Planification et contrôle de mouvements en interaction avec l'homme. Reasoning about space for human-robot interaction

Luis Felipe Marin-Urias

Université Toulouse III - Paul Sabatier
École Doctorale Systèmes

Thèse

en vue de l'obtention du

Doctorat de l'Université de Toulouse
délivré par l'Université Toulouse III - Paul Sabatier

Spécialité: Systèmes Informatiques

Présentée et soutenue publiquement par:

Luis Felipe Marín Urías

le 12 Novembre 2009

Reasoning About Space for Human-Robot Interaction

Préparée au Laboratoire d'Analyse et d'Architecture des Systèmes
sous la direction de:

M. Rachid ALAMI

Jury

M. Philippe PALANQUE	Président
M. Michael BEETZ	Rapporteur
M. Philippe FRAISSE	Rapporteur
M. Chris MELHUISH	Examinateur
M. Jean-Paul LAUMOND	Examinateur

Abstract:

Human Robot Interaction is a research area that is growing exponentially in last years. This fact brings new challenges to the robot's geometric reasoning and space sharing abilities. The robot should not only reason on its own capacities but also consider the actual situation by looking from human's eyes, thus "putting itself into human's perspective".

In humans, the "visual perspective taking" ability begins to appear by 24 months of age and is used to determine if another person can see an object or not. The implementation of this kind of social abilities will improve the robot's cognitive capabilities and will help the robot to perform a better interaction with human beings.

In this work, we present a geometric spatial reasoning mechanism that employs psychological concepts of "perspective taking" and "mental rotation" in two general frameworks:

- Motion planning for human-robot interaction: where the robot uses "egocentric perspective taking" to evaluate several configurations where the robot is able to perform different tasks of interaction.

- A face-to-face human-robot interaction: where the robot uses perspective taking of the human as a geometric tool to understand the human attention and intention in order to perform cooperative tasks.

Résumé:

L'interaction Homme-Robot est un domaine de recherche qui se développe de manière exponentielle durant ces dernières années, ceci nous procure de nouveaux défis au raisonnement géométrique du robot et au partage d'espace. Le robot pour accomplir une tâche, doit non seulement raisonner sur ses propres capacités, mais également prendre en considération la perception humaine, c'est à dire "Le robot doit se placer du point de vue de l'humain".

Chez l'homme, la capacité de prise de perspective visuelle commence à se manifester à partir du 24ème mois. Cette capacité est utilisée pour déterminer si une autre personne peut voir un objet ou pas. La mise en place de ce genre de capacités sociales améliorera les capacités cognitives du robot et aidera le robot pour une meilleure interaction avec les hommes.

Dans ce travail, nous présentons un mécanisme de raisonnement spatial de point de vue géométrique qui utilise des concepts psychologiques de la "prise de perspective" et "de la rotation mentale" dans deux cadres généraux:

- La planification de mouvement pour l'interaction homme-robot: le robot utilise "la prise de perspective égocentrique" pour évaluer plusieurs configurations où le robot peut effectuer différentes tâches d'interaction.

- Une interaction face à face entre l'homme et le robot : le robot emploie la prise de point de vue de l'humain comme un outil géométrique pour comprendre l'attention et l'intention humaine afin d'effectuer des tâches coopératives.

Acknowledgements

The list of people that have, directly or indirectly, contributed to this work is very extensive, and my memory is very short. So I hope not to forget anybody while I'm writing these acknowledgements.

First of all, I want to thank to my thesis advisor, Rachid Alami, for being such a great "coach" and friend, giving me not only good direction but also the necessary motivation and support to continue trough all these years.

I would like to extend my gratitude to Michael Beetz and Philippe Fraisse for taking some of their so constrained time to read and analyze my thesis; as also to Chris Melhuish, Jean-Paul Laumond and Philippe Palanque for being part of my jury and for their valuable comments.

All this work couldn't have been done if I wouldn't count on my friends and co-workers from different projects Samir, Riichiro, Xavier, Brice, Mathias, Raquel, Jean-Phillipe, Matthieu W., Thierry G. and very very special gratitude goes to Akin and Amit (and Sabita) which whom I have passed difficult but also very cool moments.

Also thanks for those that finished and have gone before me: Nacho, Thierry, Eduard, Claudia, Tere, Gustavo, Efrain, Felipe, Panos, Aurelie and Gaelle for all the moments, confessions and discussions held. The same for those who are still there Moky, Matthieu P., Ossama, Naveed, Layale, Francisco, Mario, Dora, Jorge, Diego, Manish, Bertrand, Lavindra, Hammada and Ahmed from whom I have received many support and help.

I couldn't thank less for Matthieu, Anthony and Jerome who work hard every day to keep the robots alive.

I want to give my special thanks to Joan who helped me on my "mind development", and also for his invaluable friendship and the alternative guidance on my work as on my sense of humor. Also I want to thank to Julie for her long distance help and support and for her car.

I couldn't stand all this time without some revolutionary ideas at coffee time from Juan, Ixbalank, David, Luiz, Gil, Andres, and the spiritual supervision of Ali. Also thanks for those with whom I shared beautiful moments as Karina, Eduardo (Lula), Farah, Kostas, Hammida, Toufik and Ricardo. Thank to you all of you guys for being there on the bad, the good and the ugly moments.

I couldn't get until the end if Assia wouldn't appeared in my life. She fulfilled the part of me that was as empty, pushing me to finish my work while feeling good with myself and

warm in my heart. Thank You Malanihaya.

The “gold” gratitude is for my parents (Mary & Pepe) who are my best career supporters and the persons that I love and admire the most. I want to add to this to my brothers Jorge and Memo for the endless talks and chats that help me to maintain my stamina and mental health. Love you guys.

Finally, I want to thank to all the people that contribute in some manner to my “séjour en France” with some short or long distance motivation, like my Grandma, my aunts Glenda and Rosy, my uncles Gaby and Martin, my dear friends Mario and Max, etc. etc. (i have a big family and the list could fulfill this thesis).

LF

A mis héroes: mis padres,
con todo mi corazón...

Contents

List of Figures

List of Tables

1 Introduction

A personal robot that brings something to drink when we arrive at home. A robot that can perform the domestic tasks while we are doing something else. A machine that always remembers to give you the medicine on time when you are sick. My personal dream and something that can help not only lazy people like me, but also people with motion disabilities.

