five* shows the experiments and results that I compute during my work on Human-Robot Interaction.
- *Chapter Six* is the experimental-quantitative part: robustness of the detection's system.
- *Chapter Seven* presents some conclusions and proposes future works.

Chapter 2

Related works

There are some works that I found significant for the creation of this project that show how much the presence of the robot is useful. As I said, robots are increasingly becoming able to perform tasks and services autonomously or in cooperation with the occupants who share the working space. Generally, their employment may move around supporting humans in performing difficult or boring tasks. I'm going to briefly present some examples of these applications, some papers, topics and projects that inspired.

2.1 Design of robot teaching assistants through multi-modal human-robot interactions [1]

In "*Design of robot teaching assistants through multi- modal human-robot interactions*" [Ferrarelli *et al.*, 2017 P. Ferrarelli, M. T. Lázaro, e L. Iocchi] the authors developed the idea that during the last years, the interest in introducing robotics in schools has increased significantly, due to the possibilities it offers from the educational perspective.
Robots in school become an interactive and more visually appealing educational tool attracting the interest of students who become active subjects during the learning process, instead of listening passively to a lesson. Through the use of robots in the classroom, the students can reinforce certain contents explained during the lessons in different ways, for example, developing themselves real applications that can be executed on the robot or through interactive sessions with the robot. This is possible because latter they propose a framework for the generation of multi-modal interfaces for human-robot interaction.

2.2 Pepper as a controller for COVID-19 rules in supermarkets

In my project for the Elective in Artificial Intelligence, in particular the course named Human-Robot Interaction, "*Pepper as a controller for COVID-19 rules in supermarkets*", the idea was to use Pepper to guarantee the application and respect of COVID-19 rules by users before entering the supermarket. Due to the pandemic, we were not able to physically work with the robot, so we used MODIM (Multi-MOdal Interaction Manager) and other tools to explain the use case and overall experience. The use case was the following: a robot is placed at the entrance of the supermarket to ensure that people are effectively wearing gloves and masks before entering; Pepper reminds the users about the importance of wearing gloves and masks, and shows to the users how to take them from the boxes located near to the robot. Pepper also shows an informative video to the users about the main informations about COVID-19 and rules we need to respect. (This project is available online in the youtube page: https://youtu.be/2h-ve8SNi0o) The project aims are to create a new experience through a robot, to interact and inform people about how to behave before entering a supermarket in the pandemic situation. The idea of this project comes out from the current need of reducing the number of interactions between people and paying attention to safety rules imposed by the COVID-19 pandemic. We imagined this humanoid as a "controller" and a "facilitator". Indeed, we think that this is a safer way to inform people in order to reduce the risk of new infections: a robot will not be infected by COVID-19 while doing its work, while a person could be exposed to huge number of people. We also think that many people are still not aware of the COVID-19 risks; a robot would communicate engaging with users on why obeying rules is essential and why all of us should follow them. The robot could also play the role of a new interactive channel to encourage safe behaviors between citizens. We are looking at guaranteeing the safety and health of different stakeholders: people who need to access a supermarket, as well as supermarket employees and business partners.

2.3 Personalized short-term multi-modal interaction for social robots assisting users in shopping malls

In the paper "*Personalized short-term multi-modal interaction for social robots assisting users in shopping malls*" [Luca Iocchi1, Maria Teresa L´azaro1, Laurent Jeanpierre2, Abdel-Illah Mouaddib] in which they talk about the new challenge for robotics in the near future that is to deploy robots in public areas (malls, touristic sites, parks, etc.) to offer services and to provide customers, visitors, elderly or disabled people, children, etc. with increased welcoming and easy to use environments. Such application domains present new scientific challenges: robots should assess the situation, estimate the needs of people, socially interact in a dynamic way and in a short time with many people, exhibit safe navigation and respect the social norms. These capabilities require the integration of many skills and technologies. Among all these capabilities, in this paper they focus on a particular form of Human-Robot Interaction (HRI): Personalized Short-term Multi-Modal Interactions. In this context, Personalized means that the robot should use dif- ferent forms of interactions to communicate the same concept to different users, in order to increase its social acceptability; Short-term means that the interactions are short and focused on only one particular communicative objective, avoiding long and complex interactions; while Multi-modality is obtained by using different interaction devices on the robot (although in this study, we focus only on speech and graphical interfaces). The solution described in this paper is developed within the context of the COACHES project1, that aims at developing and deploying autonomous robots providing personalized and socially acceptable assistance to customers and shop managers of a shopping mall. The main contribution of this paper is on the architecture of the Human-Robot Interaction module that has several novelties and advantages: 1) integrated management of all the robotic activities (including basic robotic functionalities and interactions) through the use of Petri Net Plans, 2) explicit representation of social norms that are domain and task independent, 3) personalized interactions obtained through explicit representation of information and not hand-coded in the implementation of the robot behavior.

2.4 *COACHES: An assistance Multi-Robot System in public areas*

“COACHES: An assistance Multi-Robot System in public areas” [L. Jeanpierre, A.-I. Mouaddib, L. Iocchi, M.T. Lazaro, A. Pennisi, H. Sahli, E. Erdem, E. Demirel and V. Patoglu]. COACHES project provides a modular architecture integrated in real robots. They deployed COACHES at Caen city in the ”Rives de l’Orne” shopping mall. COACHES is a cooperative system consisting of fixed cameras and the mobile robots. The fixed cameras can do object detection, tracking and abnormal events detection (objects or behavior). The robots combine these information with the ones perceived via their own sensor, to provide information through its multi- modal interface, guide people to their destinations, show tramway stations and transport goods for elderly people, etc. The COACHES robots will use different modalities (speech and displayed information) to interact with the mall visitors, shopkeepers and mall managers.

2.5 TERESA (Therapeutic Educational Robot Enhancing Social interActions)

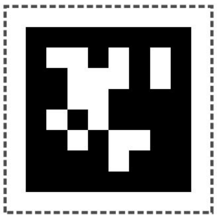
Tha main paper that inspired me and give me the possibility to do this work is the “TERESA: robot sociale per l’assistenza terapeutica”. Obesity is a chronic multifactorial disease caused by genetic, psychological, social, environmental factors, incorrect eating habits and low level of physical activity [Burgess et al., 2017]. Few subjects maintain the results achieved after diet therapy over the long term. There are several factors that hinder the maintenance of weight loss: environmental, emotional, biological, behavioral, cognitive. Another reason that hinders weight loss and its maintenance is the presence of Binge Eating Disorder (BED). In recent years, therefore, the need has arisen to promote persistent lifestyle changes in obese subjects, to improve adherence to treatment and consolidate the results obtained over time. In the literature, lifestyle modification interventions in obese patients are generally focused on nutrition, exercise and behavioral strategies and require a multidimensional and multidisciplinary approach. Therapeutic Education (ET) has taken on an important role as a multidisciplinary intervention aimed at improving lifestyle and developing new skills for disease management. Although the vast majority of people who have attended ET programs have found them useful, some do not maintain participation in these programs. Assistive technologies are powerful tools for increasing knowledge and improving participation.The main objective of the project is the evaluation of the efficacy and acceptability of the use of a social robot as a doctor's assistant during therapeutic education sessions for obese patients. To this end, a software for human robot social interaction was developed, using the humanoid-like robot SoftBank Pepper. The robot used in the project is called TERESA (acronym for Therapeutic Educational Robot Enhancing Social interActions). In particular, the TERESA social robot aims to stimulate, increase and improve social interactions between patients and doctors and between patients themselves, in order to increase motivation, participation and fun in therapeutic sessions and to reduce anxiety, embarrassments and other negative attitudes.

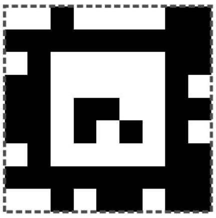
2.6 Visual Servoing

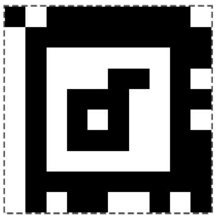
Vision allows a robotic system to obtain geometrical and qualitative information on the surrounding environment to be used both for motion planning and control. [13] In particular, control based on feedback of visual measurements is termed visual servoing. There are some basic algorithms for image processing, aimed at extracting numerical information referred to as image feature parameters. These parameters, relative to images of objects present in the scene observed by a camera, can be used to estimate the pose of the camera with respect to the object and vice versa. There are also the analytic pose estimation methods, based on the measurement of a certain number of points or correspondences. There are also the numerical pose estimation methods, based on the integration of the linear mapping between the camera velocity in the operational space and the time derivative of the feature parameters in the image plane. In cases in which multiple images of the same scene, taken from different viewpoints, are available, additional information can be obtained using stereo vision techniques and epipolar geometry. A fundamental operation is also camera calibration; to this end, a calibration method based on the measurement of a certain number of correspondences is present. There are two main approaches to visual servoing: position-based visual servoing and image-based visual servoing.

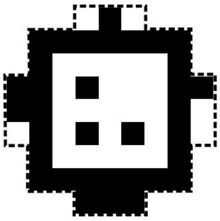
2.7 *AprilTag: A robust and flexible visual fiducial system*

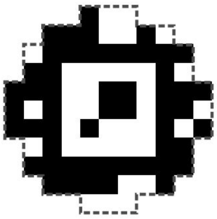
"*AprilTag: A robust and flexible visual fiducial system*" [Edwin Olson]: Visual fiducials are artificial landmarks designed to be easy to recognize and distinguish from one another. Although related to other 2D barcode systems such as QR codes [1], they have significant goals and applications. With a QR code, a human is typically involved in aligning the camera with the tag and photographs it at fairly high resolution obtaining hundreds of bytes, such as a web address. In contrast, a visual fiducial has a small information payload (perhaps 12 bits), but is designed to be automatically detected and localized even when it is at very low resolution, unevenly lit, oddly rotated, or tucked away in the corner of an otherwise cluttered image. Aiding their detection at long ranges, visual fiducials are comprised of many fewer data cells: the alignment markers of a QR tag comprise about 268 pixels (not including required headers or the payload), whereas the visual fiducials described in this paper range from about 49 to 100 pixels, *including* the payload.Unlike 2D barcode systems in which the position of the barcode in the image is unimportant, visual fiducial systems provide camera-relative position and orientation of a tag. Fiducial systems also are designed to detect multiple markers in a single image. Visual fiducial systems have been used to improve human/robot interaction, allowing humans to signal commands (such as "follow me" or "wait here") by flashing an appropriate card to a robot.

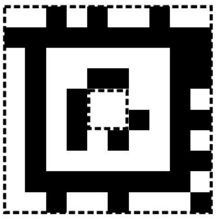

Tag36h11


TagStandard41h12


TagStandard52h13


TagCircle21h7


TagCircle49h12


TagCustom48h12

In this paper, they describe a new visual fiducial system that significantly improves performance over previous systems. The central contributions of this paper are:

1. They describe a method for robustly detecting visual fiducials. We propose a graph-based image segmentation algorithm based on local gradients that allows lines to be precisely estimated. We also describe a quad extraction method that can handle significant occlusions.
2. They demonstrate that our detection system provides significantly better localization accuracy than previous systems.
3. They describe a new coding system that addresses problems unique to 2D barcoding systems: robustness to rotation, and robustness to false positives arising from natural imagery. As demonstrated by our experimental results, our coding system provides significant theoretical and real-world benefits over previous work.
4. They specify and provide results on a set of benchmarks which will allow better comparisons of fiducial systems in the future.

2.8 *AprilTag 2: Efficient and robust fiducial detection*

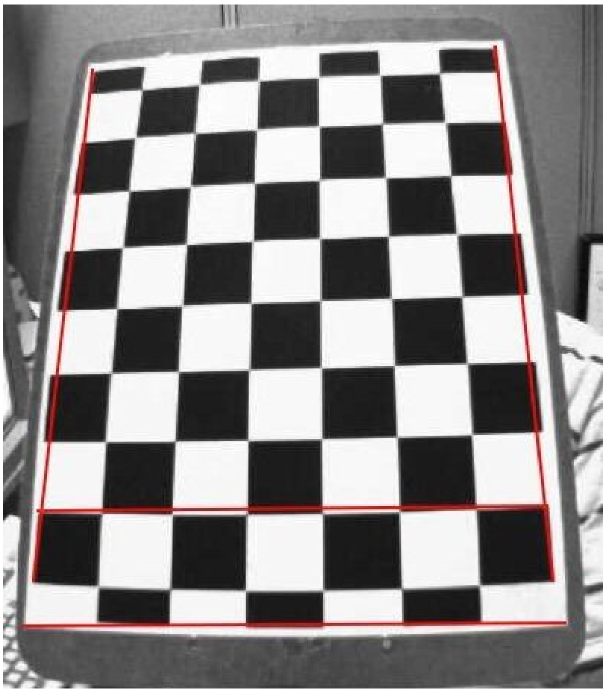
"*AprilTag 2: Efficient and robust fiducial detection*" [John Wang and Edwin Olson], Fiducials are artificial visual features designed for au- tomatic detection, and often carry a unique payload to make them distinguishable from each other. Although these types of fiducials were first developed and popularized by augmented reality applications [1], [2], they have since been widely adopted by the robotics community. Their uses range from ground truthing to object detection and tracking, where they can be used as a simplifying assumption in lieu of more sophisticated perception.A few key properties of fiducials make them useful for pose estimation or object tracking in robotics applications. Their uniqueness and high detection rate are ideal for testing SLAM systems. Fixed fiducial markers can be used for visual localization or as a ground truth estimate of robot motion. Fiducials mounted on objects can be used to identify and localize objects of interest. This work is based on the earlier AprilTag system [3]. The design of AprilTags as a black-and-white square tag with an encoded binary payload is based on the earlier ARTag [2] and ARToolkit [1]. AprilTag introduced an improved method of generating binary payloads, guaranteeing a minimum Hamming distance between tags under all possible rotations, making them more robust than earlier designs. The tag generation process, a lexicode-based process with minimum complexity heuristics, was empirically shown to reduce the false positive rate compared to ARTag designs of similar bit length.Based on feedback from AprilTag users in the robotics community, we determined that most users do not accept tags with decode errors. In these cases, features such as support for recovering partially-occluded tag borders are seldom useful. This functionality must be weighed against the costs of additional computation time and an increased false positive rate.This work describes a method for improving AprilTag detection speed and sensitivity while trading off the ability to detect partially-occluded tags. We show that this method is faster than the previous detection method, reducing the rate of false positives without sacrificing localization accuracy. The contributions of this paper are:

1. an AprilTag detection algorithm that improves detection rate for small tags, exhibits fewer false positives, and reduces computation time compared to the previous algorithm

2. a new tag boundary segmentation method that is respon- sible for many of the performance improvements, and could be applied to other fiducial detectors
3. an evaluation of the effect of fewer tag candidates on false positive rates
4. an experimental characterization of the localization performance of our detector on real and synthetic images.

2.9 Camera Calibration

Today's cheap pinhole cameras introduce a lot of distortion to images. Two major distortions are radial distortion and tangential distortion. Due to radial distortion, straight lines will appear curved. Its effect is more as we move away from the center of image. For example, one image is shown below, where two edges of a chess board are marked with red lines. But you can see that the border is not a straight line and doesn't match with the red line. All the expected straight lines are bulged out (Figure below).



This distortion is represented as follows:

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Similarly, another distortion is the tangential distortion which occurs because the image taking lense is not aligned perfectly parallel to the imaging plane. So some areas in image may look nearer than expected. It is represented as below:

$$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

In short, we need to find five parameters, known as distortion coefficients given by:

$$Distortion\ coefficients = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$

In addition to this, we need to find a few more information, like intrinsic and extrinsic parameters of a camera. Intrinsic parameters are specific to a camera. It includes information like focal length (fx, fy), optical centers (cx,cy) etc. It is also called camera matrix. It depends on the camera only, so once calculated, it can be stored for future purposes. It is expressed as a 3x3 matrix:

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Extrinsic parameters correspond to rotation and translation vectors which translate coordinates of a 3D point to a coordinate system.

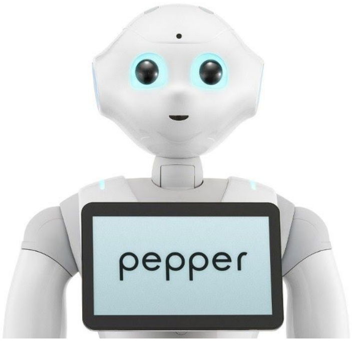
For stereo applications, these distortions need to be corrected first. To find all these parameters, what we have to do is to provide some sample images of a well defined pattern (eg, chess board). We find some specific points in it (square corners in chess board). We know its coordinates in real world space and we know its coordinates in image. With these data, some mathematical problem is solved in background to get the distortion coefficients. That is the summary of the whole story. For better results, we need at least 10 test patterns.

Chapter 3

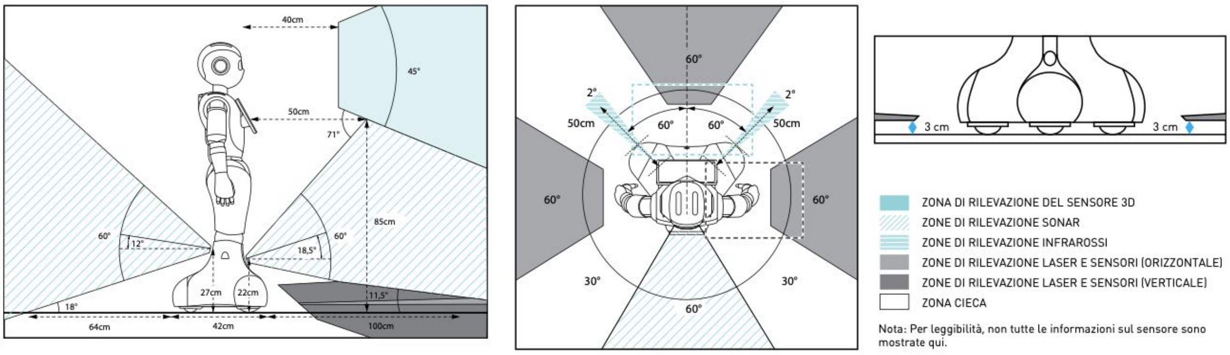
Tools and Software

3.1 Robot Pepper

In this chapter I will go into detail on the tools and software used.



As I mentioned before for this work I used the Pepper’s robot. Pepper is a humanoid developed by SoftBank Robotics. Pepper is the world’s first social humanoid robot able to recognize faces and basic human emotions. Pepper was optimized for human interaction and is able to engage with people through conversation and its touch screen.


40cm
45°
50cm
71°
85cm
60°
12°
60°
18,5°
27cm
22cm
11,5°
18°
64cm
42cm
100cm
60°
2°
2°
50cm
60°
60°
50cm
60°
60°
30°
30°
60°
3 cm
3 cm
ZONA DI RILEVAZIONE DEL SENSORE 3D
ZONE DI RILEVAZIONE SONAR
ZONE DI RILEVAZIONE INFRAROSSI
ZONE DI RILEVAZIONE LASER E SENSORI (ORIZZONTALE)
ZONE DI RILEVAZIONE LASER E SENSORI (VERTICALE)
ZONA CIECA
Nota: Per leggibilità, non tutte le informazioni sul sensore sono mostrate qui.

Pepper can detect objects around him using various built-in sensors. However, there are exceptional areas that Pepper cannot cover. Do not place any objects in those areas. Pepper was optimized for human interaction and is able to engage with people through conversation and his touch screen. It has 20 degrees of freedom for natural and expressive movements, speech recognition and dialogue available in 15 languages, perception modules to recognize and interact with the person talking to him, touch sensors, LEDs and microphones for multimodal interactions, Infrared sensors , bumpers, an inertial unit, 2D and 3D cameras, and sonars for omnidirectional and autonomous navigation and it is a fully programmable platform. At the laboratory of the Department of Computer, Automation and Management Engineering "Antonio Ruberti" there is the Pepper robot, this gave me the opportunity to put all this into practice.

3.2 MODIM and NaoQi

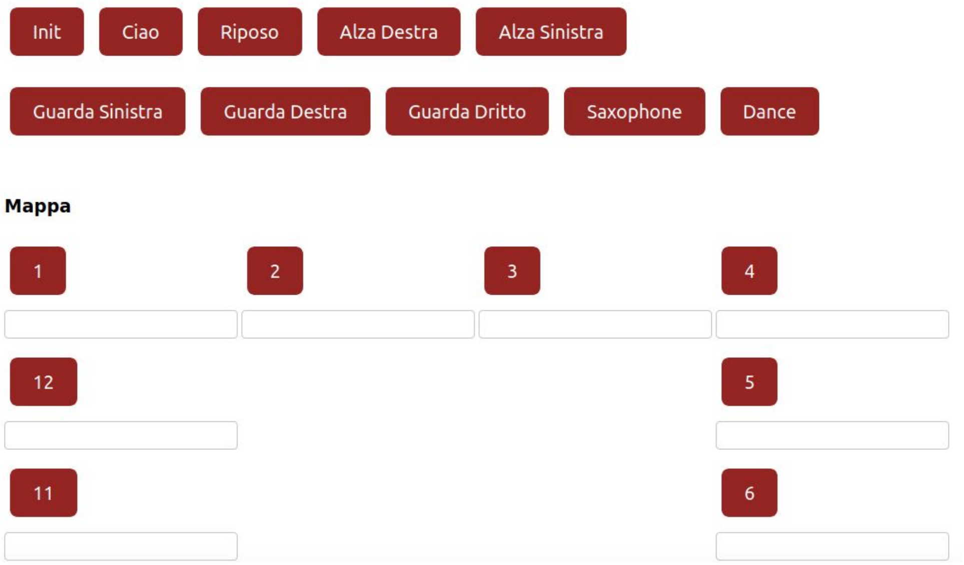
Due to the current Covid-19 situation it was not always easy to go to the laboratory to use the robot, the activity was facilitated by the MODIM tool (Multi-MO from the Interaction Manager) which was developed with the aim of being able to be easily used even by non-experts. MODIM is used in support of testing and use all functionalities we leveraged. Other simulations, without the physical robot, were made using pepper tools created by the professor that allowed me to simulate voice commands etc.