Human tends to the automation of all its activities: cars, laundry machines, mixers, dishwashers, etc., all of them machines for simplifying our lives. Due to this fact, robots are becoming more and more popular within the years, as it is for people as for science. Is for this reason that research on the Human-Robot interaction (HRI) is growing exponentially.

The insertion of the robot on the human life sets new questions to answer to researchers. One of the challenges of human-robot interaction is the environment sharing and spatial placement between the robot and the human. Most of the works on motion planning tackle the problem of the movements that the robot has to perform in order to avoid collisions from one position to another, there are few related with the selection of the destination position for performing specific tasks. And less that consider to validate the whole task.

Added to all this, there is the problem of understanding human intentions to improve the robot's cognitive capabilities in order to result in a better interaction. For these purposes, robot has to adopt different human reasoning mechanisms, like "perspective taking" and "mental rotation".

The notion of *perspective taking* comes from psychological studies on human-human interactions. It refers essentially to the fact of reasoning from other persons point of view. It is also interpreted as taking its own perspective from a different point on the space by applying what is called an *egocentric perspective taking*, rotate the image in the "mind" to know how the environment is perceived from different places. These sets of actions are used by humans in their everyday lives, and are intended to ease communication between individuals and to help to have shorter and faster interactions.

Perspective taking can be used by the robot to generate configurations to approach human, or also to compute a geometric configuration to place itself where it can perceive an object and perform a task. Another problem that it is necessary to deal with, while interacting with humans, is not only to understand the human actions, but also that the robot actions have to be understood by the human. The actions that the robot performs must be comprehensible for the persons with it is interacting. For this purpose, the robot has to have geometric reasoning abilities in order to place itself, move and reason in a "human like" way.

One of the reasoning abilities is to understand at what the human makes reference. For this,

it is mandatory to effectively be able to interpret human attention, and to behave in order to share attention. Like this, the robot will perform actions for a bidirectional understanding, actions that will ease the interaction.

The decision and the evaluation of positions and actions for the human-robot interaction are part of the interest of this work, and we believe that the activities of this interaction must take inspiration from human-to-human interactions.

In the presented manuscript we describe different spatial reasoning tools where perspective taking is used to help our robots to perform different interaction tasks.

1.1 Contributions

In this work, i have proposed a set of algorithms that provides "social skills" in the geometrical reasoning of the robot. The major contribution on motion planning, is the adaptation of the social criteria as well as the human vision model to the search of the placement configuration. All this, fulfilling the objectives of the task, like perceiving the target and/or giving an object, while minimizing the displacement trajectory distance.

The presented approach proposes a framework that serves as:

- A link between a navigation planner and manipulation planner for the achievement of complete tasks.
- A link that interprets higher level instructions and transform them into specific goals to the motion planner.
- A way of evaluating task goal achievement.
- A scheme of reasoning about space through the "eyes of the human being" to help the robot to understand and to be understood by the human community.

This spatial reasoning mechanism employs psychological concepts of "perspective taking" and "mental rotation" to attack to general problems:

- *Motion planning for human-robot interaction*: where the robot uses ‚Äú√∫egocentric perspective taking,‚Äù to evaluate several configurations where the robot is able to perform different tasks of interaction.
- *A face-to-face human-robot interaction*: where the robot uses perspective taking of the human as a geometric tool to understand the human attention and intention in order to perform cooperative tasks.

1.2 Organization of the thesis

This document consist of the following chapters:

Chapter 2 presents an introduction to the concepts of perspective taking, mental rotation and joint attention from the point of view of different research areas, going from the description on the psychology, through some implicit adaptation in computer science, until its explicit integration on robotics architectures.

Chapter 3 shows the adaptation of psychological notions of perspective taking and mental rotation to help on the process of motion planning of the computation and the decision of the final configuration to perform different interactive tasks in home environments. This chapter also presents the results obtained in simulated scenarios.

Chapter 4 demonstrates the results of the perspective placement planner on a mobile robotic platform for different tasks and on different environments.

Chapter 5 describes how the same process of integration of the concepts can be used as a tool to understanding human actions as it is his attention. Here, the robot can place itself in human eyes in order to reason about its perception, and help to achieve joint actions. Furthermore, this chapter explains the implementation in another robot platform.

Finally, we will make some general conclusions that will show the general panorama of all the possible future applications.

1.3 Publications

The following publications are related with this work:

- **Towards Shared Attention through Geometric Reasoning for Human Robot Interaction** *Luis F. Marin-Urias, Emrah Akin Sisbot, Amit Kumar Pandey, Riichiro Tadakuma and Rachid Alami,* The 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids09), Paris, France. December 7-10, 2009.

- **Geometric tools for Perspective Taking for Human-Robot Interaction**, *Luis F. Marin-Urias, Emrah Akin Sisbot and Rachid Alami*, Proceedings of the 7th Mexican International Conference on Artificial Intelligence 2008, Mexico City, Mexico. – Best Poster Award –

- **Spatial Reasoning for Human Robot Interaction**, *Emrah Akin Sisbot, Luis Felipe Marin and Rachid Alami*, 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, USA.

- **A Human Aware Mobile Robot Motion Planner**, *Emrah Akin Sisbot, Luis F. Marin-Urias, Rachid Alami and Thierry Simeon*, IEEE Transactions on Robotics, Special Issue Human Robot Interaction,Vol. 23, No. 5, Oct. 2007.

- **Implementation of human perception algorithms on a mobile robot**, *Mathias Fontmarty, Thierry Germa, Brice Burger, Luis Felipe Marin, Steffen Knoop* Le 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV 2007), September 3-5 2007, Toulouse, France.

- **A mobile robot that performs human acceptable motion** , *E. Akin Sisbot, Luis F. Marin Urias, Rachid Alami and Thierry Siméon,* 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2006), Beijing, China.

- **Implementing a Human-Aware Robot System** , *E. Akin Sisbot, Aurelie Clodic, Luis F. Marin Urias, Mathias Fontmarty, Ludovic Brèthes and Rachid Alami*, IEEE International Symposium on Robot and Human Interactive Communication 2006 (RO-MAN 06), Hatfield, U.K.