All of this was used in a docker container. Docker is an open-source project that automates the deployment of applications within software containers, providing an additional abstraction through Linux OS-level virtualization.

The Operating System of the robot is NAOqi OS, it is a GNU/Linux distribution based on Gentoo. It’s an embedded GNU/Linux distribution specifically developed to fit the SoftBank Robotics robot needs. It provides and runs numbers of programs and libraries, among these, all the required one by NAOqi, the piece of software giving life to the robot.

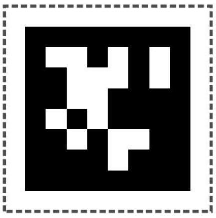
3.3 Joystick

The "joystick" was created using HTML / Javascript language. There are buttons, text fields etc that allow the doctor to decide what to do, say etc to the robot. We also used HTML language to open pages and to play with the design of the tablet in front of the robot: we use some colors, text gradient for text, and pictures to make the screen experience more engaging for the users. For the animation of the robot, we used the documentation provided by SoftBank Robotics.



3.4 AprilTag

AprilTag is a visual fiducial system, useful for a wide variety of tasks including augmented reality, robotics, and camera calibration. Targets can be created from an ordinary printer, and the AprilTag detection software computes the precise 3D position, orientation, and identity of the tags relative to the camera. The AprilTag library is implemented in C with no external dependencies. It is designed to be easily included in other applications, as well as be portable to embedded devices. Real-time performance can be achieved even on cell-phone grade processors. AprilTags are conceptually similar to QR Codes, in that they are a type of two-dimensional bar code. However, they are designed to encode far smaller data payloads (between 4 and 12 bits), allowing them to be detected more robustly and from longer ranges. Further, they are designed for high localization accuracy, you can compute the precise 3D position of the AprilTag with respect to the camera. In this work I used the tag36h11.



Tag36h11

The system is composed of two major components: the tag detector and the coding system. The detector whose job is to estimate the position of possible tags in an image. Loosely speaking, the detector attempts to find four-sided regions (“quads”) that have a darker interior than their exterior. The tags themselves have black and white borders in order to facilitate this.

3.4.1 Tag Detector

The detection process is comprised of several distinct phases:

a) Detecting Line Segments [6]

Their approach begins by detecting lines in the image. Their approach, similar in basic approach to the ARTag detector, computes the gradient direction and magnitude at every pixel and agglomerative clusters the pixels into components with similar gradient directions and magnitudes.

The clustering algorithm is similar to the graph-based method of Felzenszwalb [10]: a graph is created in which each node represents a pixel. Edges are added between adjacent pixels with an edge weight equal to the pixels' difference in gradient direction. These edges are then sorted and processed in terms of increasing edge weight: for each edge, they test whether the connected components that the pixels belong to should be joined together. Given a component n, we denote the range of gradient directions as $D(n)$ and the range of magnitudes as $M(n)$. Put another way, $D(n)$ and $M(n)$ are scalar values representing the difference between the maximum and minimum values of the gradient direction and magnitude respectively. In the case of $D()$, some care must be taken to handle 2π wrap-around. However, since useful edges will have a span of much less than π degrees, this is straightforward. Given two components n and m, they join them together if both of the conditions below are satisfied:

$$\begin{aligned} D(n \cup m) &\leq \min(D(n), D(m)) + K_D/|n \cup m| \quad (1) \\ M(n \cup m) &\leq \min(M(n), M(m)) + K_M/|n \cup m| \end{aligned}$$

The conditions are adapted from [10] and can be intuitively understood: small values of $D()$ and $M()$ indicate components with little intra-component variation. Two clusters are joined together if their union is about as uniform as the clusters taken individually. A modest increase in intra-component variation is permitted via the KD and KM parameters, however this rapidly shrinks as the components become larger. During early iterations, the K parameters essentially allow each component to "learn" its intra-cluster variation.

For performance reasons, the edge weights are quantized and stored as fixed-point numbers. This allows the edges to be sorted using a linear-time counting sort [20]. The actual merging operation can be efficiently carried out by the union-find algorithm [20] with the upper and lower bounds of gradient direction and magnitude stored in a simple array indexed by each component's representative member.

This gradient-based clustering method is sensitive to noise in the image: even modest amounts of noise will cause local gradient directions to vary, inhibiting the growth of the components. The solution to this problem is to low-pass filter the image [10]. Unlike other problem domains where this filtering can blur useful information in the image, the edges of a tag are intrinsically large-scale features (particularly in comparison to the data field), and so this filtering does not cause information loss. They recommend a value of $\sigma = 0.8$.

Once the clustering operation is complete, line segments are fit to each connected component using a traditional least-squares procedure, weighting each point by its gradient magnitude. They adjust each line segment so that the dark side of the line is on its left, and the light side is on its right. In the next phase of processing, this allows us to enforce a winding rule around each quad.

The segmentation algorithm is the slowest phase in our detection scheme. As an option, this segmentation can be performed at half the image resolution with a 4x improvement in speed. The sub-sampling operation can be efficiently combined with the recommended low-pass filter. The consequence of this optimization is a modestly reduced detection range, since very small quads may no longer be detected.

b) Quad detection

At this point, a set of directed line segments have been computed for an image. The next task is to find sequences of line segments that form a 4-sided shape, i.e., a quad. The challenge is to do this while being as robust as possible to occlusions and noise in the line segmentations.

Their approach is based on a recursive depth-first search with a depth of four: each level of the search tree adds an edge to the quad. At depth one, they consider all line segments. At depths two through four, they consider all of the line segments that begin "close enough" to where the previous line segment ended *and* which obey a counter-clockwise winding order. Robustness to occlusions and segmentation errors is handled by adjusting the "close enough" threshold: by making the threshold large, significant gaps around the edges can be handled. Their threshold for “close enough” is twice the length of the line plus five additional pixels. This is a large threshold which leads to a low false negative rate, but also results in a high false positive rate.

We populate a two-dimensional lookup table to accelerate queries for line segments that begin near a point in space. With this optimization, along with early rejection of candidate quads that do not obey the winding rule, or which use a segment more than once, the quad detection algorithm represents a small fraction of the total computational requirements.

Once four lines have been found, a candidate quad detection is created. The corners of this quad are the intersections of the lines that comprise it. Because the lines are fit using data from many pixels, these corner estimates are accurate to a small fraction of a pixel.

c) Homography and extrinsics estimation

They compute the 3×3 homography matrix that projects 2D points in homogeneous coordinates from the tag’s coordinate system (in which [0 0 1]T is at the center of the tag and the tag extends one unit in the xˆ and yˆ directions) to the 2D image coordinate system. The homography is computed using the Direct Linear Transform (DLT) algorithm [11]. Note that since the homography projects points in homogeneous coordinates, it is defined only up to scale.

Computation of the tag’s position and orientation requires additional information: the camera’s focal length and the physical size of the tag. The 3 × 3 homography matrix (computed by the DLT) can be written as the product of the 3 × 4 camera projection matrix P (which we assume is known) and the 4×3 truncated extrinsics matrix E. Extrinsics matrix are typically 4 × 4, but every position on the tag is at $z = 0$ in the tag’s coordinate system. Thus, they can rewrite every tag coordinate as a 2D homogeneous point with z implicitly zero, and remove the third column of the extrinsics matrix, forming the truncated extrinsics matrix. They represent the rotation components of P as Rij and the translation components as Tk. They also represent the unknown scale factor as s:

$$\begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} = sPE$$

$$= s \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{00} & R_{01} & T_x \\ R_{10} & R_{11} & T_y \\ R_{20} & R_{21} & T_z \\ 0 & 0 & 1 \end{bmatrix}$$

Note that we cannot directly solve for E because P is rank deficient. We can expand the right hand side of Eqn. 2, and write the expression for each hij as a set of simultaneous equations:

$$\begin{aligned} h_{00} &= sR_{00}f_x \\ h_{01} &= sR_{01}f_x \\ h_{02} &= sT_xf_x \\ &\dots \end{aligned}$$

These are all easily solved for the elements of Rij and Tk except for the unknown scale factor s. However, since the columns of a rotation matrix must all be of unit magnitude, we can constrain the magnitude of s. We have two columns of the rotation matrix, so we compute s as the geometric average of their magnitudes. The sign of s can be recovered by requiring that the tag appear in front of the camera, i.e., that $Tz < 0$. The third column of the rotation matrix can be recovered by computing the cross product of the two known columns, since the columns of a rotation matrix must be orthonormal.

The DLT procedure and the normalization procedure above do not guarantee that the rotation matrix is strictly orthonormal. To correct this, we compute the polar decomposition of R, which yields a proper rotation matrix while minimizing the Frobenius matrix norm of the error [12].

The *vpDetectorAprilTag* class inherits from vpDetectorBase class, a generic class dedicated to detection. For each detected tag, it allows retrieving some characteristics such as the tag id, and in the image, the polygon that contains the tag and corresponds to its 4 corner coordinates, the bounding box and the center of gravity of the tag. Moreover, *vpDetectorAprilTag* class allows estimating the

3D pose of the tag. To this end, the camera parameters as well as the size of the tag are required. Using this module I can identify all apriltags visible in an image, and get information about the location and orientation of the tags.

Chapter 4

My solution

4.1 Interaction Personalization

As I explained earlier, the start of my work is the TERESA project, it consists of a series of meetings with 10 patients suffering from obesity with two doctors, one of whom is a psychologist and Pepper the robot as assistant. The meetings are held on a monthly basis, exactly at the Dipartimento di Ingegneria Informatica, Automatica e Gestionale "Antonio Ruberti" in room B101. The patients sit in a circle (Figure 1).

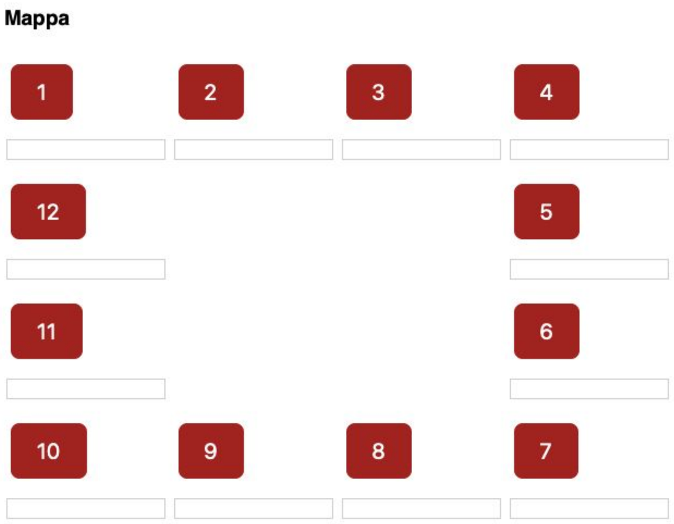


Figure 1

The psychotherapist doctor sits sideways with respect to the circle of patients' chairs as she needs the computer to use the "joystick" with the commands to give to the pepper robot. The so-called "joystick" is a web interface connected to the robot that allows the doctor to decide when to make it say something, show something on its tablet, where to look, etc.

At first the Pepper was in a fixed position in the room, exactly in one of the chairs between the patients. At the time of the doctor's commands, Pepper was addressing patients in general and looking left-right without real logic.

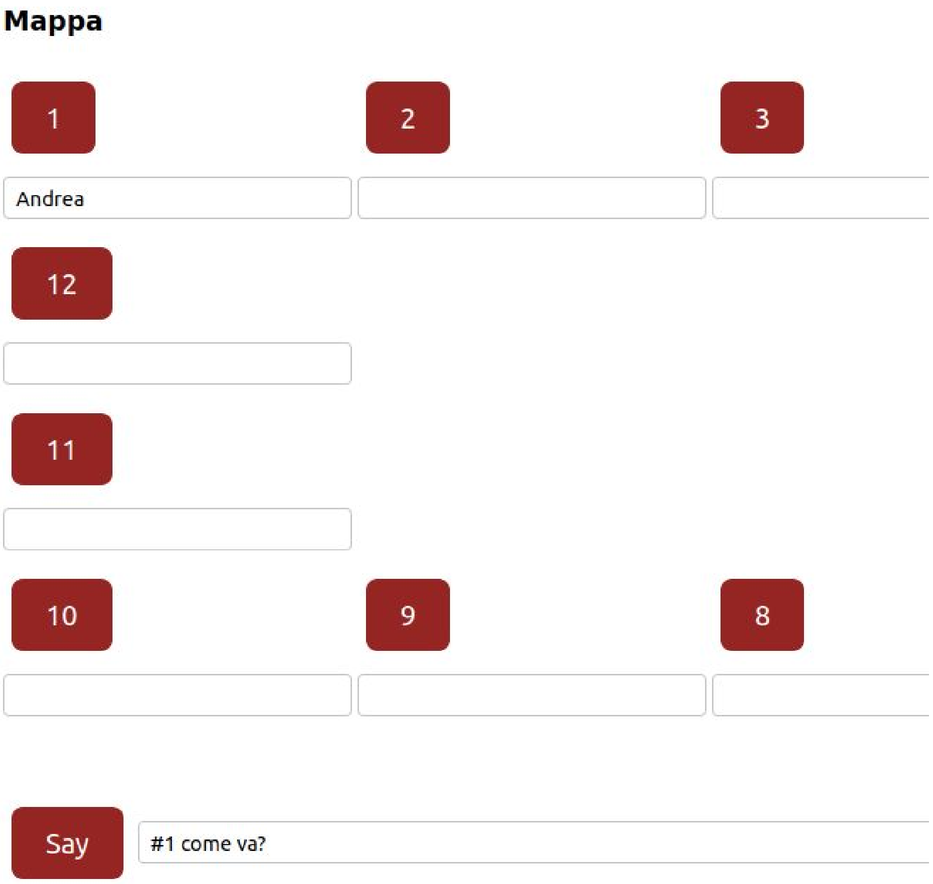
According to the doctors, this aspect turned out to be a problem, as they believed that a personalization of the interaction would be truly fundamental for the therapeutic path of these patients. The possibility of rendering a dialogue, a personalized robot-patient look could make the patient feel more "understood" and "in intimacy" with the robot.

The first part of my work was born from this concept: personalizing the robot-patient interaction.

First of all, as you can see in Figure 1, I extended the interface used by the doctor, the so-called "joystick", by adding the map of the twelve people present at each meeting. The main novelty is the possibility of labeling the place with the relative name of the person occupying it. This represents the first step for personalization.

The fact that at each meeting the doctor can assign a name to a certain place gives the possibility to develop a system whereby the robot is able to address a specific person, calling the person by the name and turning towards the person.

In practice, after having labeled the places with the names of the patients, the possibility of addressing the person calling him/her by name is performed as follows: in the "say" field #*number* will be inserted with the number corresponding to the person to whom you want to address, the program will recognize that seat #*number* corresponds to a certain person and then replace #*number* with the name corresponding to the seat where the chosen person is sitting.



Mappa

1: Andrea

Say: #1 come va?

Fig. 2

As you can see from Fig.2, the seat corresponding to chair one is labeled with the name "Andrea", when the doctor writes "# 1 how are you?", Pepper will address the person saying "Andrea how are you?" (Fig. 3).

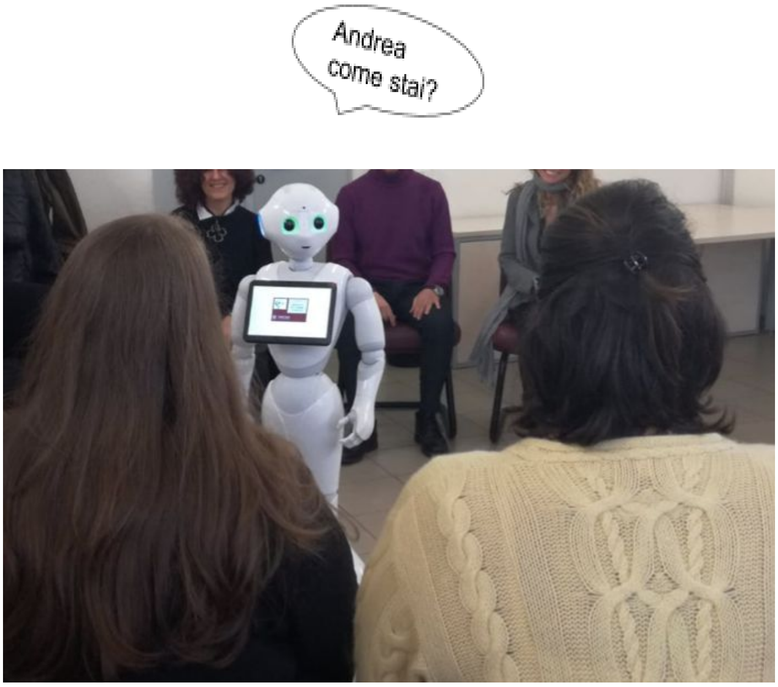
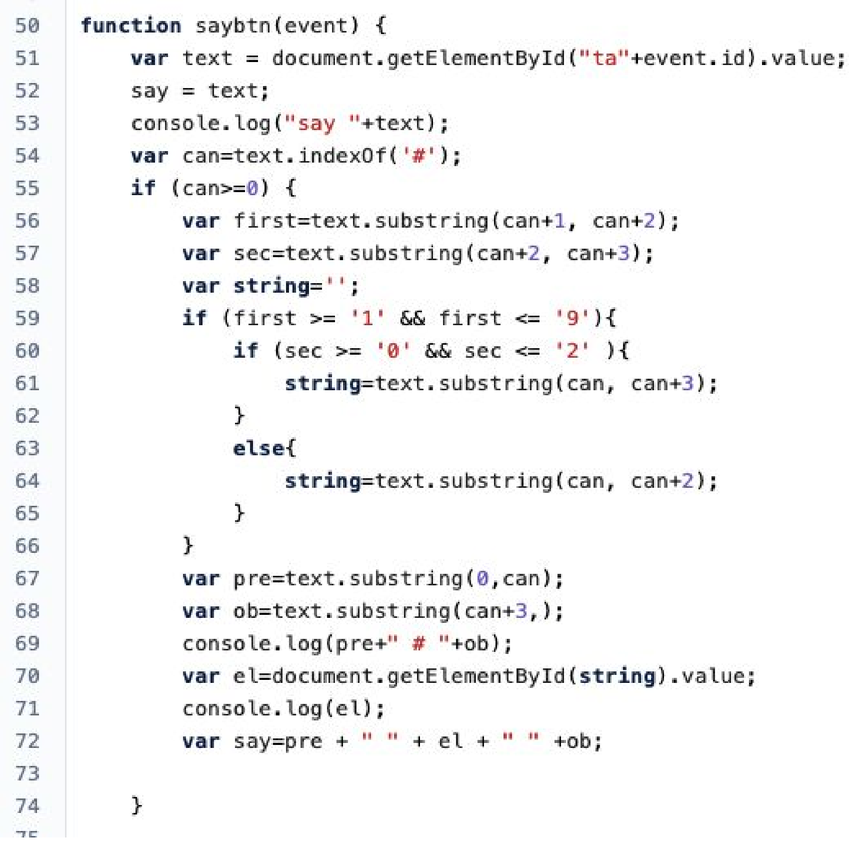


Fig. 3

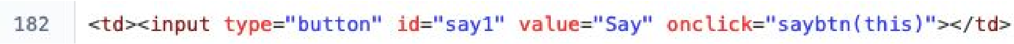
This function *saybtn* can be found in the file teresa.js:



```
function saybtn(event) {
    var text = document.getElementById("ta"+event.id).value;
    say = text;
    console.log("say "+text);
    var can=text.indexOf('#');
    if (can>=0) {
        var first=text.substring(can+1, can+2);
        var sec=text.substring(can+2, can+3);
        var string='';
        if (first >= '1' && first <= '9'){
            if (sec >= '0' && sec <= '2' ){
                string=text.substring(can, can+3);
            }
            else{
                string=text.substring(can, can+2);
            }
        }
        var pre=text.substring(0,can);
        var ob=text.substring(can+3,);
        console.log(pre+" # "+ob);
        var el=document.getElementById(string).value;
        console.log(el);
        var say=pre + " " + el + " " +ob;

    }
```

it is call back in the *html* file in this way:



```
<td><input type="button" id="say1" value="Say" onclick="saybtn(this)"></td>
```

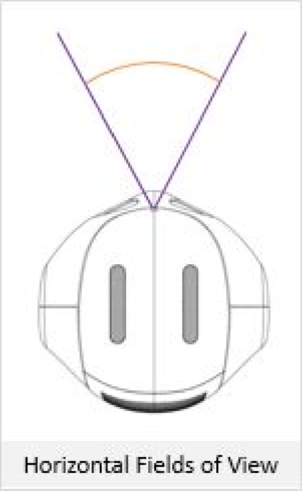
This addition, as well as personalizing the interaction with the patient, allows the doctor to call attention to patients who may appear distracted. I also added the possibility to move toward the chosen person of 1 m.