2 State of the Art

2.1 Introduction

Even when the dream of some science-fiction writers and creators has been to haven a robot that understands our actions and helps on any domestic task, the human robot interaction is an area that has not been yet explored as wide as other areas of robotics. This is maybe, due to the complexity of reproducing "simple" social actions that we humans accomplish with no effort.

In order to interact with a human, the robot has not only to understand human actions, but also it has to be understood by the person whom it is interacting with. For achieving this, some authors implement basic "social skills" like eye-contact, voice recognition and generation, etc. on their systems in order to interact with people in their own "terms" of communication [Alami 06, Mitsunaga 06, Ido 06].

All these activities or skills help on having smoother social interactions, but they are not part of the robot autonomous reasoning. Psychological techniques have to be developed and implemented on robots, in order to obtain more "natural" and non-repetitive behaviors.

The concepts we are interested in are described as follows:

- *Mental Rotation:* The ability to rotate mental representations of two-dimensional and three-dimensional objects.
- *Perspective Taking:* The ability to perceive and understand what others say, do or see from their point of view. Here we focus on the "visual perspective taking"; in other words, understanding what others see from the place where they are situated.
- *Joint Attention:* The fact that two or more agents are paying attention at the same object or place for a common intention of communication.

This chapter presents an overview of the literature in different areas, about how psychological concepts of mental spatial transformation and/or spatial reasoning have, or may have, an influence on robot activities and actions to interact with a human.

Section 2.2 opens this chapter with an introduction to the concepts of *Perspective Taking* and *Mental Rotation* from a psychological point of view. Then we describe its explicit and implicit implementation in some areas of computer science and robotics. In Section 2.3, we deal with another important concept called "Joint Attention" which allows persons to create connections with objects for a common purpose or merely for competitive purposes.

Complementary to this, the robot needs to know its own limits. To this end, the automated acquisition and calculation of the robot capabilities of manipulation are necessary and are mentioned in Section 2.4.

Section 2.5, presents various human models that are found in literature and that are used in interaction experiments between humans and robots.

The final section describes a brief and general discussion of the presented works that serves as an introduction to the following chapters.

2.2 Perspective Taking and Mental Rotation

2.2.1 Psychology

The terms of *perspective taking* and *mental rotation* comes from psychological studies on human-to-human interaction. Inspired on the philosophical "Theory of mind"[Davidson 84], the human should be able to understand, imagine or "feel" that his interaction partner also has these abilities while they are achieving a task together. People without these skills are unable to interact with other persons and are considered as interaction impaired or "mind blinded" [Baron-Cohen 95b, Frith 01].

Most of the psychological studies about perspective taking are inspired on the work of Piaget on children [Piaget 52, Piaget 56]. Piaget has proposed an experiment to evaluate children's abilities to coordinate spatial perspectives. The task consists on showing participants (between 4-11 years old) a three mountain 3D model built on a table surrounded by four chairs. Then, the researchers place a doll on different chairs looking at the mountains, with different doll's points of view. Each child was asked to select the appropriate picture of the doll's perspective (i.e. "how is the doll watching right now?").

Piaget found that younger children (under 6 years old) selected the pictures depicting their own perspective.

These children were not able to distinguish between their own view and the one of the doll. He defined this as an egocentric point of view.

Flavell also conducted several studies on children. In [Flavell 92], he mentions the existence of *cognitive connections* between two persons through objects on the environment. One of these connections is seeing something and noticing that the other person is also capable of seeing the same thing.

The ability of changing perspectives with the other person is *perspective-taking*, and more precisely *visual perspective-taking*, when referring to the change of the point of view between persons.

Flavell categorizes perspective taking in two levels:

- *Level 1:* Being capable of noticing that the objects that are perceived from both observers may differ. In other words, objects perceived by one may or may not be perceived by the other.
- *Level 2:* When a person can infer the view point of the observer, and detect what is perceived from there (i.e. Same object can be perceived differently by the observers).

Both levels are illustrated on figures 2.1, on perspective taking level 1, the human has the ability of knowing that the monitor prevents the observer to see the object. On level 2, he knows how the observer perceives objects. e.g. the front of the screen.

(a) Human Perspective

(b) Level 1

(c) Level 2

Figure 2.1: Flavel's Perspective-Taking Levels. a) In this perspective the human is capable of knowing: b) which objects can be perceived by its partner (correct symbol) and which ones can't (cross symbol) (level 1) and c) how the objects can be perceived, face of the objects and their relative position (Level 2)

Inspired by these notions, Moll et al. [Moll 06] present studies to know at what age human develops visual perspective skills. They attest that this ability is present in 24 month-old children but not in 18 month-old children. Their experimental layout consists on face-to-face scenario with the experimenter and the child, where two objects placed between them and one of these objects is hidden from the experimenter's perspective.

Tversky et al. [Taylor 96, Tversky 99, Lee 01] show the effectiveness of changing perspectives between persons in face-to-face communication scenarios for spatial descriptions. They study the verbal method of the human for describing a scenario or an environment.

Taylor et al. [Taylor 96], have developed three key experiments to analyze descriptions from three different perspectives: route, survey and mixed. Route perspective refers to guidance descriptions that are referent to the listener as if they were walking on the scene (turn to your right). Survey perspective describes indications on global coordinates (north, south, east and west). Mixed perspective, as its name says, is the utilization of both types of perspectives.

They also mention that spatial descriptions have statements that need the localization of an object with respect to a "reference frame". This frame may be a coordinate system, a point of view, reference object, etc (the house at the left of the tree).

In this same work, a loose classification of reference frames from studies on spatial language

is suggested:

- *Deictic:* or Viewer-centered, when references are given on the observer coordinates.
- *Intrinsic:* or Object-centered, where references are with respect to an object.
- *Extrinsic:* or Environment-centered. Where references are given in global coordinates.

Taylor et al. also clarify that persons change perspectives if a doubt is present on the understanding of a description, in other words for helping its partner to understand. Another important element to notice in this work is the fact that, when making a spatial description in a "route perspective" people tend to imagine themselves as navigating on the environment.

The fact of imagining ourselves on different parts of the environment is considered by Zacks et Al. [Zacks 01] as a form of perspective changing called "Egocentric perspective transformation". Together with "mental rotation", these two notions conform two types of mental spatial transformations. The results of their experiments revealed that the two types of spatial transformations (rotating objects and changing perspectives) represent two different abilities.

Mental Rotation is a concept widely studied by Shepard and his co-workers [Shepard 71, Shepard 96]. It refers to the ability of rotating an object in the mind. Ackerman [Ackerman 96], investigates this phenomena and explains how a person can estimate his relative position in the world by 3D mental transformation of an object or of an entire place.

Lambrey et al. [Lambrey 08] directly study and evaluate which parts of the brain are involved in the "capacity of imagining the perspective of another observer" from the neuroscience point of view. In other words, when persons perform viewpoint recognition from another person.

The authors use virtual environments and an avatar to conduct their experiments with a computer and a person in front of the screen. They reproduce the images perceived by the avatar, show them together and then measure if the human can distinguish between the avatar's real perspective and the modified perspective.

Another important result that we can obtain from the studies of Lambrey et al. is the fact that, depending on the desired task, perspective taking may imply at least two different strategies or tasks:

- *object location memory:* Remember where a previously detected object was situated, and detect if this object has changed its position from a different point of view.
- *view point recognition:* Imagine the view point of the observer, and detect what is perceived from that point.

2.2.2 Computer Science and Human Computer Interaction (HCI)

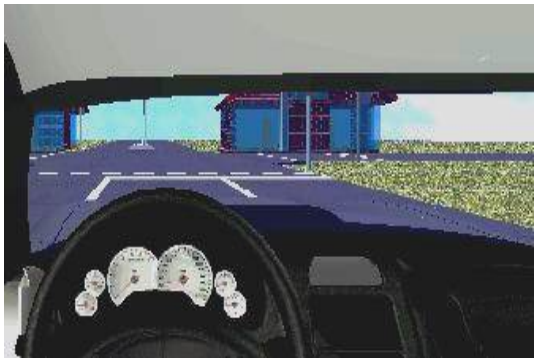
As mental rotation and perspective taking are important for a "social" interaction, they are often used in several areas involving human-machine interaction.

In computer graphics, mental rotation is used for simulating human-like view. For instance [Zhang 97] and [Chhugani 05], proposed efficient algorithms to compute which objects are visible and which ones are hidden, depending on the point of view.

These computer graphics algorithms are supposed to run in real time, and their main applications are video games, animation and interface design in order to perform a more intelligent and realistic human-computer interaction. Hsu et al. [Hsu 06] implement a motion planner into the avatar control to give a more realistic perspective while avoiding virtual obstacles.

For environment design in virtual reality, the conception of cars [de Sá 98, Rix 99], buildings [Mueller 06, Yin 09] or entire colonies [Ali 09] are done by taking into account the human perspective. This perspective is done by placing a virtual camera in different parts and/or positions and acquiring its perception from a simulated field of view. Perspective taking is important for establishing security and comfort parameters and also for aesthetical purposes, as shown on figure 2.2, where the different perspectives are measured for man and woman on a car design.

(a) Human FOV



(b) Male Perspective

(c) Female Perspective

Figure 2.2: Car virtual testing in a city environment [Rix 99]. The driver posture and field of view provides important information for car designers to perform modifications before a real model is obtained.

Human training [Menchaca-Brandan 07] is one of the important elements in missions, where the environment is not accessible or dangerous for humans (e.g. in space missions, hazardous

environments). Before the human goes to this type of missions, trainers use computer-aided tuition in simulation to measure human responses and actions in various situations. Figure 2.3 illustrates a human training on a system that simulates a mission on the space, where he has an assembly task.

Figure 2.3: Menchaca's training system for astronauts [Menchaca-Brandan 07], where the mental rotation and other abilities are evaluated and tested.

2.2.3 Robotics

Robots Simulators

"Mental rotation" has also found its place in robotics, and more precisely in mobile robot simulators [Faust 06], where a virtual environment simulates sensor data for the robot. The robot navigates and reacts on the virtual environment as it was on the real one. On these types of simulators, researchers can test different robot algorithms before testing them on real robots.

Object Modeling with Robots

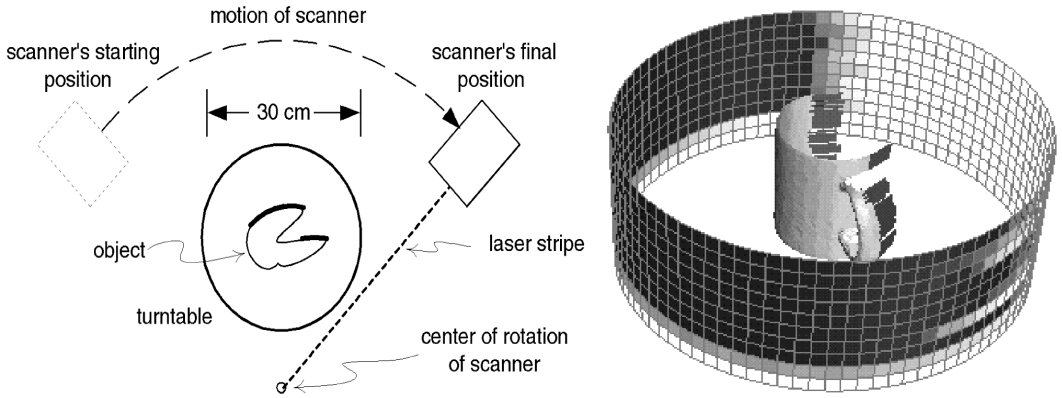
On the automated surface acquisition, there is the problem of determining which positions or orientations of a sensor give the most informative, and less repetitive, scanned surfaces from an unknown object or scene.

The approaches that attempt to resolve the ‚Äúnext best view problem‚Äù (NBV), as it is called, are also applications where a kind of mental rotation is applied. In these approaches, the sensor is the one that must rotate in the space to obtain different points of view.

To obtain a series of positions to construct 3D models of objects, the NBV approaches consist mainly on displacing on the contour of the objective in a circular way, where the sensor is oriented to the rotation axis [Pito 99] (shown in figure 2.4). It can also be extended to a spheric (or semi-spheric) form covering all the space around the object [Li 05] [Banta 00].

Other methods cover the surface of an object based on what is being incrementally perceived, like it is done on [Bottino 06] where the next positions are acquired from edges of silhouettes from 2D images.