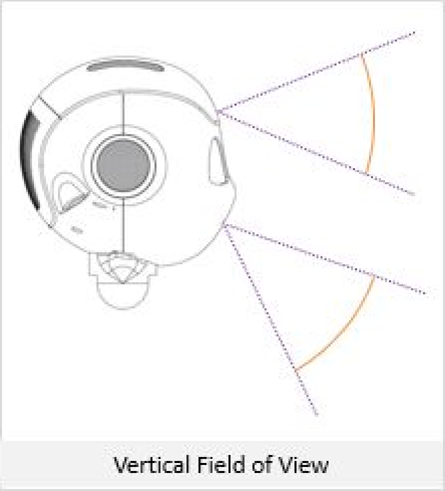
4.2 Interaction Personalization: tags and localization

The second step of my project, in my opinion the most interesting part, concerns the movement of the robot.

As previously mentioned, this part of the work is slightly detached from the TERESA's project, although the experiments will be performed by simulating the environment and the situation of the encounters, it can easily be seen that it can be extended to any other type of setting and robot.

It is important to specify that Pepper has 2D chambers, one front and one under the chin.





In particular, we will use the chamber under the chin to take samples for the various experiments (example in Fig. 5).

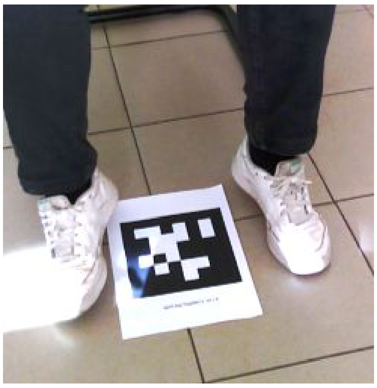
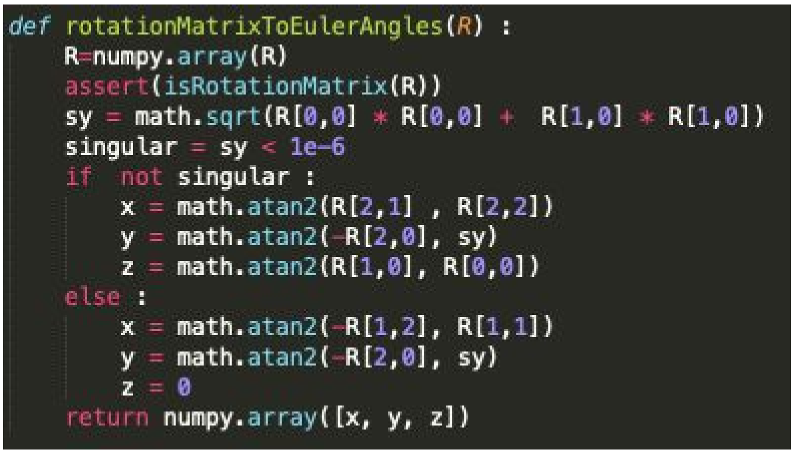


Fig. 5

The tag we will use for this work is Tag36h11, each tag have its own ID.
The python program *apriltag.py* allows to recognize the tags present in the various images, the corresponding ID, the position, position of the corners, the center and the homography.
The program allows you to choose an ID and have the data match the requested tag.



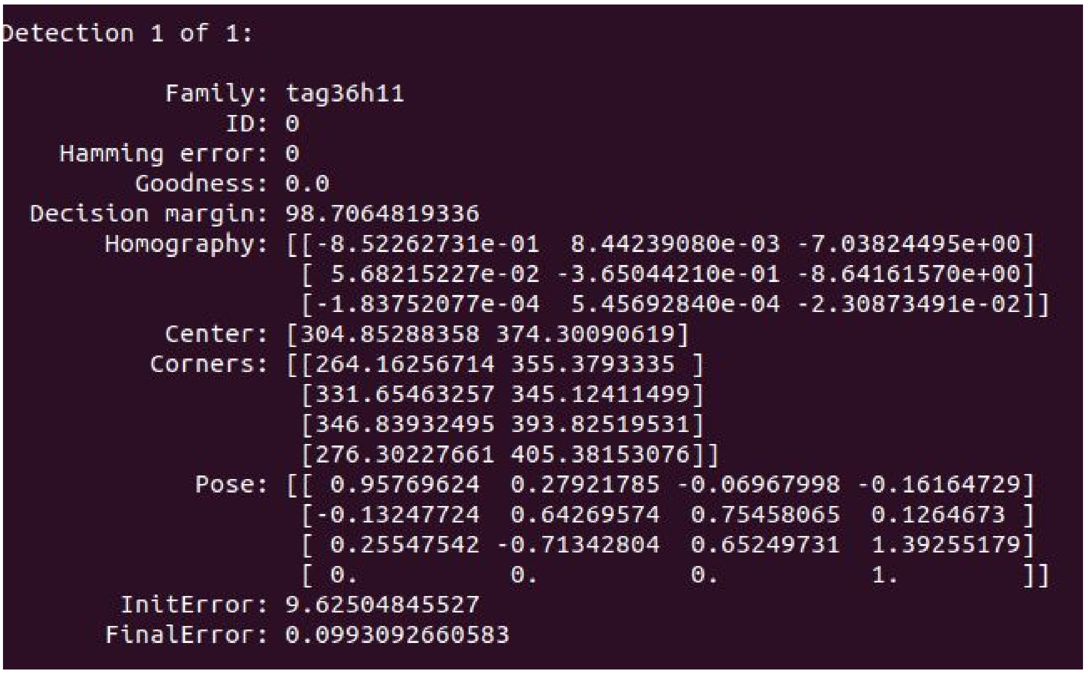
```
def rotationMatrixToEulerAngles(R) :
    R=numpy.array(R)
    assert(isRotationMatrix(R))
    sy = math.sqrt(R[0,0] * R[0,0] +  R[1,0] * R[1,0])
    singular = sy < 1e-6
    if  not singular :
        x = math.atan2(R[2,1] , R[2,2])
        y = math.atan2(-R[2,0], sy)
        z = math.atan2(R[1,0], R[0,0])
    else :
        x = math.atan2(-R[1,2], R[1,1])
        y = math.atan2(-R[2,0], sy)
        z = 0
    return numpy.array([x, y, z])
```

Figure 6

The transformation in Figure 6 gives us the possibility to obtain the angle corresponding to the required ID corresponding to the camera. The values that I found are the roll (x-axis rotation), pitch (y-axis rotation) and yaw (z-axis rotation), Euler's angles.

The tags are placed in front of each chair, in ascending order of ID starting from position one on the map.

For example, when the program detects the tag in the photo, the output is as follows:



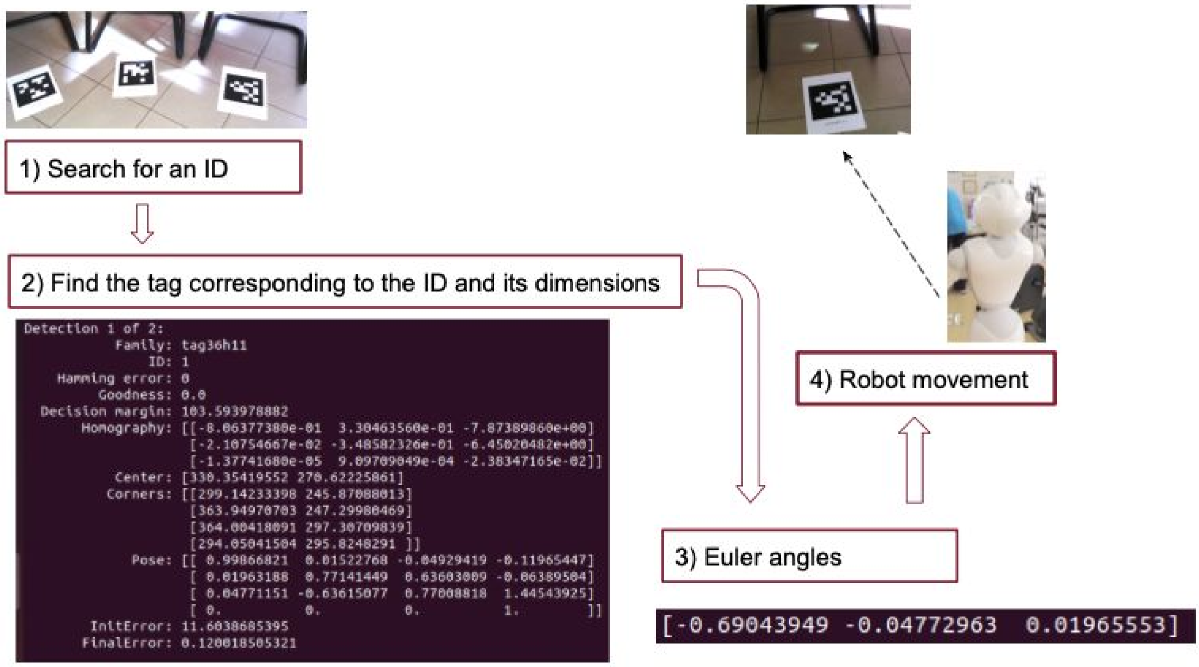
```
Detection 1 of 1:

          Family: tag36h11
              ID: 0
   Hamming error: 0
        Goodness: 0.0
 Decision margin: 98.7064819336
      Homography: [[-8.52262731e-01  8.44239080e-03 -7.03824495e+00]
                   [ 5.68215227e-02 -3.65044210e-01 -8.64161570e+00]
                   [-1.83752077e-04  5.45692840e-04 -2.30873491e-02]]
          Center: [304.85288358 374.30090619]
         Corners: [[264.16256714 355.3793335 ]
                   [331.65463257 345.12411499]
                   [346.83932495 393.82519531]
                   [276.30227661 405.38153076]]
            Pose: [[ 0.95769624  0.27921785 -0.06967998 -0.16164729]
                   [-0.13247724  0.64269574  0.75458065  0.1264673 ]
                   [ 0.25547542 -0.71342804  0.65249731  1.39255179]
                   [ 0.          0.          0.          1.        ]]
       InitError: 9.62504845527
      FinalError: 0.0993092660583
```

Figure 7

So the program gives us the position of the tag, the center, the initial and final error ecc.

The main idea is represented by the schema above (the “if part”):



```
Detection 1 of 2:
           Family: tag36h11
               ID: 1
    Hamming error: 0
         Goodness: 0.0
  Decision margin: 103.593978882
       Homography: [[-8.06377380e-01  3.30463560e-01 -7.87389860e+00]
                    [-2.10754667e-02 -3.48582326e-01 -6.45020482e+00]
                    [-1.37741680e-05  9.09709049e-04 -2.38347165e-02]]
           Center: [330.35419552 270.62225861]
          Corners: [[299.14233398 245.87088013]
                    [363.94970703 247.29980469]
                    [364.00418091 297.30709839]
                    [294.05041504 295.8248291 ]]
             Pose: [[ 0.99866821  0.01522768 -0.04929419 -0.11965447]
                    [ 0.01963188  0.77141449  0.63603009 -0.06389504]
                    [ 0.04771151 -0.63615077  0.77008818  1.44543925]
                    [ 0.          0.          0.          1.        ]]
        InitError: 11.6038685395
       FinalError: 0.120018505321
```

```
[-0.69043949 -0.04772963  0.01965553]
```

Figure 4

The first step of the detection is represented by the schema above: if the robot detects the tag corresponding to the required ID, it remains in front of the chosen chair and goes toward it. The output of the program is as in Figure 7.

The “else part” is represented in Figure 8. If the required ID is not in the visual area, since the tags are in ascending order, the robot will still be able to understand which way to turn to find the required ID (we assume that each chairs’ distance is 30 degrees). For example: Pepper detects three tags with ID = 2, ID = 3 and ID = 4, if the required ID is the number 8 the robot will be able to understand that to find it it will have to turn to the right, not to the left, in particular it must rotate by 30 degrees multiplied by 8-4, the positions that separate the two places. Scrolling through the various tags to the right, you will find the required one.

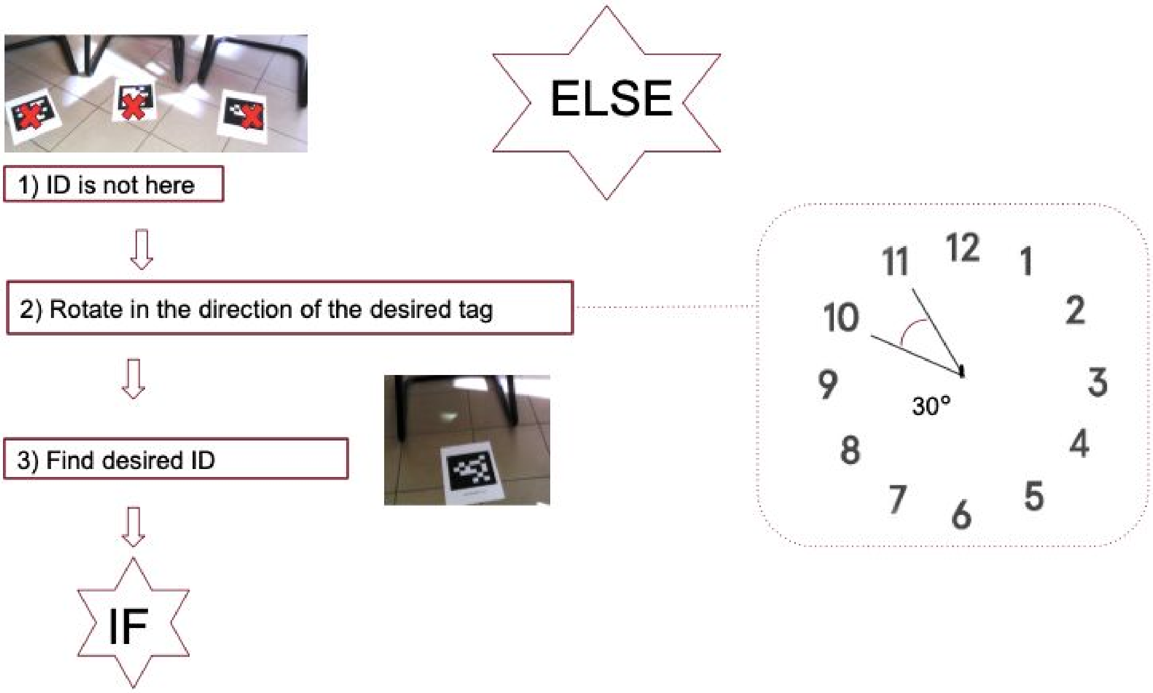


Figure 8

Obviously in the real meetings of the TERESA's project the IDs will correspond to the names of the patients. In this part I have explained using the IDs to underline the generality of this approach which goes outside the TERESA project.

Chapter 5

Robot-Patient Interaction

5.1 Experiments and results

To reproduce the situation of the meetings as in the TERESA's project, I placed 4/5 chairs, about 30 degrees apart. Unfortunately, due to Covid-19, it was not possible to carry out everything with real patients, in fact, as can be seen from

Figure 9, the chairs are empty. At the foot of each chair there is a tag, precisely from 1 to 12 in hourly order. First of all, I reproduced the moment of the personalized interaction in which Pepper addresses the individual patient, calling him/her by name and asking something. To this, during the simulation phase, I added the rotation of the head and I made the robot move towards the person in question to make the interaction even more "intimate".

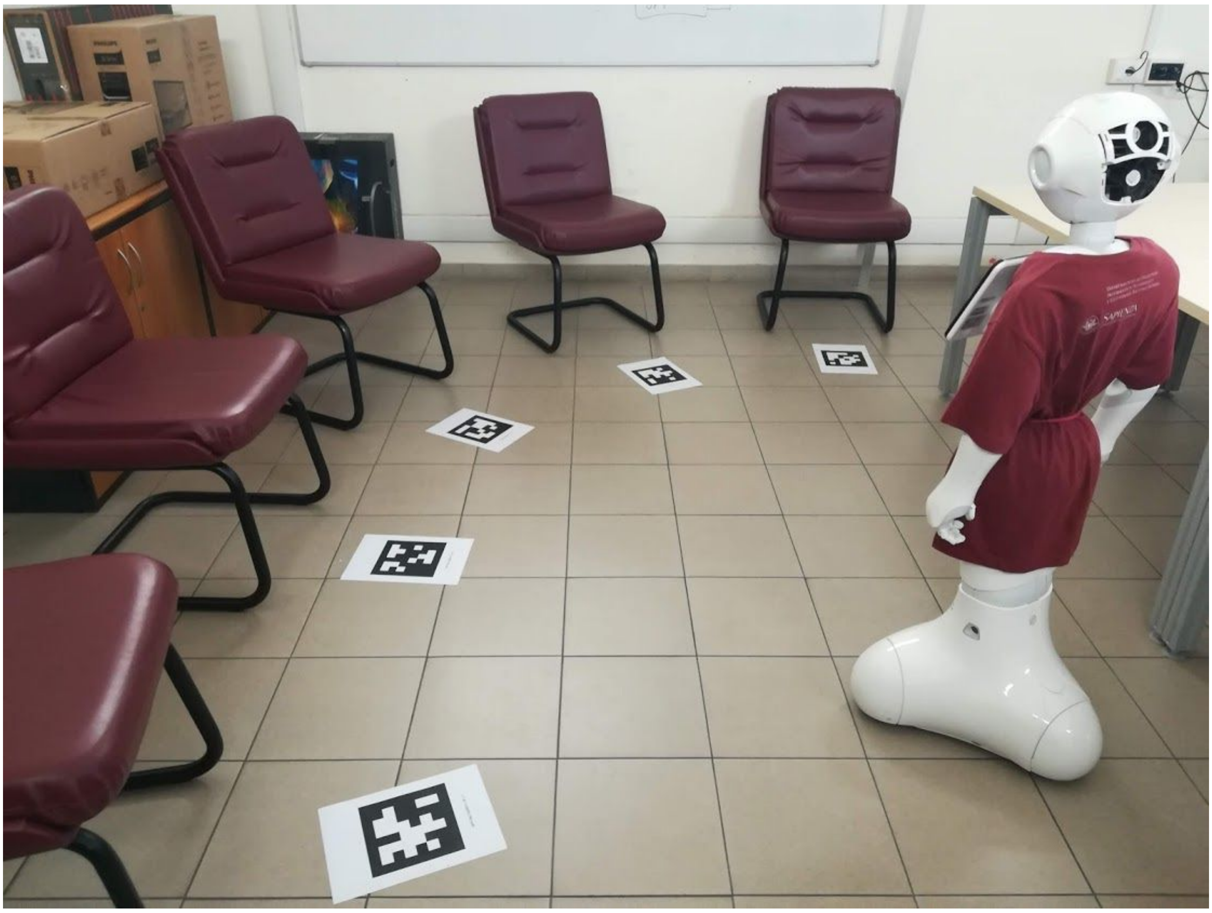


Figure 9

Pepper was placed in the center of the chairs.

When I wrote in the Say field "# 1, how are you?" the system understood that in place 1 of the map there was the name Andrea and Pepper immediately pronounced "Andrea, how are you?". As mentioned earlier, I added the rotation of the head towards the person in question and the advancement of half a meter towards him. I also tried that the system worked for stations corresponding to double digit numbers or sentences with *#number* in between, not just at the beginning or end of the sentence. Also in these two cases the feedback was positive as Anna was sitting in chair 12 for my map and when I clicked on the say "hello # 12, what are you saying?", Pepper said "Hi Anna, what are you saying?", also in this case by turning to her.

As for the part of the movement of the robot through the tags, I have done many experiments. The posts were positioned approximately 30 degrees apart each, each with a tag in front. When the camera under the robot's chin captures the image in front of it, the program detects the data, ID, position, etc. of the tag. The basic idea was to understand if Pepper behaved in the right way in front of these tags, that is: once the user requests a specific ID if the robot is in front of it it must remain stationary while if not find must rotate tot degrees to find it. I have adjusted the program so that it can calculate the magnitude of the distance between the ID of the tag found and the ID of the requested tag, obviously multiplied by 30 degrees (distance between the positions). If the robot detects ID = 2, for example, but needs to turn towards ID = 5, it will have to rotate 5-2 = 3 multiplied by 30 degrees, then 90 degrees. First of all I had to fix the image program where more tags are captured. First of all I had to fix the image program where more tags are captured. This is because to find the distance between the ID requested and the ID found, in the presence of two tags in the photo it initially gave me problems because the system took one of the two tags without logic. I fixed this by adding the value of the minimum distance, in order to make the program take the tag closest to the requested ID.