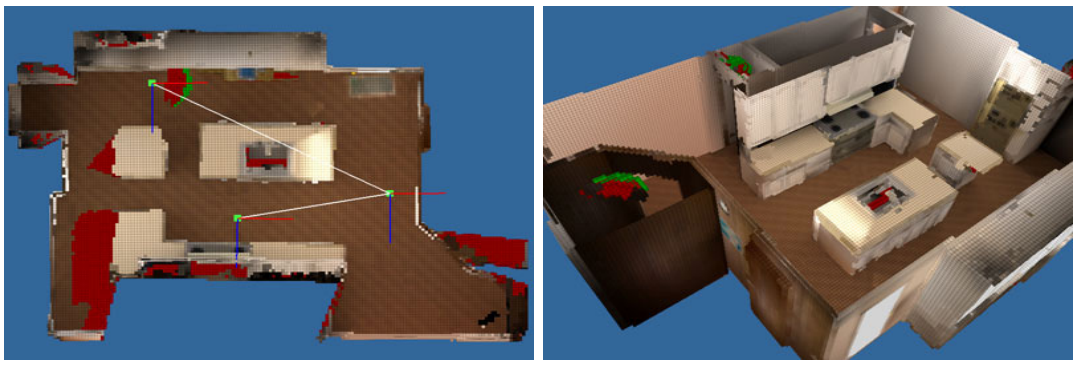
Not only single objects are the interest of NBV, but also the whole environments. [Null 06, Sanchiz 99, Wong 99] show occupancy voxel and ray tracing approaches to perform exploration



(a) (b)

Figure 2.4: Surrounding the objective to model it [Pito 99].

task in order to determine what is perceived, which parts are unknown, and which is the next place that will give more information [Low 06], as illustrated on figure 2.5.



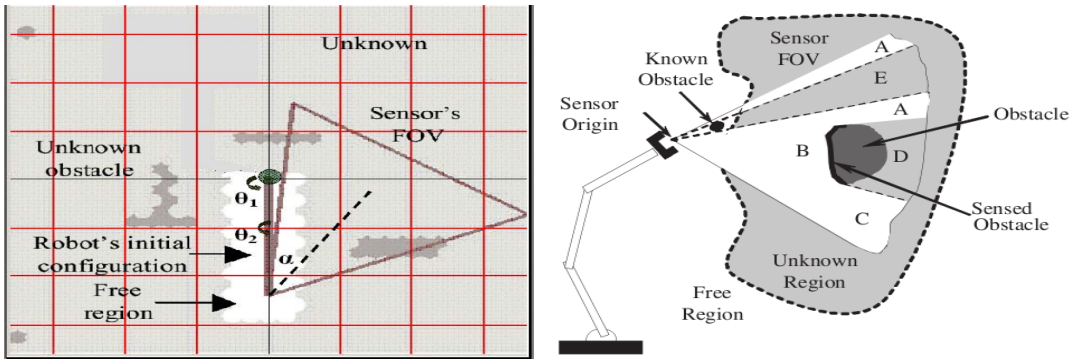
(a) Sensor Positions (b) Environment modelled on voxels

Figure 2.5: Calculating points on the environment where the sensor can acquire and model the more informative hidden regions on a room [Low 06].

The methods mentioned above attack essentially the optimal meshing of objects or environments problem. These methods consist on placing a "free-fly" point of view on the environment without taking into account the fact that the robot has to place the sensor attached to a body part.

Motions and positions are constrained by the morphology of the robot as also on the sensor's limited field of view. This problem is also called sensor-based motion planning, as tackle by Wang in [Wang 06] where he shows how to plan motions for an arm, with a camera on the extremity, in an unknown environment. Wang in his approach takes into account not only the possible motions but also the sensor perception. Here, the robot incrementally plans its movements in the free configuration space, and chooses the next arm configuration based on the zones perceived on its field of view (Figure 2.6-a), represented by a triangle in a 2D space as shown on Figure 2.6-b.

Since the robot has to place the sensor in a 3D world, 2D free space information is not enough to determine if the chosen position for a sensor is reachable and feasible by the robot.



(a) (b)

Figure 2.6: Wang's method [Wang 06] to move a robotic arm that incrementally acquire the free space depending on sensor data, in order to model the environment. The next configuration is chosen according to the information that the sensor has obtain at this position

Foissotte presents in [Foissotte 09], an approach that finds different collision free configurations for a humanoid robot, this in order to have several points of view for modeling an object, while satisfying stability constraint.

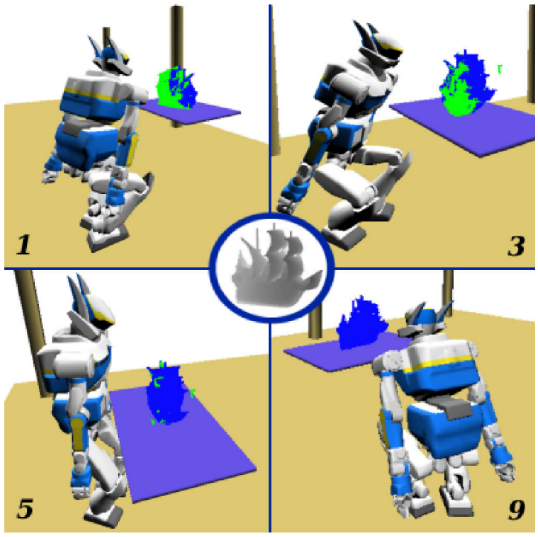


Figure 2.7: The whole structure must be taken into account when it is chosen a position of the sensor [Foissotte 09]. Many configurations are not possible due to the environment or the stability of the robot.

Here, the robot's next position is computed on-line. As the robot perceives from its cameras, it is predicting the next orientation and position of the humanoid's head.

These approaches ([Foissotte 09] and [Wang 06]), show the importance of the robot structure and its movement capacities for placing a sensor on an environment where not only different objects can prevent the robot to reach the desired position, but also where the field of view should be taken into account.

Nevertheless, some parameters are not taken into account, as the limited 3D field of view, collision on a 3D environment and displacement of the robot. This is due to the non-negligible increase of the computational time while determining all object-modeling positions and choosing the best one.

Furthermore, all the approaches mentioned above are for modeling an unknown object or an unknown environment. Moreover, the methods do not take into account a position where to place the robot in an acceptable posture where it can perceive a known object without visual occlusions.

Human Robot Interaction

Although human-robot interaction is a very active research field, there is no extensive amount of research on perspective taking applied to it.

Perspective taking has begun to appear in the HRI research field in the last years. Richarz et al.[Richarz 06] define a limited area in front of a human in order to obtain pointing places on the floor for a "come here task". The robot moves to the indicated point placing itself in the visible (for the human) and "pointable" zone of the person. In this work, the used area is defined by the intersection between what the human can point and what he can see.

Trafton et al. present in [Trafton 05a] a robot system that uses spatial reasoning with perspective taking to make decisions about human's point of view. They made studies on human-human interaction based on conversations of astronauts. Studies resulted on utterances that need perspective taking activities (listener to speaker and vice versa).

They proposed "Polyscheme", a cognitive and multi modular architecture that model human methods of representation, reasoning and problem solving. This architecture provides symbolic reasoning and planning through its modules called "specialists". One of these specialist is the "perspective specialist" that provides information of which objects are perceived by the human from his current position, and convert it to a symbolic level (helped with a vision system that recognizes determined colored objects).

The system proposed by Trafton and his co-workers aims at making decisions in ambiguous scenarios based on the task definition. For example, they analyze the sentence "give me the wrench" in a situation where two wrenches are present in the environment but where one of them is occluded to the human. The question that rises from this situation is: "Which wrench he wants?".

To answer this question the authors test their system that takes into account the perspective of the human partner on 4 different scenarios (with red cones instead of wrenches), obtained from the possible given situations on the astronauts conversations:

1. *One visible object:* No ambiguity is present.

2. *Two objects, One visible:* Medium ambiguity.

3. *One object hidden to the robot:* Medium ambiguity.

4. *Two visible objects:* High ambiguity.

The experiment results were encouraging on the application of perspective taking on the robot systems, where the robot could resolve the first 3 scenarios thanks to its perspective taking ability. They claim that the way of achieving a relevant human-robot interaction is through human-human interaction studies.

Nevertheless, further studies of Trafton et al. [Trafton 05b] reveal that humans have a slight preference on being asked what action to perform, rather than the robot taking the decision by itself. The preference was not much different from the robot taking the "good" decision (choosing the cone that was perceived by both, speaker and listener), but high enough to distinguish between the preferences of choosing the object out of speaker's sight. Asking is preferable for some people, but in a continuous interaction, like astronauts asking for a confirmation at every step of the communication results on a slow and non-fluent interaction. That is the main reason why the ability of taking the perspective of others is beneficial for robotic systems.

The works of Trafton et al. give us a good reason to use perspective taking on robots, but also they surface a question that is not clearly explained; "How the robot has to put itself on human's place and share his state of mind?", in other words, "How to take the perspective of the human?".

Breazeal et al. in [Breazeal 06] and [Berlin 06], partially answer this question. They mention the utilization of a cone for representing the perception of the human point of view, to mark the objects that are on "known hidden areas". The authors present a learning algorithm that takes into account the information on the teacher's visual perspective (with a predefined perspective information), to deal, as Trafton et al., with ambiguous demonstrations or references from a human teacher. In their experiments, they work on a scenario that consists on two visible buttons and one hidden button (from the human visual perspective). It can also be described as an example on how they deal with ambiguity in a case where the human teacher establishes the sentence "put all the buttons on". In this situation, the teacher only shows which should be the state of the two buttons that are perceived by him. The robot takes this fact into account and the non-perceived object is eliminated from the learning process.

In addition to this, the authors mention that they have implemented "social" and expressive skills as predefined behaviors (e.g. eye movements, hand gestures, etc.). These abilities are important to give a feedback to the tutor when an action is being developed, in order to give some "natural" communication between the robot and the teacher. One of these skills is the concept of joint attention that we are going to discuss on the next section.

2.3 Joint Attention

2.3.1 Psychology

The Joint attention is also known as Shared attention, but the concept has no standard definition. Some authors define it as the fact that two (or more) persons are looking at the same object[Butterworth 95]. Nevertheless, most of the authors agree with Tomasello [Tomasello 95] that simultaneous looking is only a part of the whole mechanism. There must be a "correspondance" between the persons involved in the interaction through a communication channel (the object) as the Flavell's "cognitive connections" [Flavell 92] mentioned on the previous section of perspective taking. Tomasello also adds that there should be a "mutual knowledge", an idea that concords with Baron-Cohen's description of the shared attention mechanism [Baron-Cohen 95a], and also mentioned and studied by Warreyn in [Warreyn 05].

Tomasello [Tomasello 99] divides the shared attention in three types:

- *Check Attention:* Paying attention to the partner or the object that he/she explicitly shows (Fig. 2.8-a).
- *Follow Attention:* Follow the gaze or the pointing direction of the partner(Fig. 2.8-b).

- *Direct Attention:* Influencing on the partner attention, with voice, eye gaze or pointing gestures(Fig. 2.8-c).

(a) Checking the attention

(b) Following the attention

(c) Father directing the attention of the child

Figure 2.8: Tomasello's Joint attention types. a) Check Attention of the partner b) Follow the gaze of the partner c) the father is influencing the attention of Unai (the child).

Joint attention is proved to be a basic ability for survival purposes in humans and it is present even on primates like chimpanzees [Tomasello 08]. Either for cooperation or competition, Joint/Shared attention together with Perspective Taking and Mental Rotation are psychological abilities in human to human interaction that should be used in human-machine interaction. Many authors have the same perception, as we can see on the following section.

2.3.2 HCI and Human Robot Interaction (HRI)

Different approaches mention the importance of including joint attention feature while interacting with machines. The method that most of these studies attack the problem is by following the gaze of the human partner and detecting the salient object in this direction.

Peters et al. in [Peters 08] show some preliminary work about measuring engagement on the interaction between the human user and a virtual agent through virtual objects on the screen. The measurement is done through a gaze following system, for detecting the object of interest of the human, which is displayed on the system screen. An object becomes "of interest" if the person is concentrating his gaze for a determined period of time, and if the agent makes reference to it.

In [Huang 08], the authors show an algorithm for tracking human faces and optical flow to obtain gaze direction with pre-established aims around the face. The joint attention is obtained by searching the object of human attention following the line from the face on the gaze direction.