Also for these experiments I placed 5 chairs that form a semicircle (Figure 10) to reproduce the situation of TERESA's meetings.

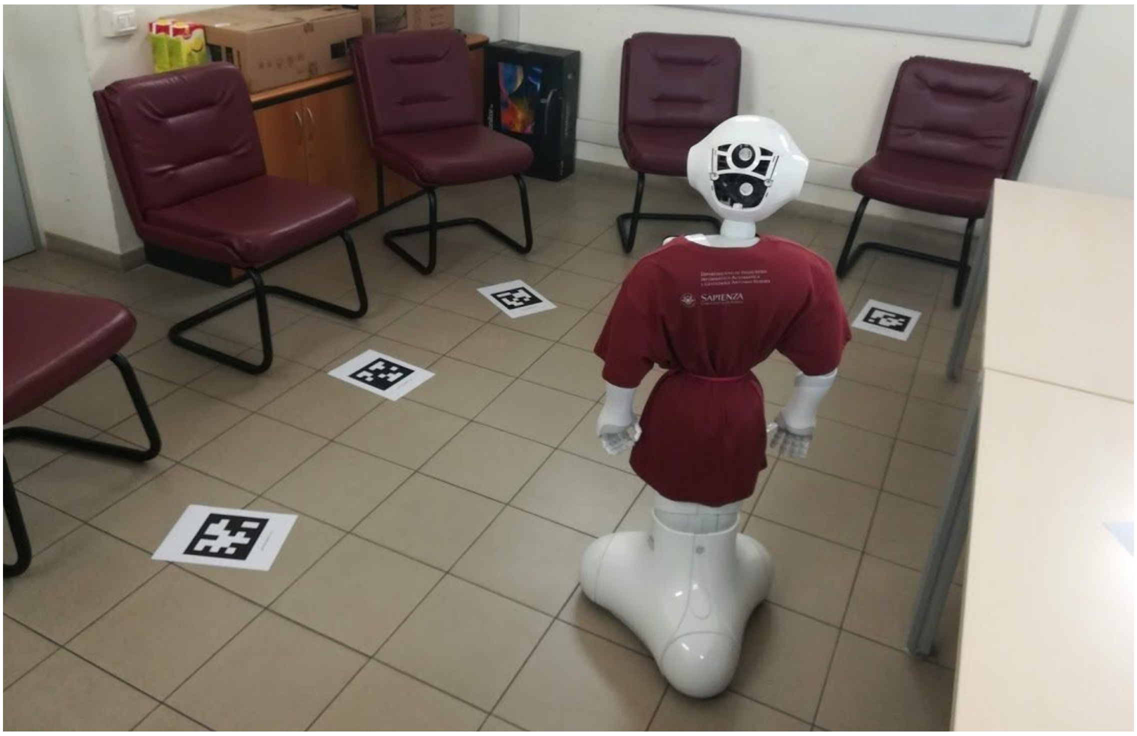


Figure 10

The distance between each chair is about 30 degrees, initially positioned by eye and later, thanks to the robot, I moved them in order to separate them by 30 degrees precisely.

The tags placed are from ID 1 to ID 5, in chronological order. The robot was placed in the center at a distance of about 1.5m from each chair, when I requested ID 1 Pepper, being already in front of the requested tag, it did not need to turn around and said "ID 1 found ", after which the robot approached the requested position by about 1 m. As for the request for an ID that is not in the robot's visual range, the robot responded well to the rotation to get to the requested tag when asked for an ID. Once the ID is found, it returns the tag data as it does when it is in front of the ID right away. This type of experiment works both when IDs are requested that are after (numerically) those in the camera's visual range, and when IDs preceding the current ones are requested.

To reproduce the situation as real as possible, I wrote in the field say "# 5, how are you" ?, so the system recognized that I was referring to seat 5 where Luca is sitting. In the same way, place 5 corresponds to ID 5, so TERESA, in front of ID 1 and 2, searches for ID 5. ID=2 corresponds to Maria’s chair, so TERESA says “I’m searching for Luca”. After turning 90 degrees, it finds ID 5, that is Luca. After finding it, TERESA says what is written in the say field of the joystick. (All these experiments can be found on the video presentation of my thesis work on my youtube channel: https://youtu.be/JZgT9YgyBQQ).

Chapter 6

Robustness of the detection's system

After several tests we can say that tag detection works perfectly with clearly visible tags. In this chapter I wanted to analyze the robustness of this system, if it also works with partial occlusions and with light changes. In particular I will show that by inserting disturbances the performance will regress. Everything will be explained through the use of histograms. Specifically, I will show two main experiments: the first as regards partial occlusions and the second as regards the change of lights. I also created a program to plot the results by histogram but I didn't insert them because the results were not significant.

6.1 Partial occlusions experiments and results

As a first experiment I tried to partially conceal the tags, by placing a person's shoes on the tag (Figure 11). As expected, the results are really bad. Only the detection of the first tag gives good results when the person does not touch the tag. The system is very sensitive to even the slightest disturbance. I created a system whereby the detections are arranged in ascending order with respect to the final error. When the noise increases, the final error also increases.

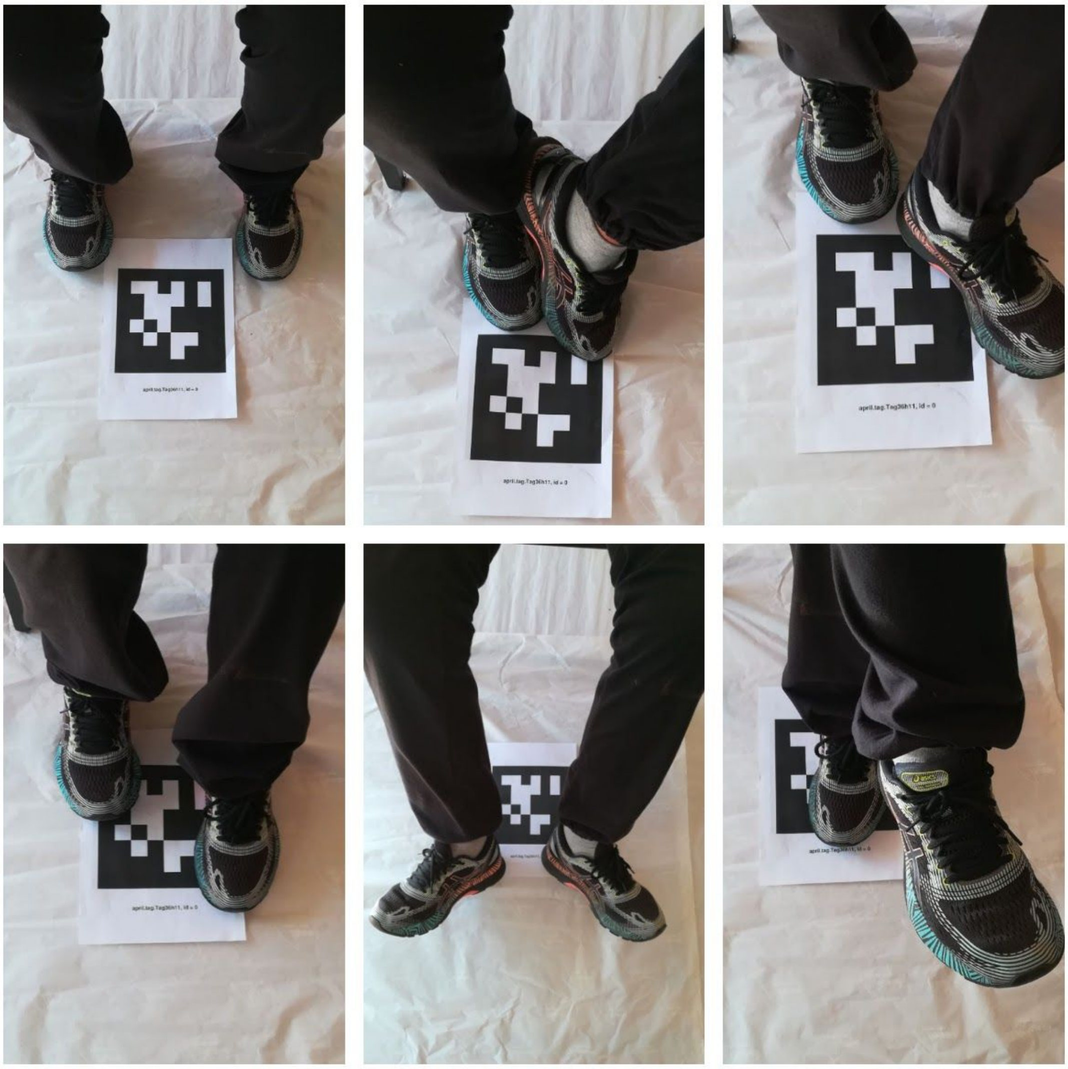


Figure 11

The final error is low for the first image where the tag is clearly visible, while the other tags in the presence of minimal noise are not detected. The final error indicates the efficiency of the detection.

These poor results were expected as I had already been able to verify with the movement of the robot that the system was really sensitive to this type of disturbance.

6.2 Light changes experiments and results

The second type of experiment is executed leveraging lighting changes; as you can see from figure 12 I tried to illuminate the tag with a red light torch.