Similar work is performed by Nagai on human-robot interaction [Nagai 03]. They present a learning algorithm to acquire the ability of human gaze following until detecting a salient object. Pan and tilt of the robot's cameras are introduced as parameters for the learning process.

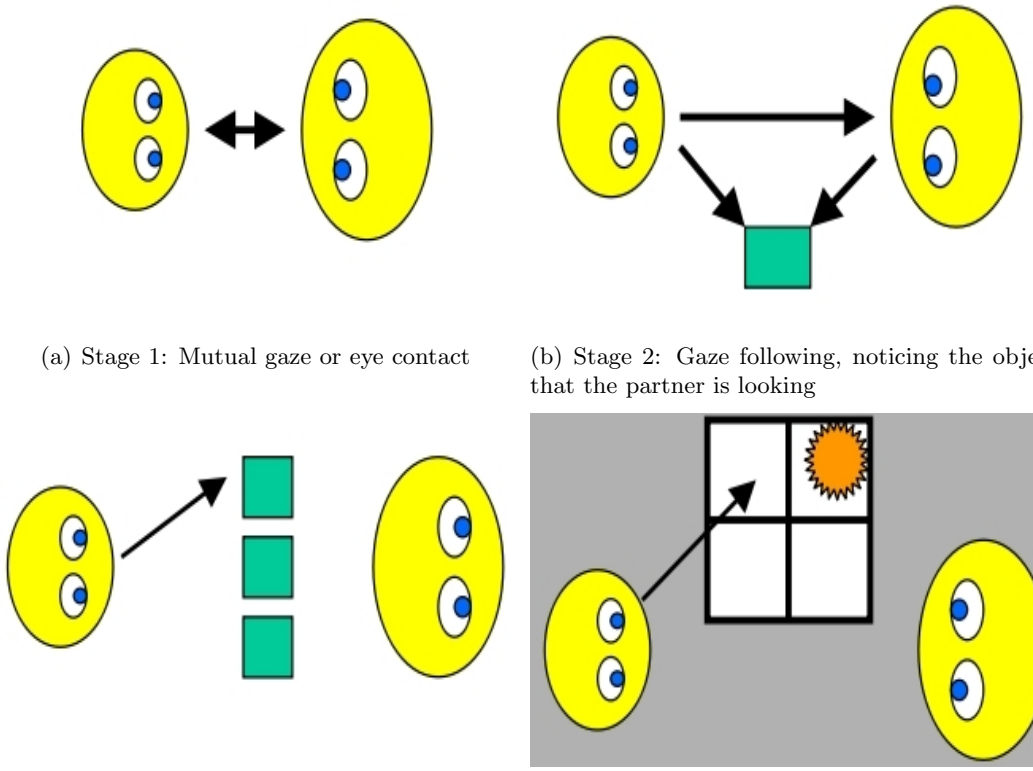
Inspired on this work, Sumioka et al. [Sumioka 07] attempt to find the causality between the perception and the following action by "transfer entropy". In other words they propose that the robot can autonomously select a pair of variables (perception, action) that forms a causal structure. For example, if the "caregiver" (as it is called the human in front of the robot) looks at an object, the robot can either move the arm or follow the gaze of the caregiver. All the transfer entropy quantification is based on the probabilities given on the caregiver's model. For their experiments, they use a simulated environment and their results tend to obtain a simultaneous looking, or what they call joint attention.

Looking at one object at the same time by gaze following (or other gesture) is only one of the aspects to be considered to perform joint attention. Kaplan and Hafner [Kaplan 06] take Tomasello's definition and remark that this activity is only one of the skills to be implemented to obtain joint attention. They describe the concept "joint attention" as a bilateral process between at least two agents (human, robot, etc.), both aware about the intentions of the other. They also mention that this process has to fulfill a least four prerequisites:

- Attention detection. To follow other agent's attentional behavior (i.e. Gaze following).
- Attention manipulation. To influence on the attentional behavior of the other agents.
- Social coordination. To achieve joint coordinated actions (turn-taking, role-switching, etc.)
- Intentional understanding. To notice if they share the same intention to achieve same goal.

Based on these notions, on robots equipped with arms, activities like pointing gesture generation are also implemented [Ido 06, Berlin 06] to recognize where the objects are located. Scassellati describes that the imitation of some other "human social cues" has to be taken into account (added to tracking gaze system) in the robot abilities [Scassellati 99]. He mention that the recognition or execution of a gesture that can manipulate the attention of a partner (i.e. declarative and imperative pointing, "eye contact", etc.), helps to the development of better social behaviors. At that time, he had implemented what he called "mutual gaze" and the conception of the "pointing gesture" based on the kinematics of the robot.

Scassellati has also shown a task-based skill decomposition to achieve Joint attention, and that is closer to the psychological definition. The first step is the recognition and maintenance of eye contact. He mentions that this ability is present in many animals but that only in human and great apes has a social meaning. The second step is to follow the gaze of the partner. The third stage is to point for an object out of the reach. The final skill evolution step is to point a distal object, to influence partner,Äôs attention. The joint attention decomposition on its different steps is represented in Figure 2.9.



(a) Stage 1: Mutual gaze or eye contact

(b) Stage 2: Gaze following, noticing the object that the partner is looking

(c) Stage 3: Imperative pointing of an object out of reach ("want that")

(d) Stage 4: Declarative pointing of a distal object ("look at that")

Figure 2.9: Scassellati's [Scassellati 99] task-based joint attention decomposition. Arrows indicate the attention direction.

Imai et al. presented a robot platform [Imai 03] that performs vocal utterances added to "eye contact" with predefined pointing gestures which carries the attention of the human to an referenced object (performing joint attention). A basic geometric reasoning is employed to infer the position of the pointed object. Following the work of Imai, Kanda et al. [Kanda 07] find rules for selecting "communicative units" through experiments with a WOZ[1] method. They also demonstrate that a robot that behaves like a human listener and that develops a cooperative behavior is more socially accepted. Imitation of human actions is a very common procedure in HRI. In [Ogata 09], a robot learns how to interact with an object by observing the object states changed by the human. Similarly, in [Brooks 04], Brooks et al. presented their robot Leonardo showing also some characteristics of shared attention with its gestures to communicate and learn with and from human, while playing social games.