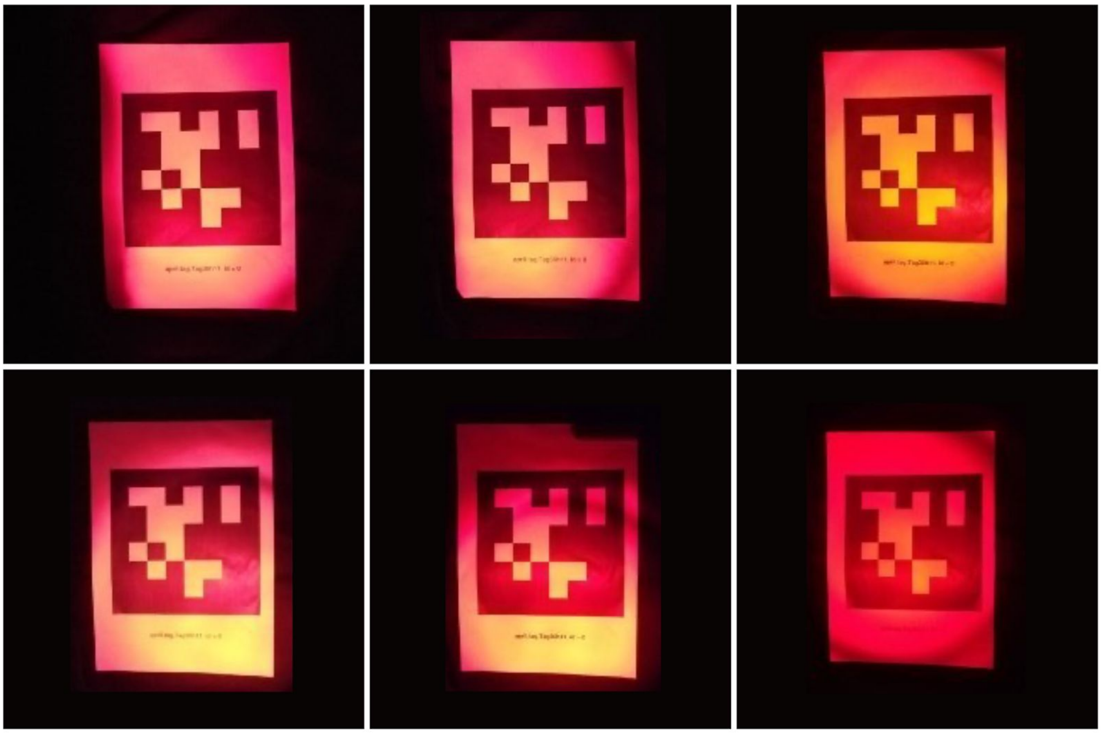


Figure 12

The results of this experiment are better than the previous experiment. Three tags are now detected out of six.

I replicated the same experiment using a standard flashlight, Figure 13. The results of this test are as bad as the shoe experiment. My deduction is the following: the fact that the light creates "black/white shadows does not allow the system to distinguish the parts of the tag and those from the noise. This does not happen with the red light because the colour is different.

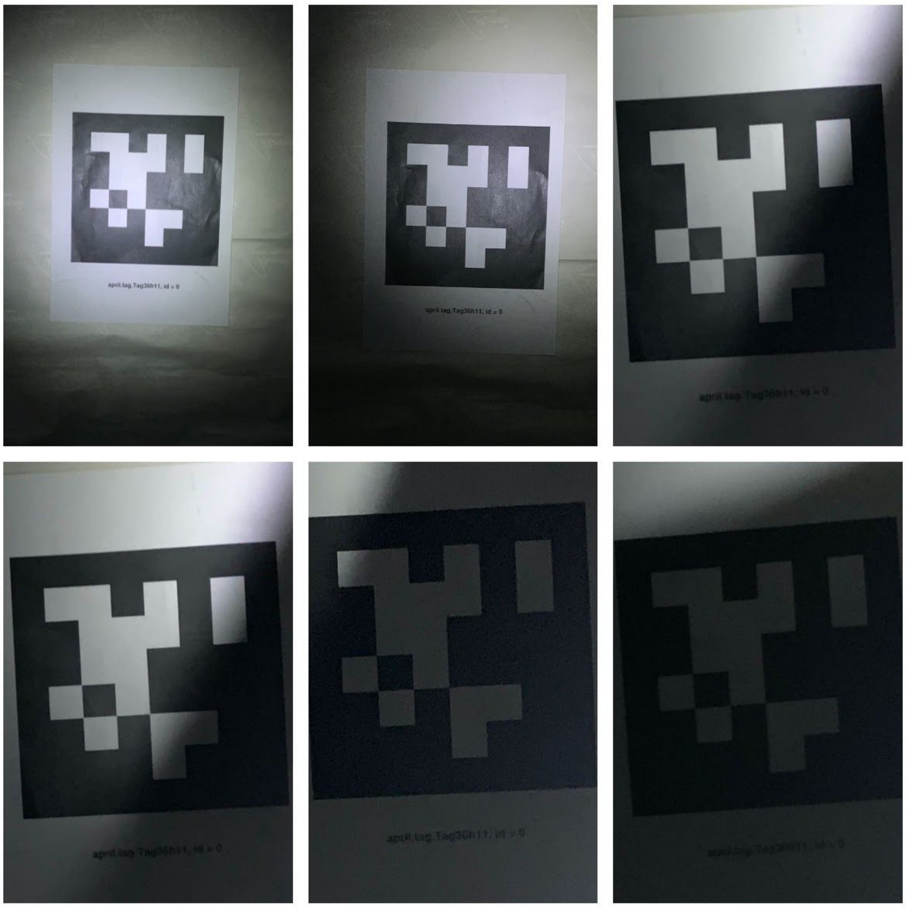

april.tag.Tag36h11, id = 0
april.tag.Tag36h11, id = 0
april.tag.Tag36h11, id = 0

Figure 13

These experiments show that the system is efficient in optimal situations, while this immediately degrades in the presence of minimal disturbances.

In this case, the Human-Robot interaction can solve the problem of the not-very-robust system; indeed Pepper faced a tag that is not clearly visible; for example, in case the shoes of the person sitting on the chair covered the tag, the robot could suggest the user to move the feet away and solve the problem.

Chapter 7

Conclusions

With the growing deployment of robots in different aspects of daily life, it has been fascinating to study and extend the leverage of a robot as a medical assistant. This type of collaboration will improve the future while it is still in its preliminary phase with many aspects to study and improve. I chose this as the subject of my thesis as I immediately found the potential with the Pepper robot very promising. Thanks to Professor Iocchi, who followed me through my thesis preparation, I was able to access the physical robot available in our university department. It was fascinating to see how ignoring the programming behind it, the robot can interact in a very human manner, in its movements and interactions. Choosing this thesis project was stimulating. Step by step, I discovered and learned new things from very different points of view that brought me to have a holistic understanding of how to apply robotics. After hours spent testing Pepper, it was stimulating to have a positive response after the performance of the experiments. Initially, it was difficult to understand the sensitivity of the robot - how it responds to specific commands and how to assess and understand the margin of error in its rotation and movement. For the TERESA project, I believe that the most critical aspect was the personalization of the interaction with the patient and how to observe and improve this further. I slowly understood how it works and what are the specific dimensions to monitor, this with iteration to the point of reaching the expected result in the movement of the robot. I was able to make the robot move expectedly, through the use of tags, to manage rotation to desired positions based on rotation angles. When months ago, I started working on this project; I always imagined the day I would have been able to attend a meeting where TERESA would be deployed in support of the educational program. Unfortunately, due to the pandemic situation, this was not possible, but simulating the meeting, allowed me to verify the validity of the work done. As for chapter Six, the study of robustness has given negative results; the detection system is not able to work in front of noise and disturbances. The disturbances such as occlusion of the tag have almost non-existent effects as the detection does not detect the tag. In the same way, as the change of light using, for example, a normal torch. Slightly better results are obtained with changing light with the illumination of a red flashlight. This made me reflect on the fact that this is caused by the black and white shadows that confuse the system. Thanks to HRI, it is also possible to solve the problem of system robustness. It was stimulating to work on this project, and I hope to have the opportunity to continue it outside my academic path. I asked to Dr. Enrico Prosper how he noticed that TERESA had made a difference during the meetings. The doctor replied: "With the presentation of TERESA during the first meeting, those barriers of shyness and fear of making mistakes that can be present in a new situation in which no one knows each other have been broken down.The questions posed by the robot during the beginning of the sessions, to know what individual patients had experienced during the days that separated the individual meetings, may have been experienced with less anxiety than feeling "questioned" by the doctor. TERESA certainly made the meetings more fun, through personal examples that had obviously been previously defined by the health leaders, and games that actively involved the participants. This can reduce the embarrassment that often hinders the active participation of all patients. Simple games such as "True or false" or "Guess who" allowed to observe the knowledge acquired or already possessed by the group participants and at the same time allowed the conductors to deepen some topics of the sessions".

7.1 Future works

It will be perfect to be able to put this work into practice during the sessions held in the TERESA project. It would be nice to continue working on this project even outside the university path and to be able to integrate the personalized robot-patient interaction with the movement of the robot through tags. In particular, the current work could be extended with the correspondence between ID and name of the real seated person. This for TERESA to search for the real person instead of the ID. I very much hope that this global emergency will end soon so that we can attend one of the project meetings and observe these improvements implemented. I would like to be able to extend this to situations where there are obstacles in the path of the robot and see how it is possible to make the process flawless. It would be interesting to extend the work on robot movement through tags also outside the context of the TERESA project. For example, by positioning the tags in a different way. I hope all of this would be put into practice in the shorter term and improve the experience in the TERESA educational meetings.

References

[1] *Paola Ferrarelli, Mar´ıa T. L´azaro, and Luca Iocchi* Design of robot teaching assistants through multi-modal human-robot interactions

[2] *Luca Iocchi, Maria Teresa L´azaro, Laurent Jeanpierre, Abdel-Illah Mouaddib*
Personalized short-term multi-modal interaction for social robots assisting users in shopping malls. Dept. of Computer, Control and Management Engineering Sapienza University of Rome, Italy.GREYC, University of Caen Lower-Normandy, France

[3] *Pennisi, Andrea; Sahli, Hichem; Jeanpierre,, Laurent; Mouaddib, Abdel-Illah; Iocchi, Luca; Lazaro, Maria Teresa; Erdem, Esra; Demirel, Ezgi; Patoglu, Volkan.* COACHES: An assistance Multi-Robot System in public areas

[4] *Enrico Prosperi, Giada Guidi, Lucio Gnessi, Luca Iocchi.* TERESA: robot sociale per l'assistenza terapeutica. Società Italiana di Educazione Terapeutica, Roma, Italy Dip. di Medicina Sperimentale, Sapienza Università di Roma, Italy. Dip. di Ingegneria Informatica Automatica e Gestionale, Sapienza Università di Roma, Italy

[5] *Peter I. Corke.* VISUAL CONTROL OF ROBOTS: High-Performance Visual Servoing. CSIRO Division of Manufacturing Technology, Australia.

[6] *Edwin Olson*. AprilTag: A robust and flexible visual fiducial system. University of Michigan

[7] *John Wang and Edwin Olson.* AprilTag 2: Efficient and robust fiducial detection.

[8] *Luca Iocchi, Dirk Holz, Javier Ruiz-del-Solar, Komei Sugiura, Tijn van der Zant:* Analysis and results of evolving competitions for domestic and service robots.

[9] *Satya Mallick.* Computer Vision Resources

[10] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based im- age segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[12] K. Shoemake and T. Duff, “Matrix animation and polar decomposi- tion,” in *In Proceedings of the conference on Graphics interface 92*. Morgan Kaufmann Publishers Inc, 1992, pp. 258–264.

[13] *Robotics Modelling Planning and Control* B.Siciliano