The authors believe that embodied socially aware characters can have the ability to improve the experience for the human player. The robot performs head motions to look at the same button, that the human in front of it is making reference (included reasoning on human's pointing gestures as shown on Figure 2.10). An imitation process of the teacher,Äôs gaze orientation obtains the head turning motion. The robot also "tries" to infer the human state through the person's facial

[1]Wizard of Oz: Experiment method where a teleoperated robot is placed with users that naively believe on its autonomy.

expressions.

(a) Recognizing pointing gesture (b) Confirming Information (c) Expresing a doubt, asking for confirmation

Figure 2.10: Leonardo [Brooks 04], performing a gaze and pointing while learning games from a human teacher.

Johnson et al. [Johnson 05] propose internal inverse models, that are representations of motor activities added to the simulation of the perspective of another agent. This models intend to predict the external agent intention through the inner model of the robot's capabilities. The possible actions are chosen in terms of eligibility and applicability of the action.

Okamoto [Okamoto 05] presents a theoretical approach for a robot listener, defining what he calls an "empathy channel". He adopts the concept of empathy as the fact of taking the others point of view (what here we call perspective taking). The empathy activity is performed through a channel, once they are in a common shared space. This space is the result of combining both partners' view areas with a common objective.

As Okamoto et al., Green [Green 07b, Green 07a], Sidner et al. [Sidner 04, Sidner 08] and Staude and Crocker [Staudte 09] show that information obtained from the other actions added to the visual attention can help not only to recognize the object of attention, but also to know the intention of the human partner. Staude achieves in her experiments, not to follow the gaze of the human, but to conduct the human towards the robot's gaze. In this way, the human's attention is drawn to the object, accomplishing the Kaplan's second prerequisite of joint attention.

2.4 Robot Affordance

If the goal of the interaction is that, one or both actors reach the interaction object (as the work of Berlin, Breazeal or Brooks and others), then affordance tests must be performed by the robot, either for itself or for its partner. Tomasello [Tomasello 08] mentions that chimpanzees can measure not only their own affordance but also the affordance of their partner (for competitive reasons). Hoffman et al. [Hoffman 04] treat partially this problem at symbolic level, by classifying the objects as "achievable", "impossible" or "irrelevant" (for the robot). Here, the authors present a goal-driven hierarchical task representation, and a resulting collaborative turn-taking system.

However, the affordance or the reachability in terms of the robot capabilities has been widely studied. Most of the work on this area is often presented as manipulation or motion planning problems. Zacharias et al. [Zacharias 07] present a method to generate a capability map of a two-arm robot. This capability map is the workspace of the manipulator capabilities to take an

object from different positions and different orientations. Fedrizzi et al. [Fedrizzi 09] by their part, construct what they call an ARPlace that is a zone where the robot has better probability to accomplish multiple tasks. This ARPlace is build by the intersection of previously learned zone obtained from successful task accomplishments.

2.5 Human Models on HRI

When the intention of a mobile robot is to interact with a human, it must have a representation of this human to know where he is located, where he is looking at or how close he is. Often, the method to treat the human is in the same way as an obstacle or in forms that the human is the one that has to adapt to the robot's actions. Human model is important for the robot system design to adopt human friendly behaviors that promote the acceptability and the interaction with robots on human environments.

2.5.1 Position and orientation

The simplest method for representing the human position and orientation used by many authors [Feil-Seifer 05, Nakauchi 02, Takemura 07, Yoshimi 06], basically consists of taking a 2D position of the person's center of mass projected on the floor (supposing that human can not be suspended in the air all the time), and of assigning an orientation (x, y, θ). On this work we will reference to this method as the "flat method".

For systems working on mobile robots, the detection of the head orientation is often difficult, without mentioning the detection of the eye orientation. Depending on the sensors, some authors opt for taking the chest orientation [Yoda 97], but if the robot is equipped only with a laser sensor then the position is given by leg detection and if human is moving then the orientation is on the direction of the walking direction[Baba 06, Hoeller 07]. Despite being a simplistic approach, studies on human walking proved that the human naturally looks at the same direction of their motions [Hicheur 05].

2.5.2 Human Field of View (FOV)

As the orientation, human field-of-view is often represented in the flat method. Depending on the interest of the approaches, it is represented as a semi-circular area in front of the human gaze orientation around $180°$ as defined by Schmalstieg [Schmalstieg 97] or Costella [Costella 95].

This value is wide enough and events that occur on this area can stimulate the attention of the human, but the human cannot focus on every single object in this area. Also the reaction time for detecting events varies incrementally from the center of the FOV to its side limits [Mizuhara 99]. Furthermore, from $154°$ from the gaze direction, there is a slope increase of the reaction time. The area where the human is concentrating his attention is called "the visual attention area". Inside this area, we can determine the objects that are the focus of a person's attention [Müller 05, Nakayasu 07].

The FOV and its visible region can also be modeled in a three dimension space through a cone, as represented by 3D graphics researchers (see Section 2.2.2).

2.5.3 Personal Spaces and Pose Estimation

An important aspect to take into account in a social mobile robot is how and where to approach the person to interact with. Satake et al [Satake 09] estimate the human motion to move the robot to a future rendezvous point. The placement of this point of interaction is delimited by human preferences on interaction zones, often defined by distance ranges between the robot and the human [Hall 66, Lee 02, Pacchierotti 05].

Moreover, the social robot has to decide where and how to place itself inside these zones. This depends mainly on the interaction itself as Huettenrauch [Huettenrauch 06] showed on WOZ experiments which measure the preferences on the Kendon's F-formation arrangements [Kendon 90].

On the inverse direction, Svenstrup et al. [Svenstrup 09] model the interaction zones with a potential function, while the human is the one that approaches the robot to interact. The authors classify the zones of the persons that either are not interested for interaction, or considered for interaction or a person interested on interaction.

Yamaoka et al. [Yamaoka 08] use zones (called O-space) to find a position in a triangular formation between the human, the object to show and the robot presenter, as shown on the figure 2.11. The presented approach consists on mixing the zones of the human and the object, and then on the closest vertex of the intersection, the position is defined. The robot orientation is determined by the human orientation and the object position. Both, human model and its FOV are represented in flat method. Yamaoka in this work and in [Yamaoka 09], presents joint attention tasks, where the robot has not only to arrive to an interaction point but also consider human capabilities for interaction.

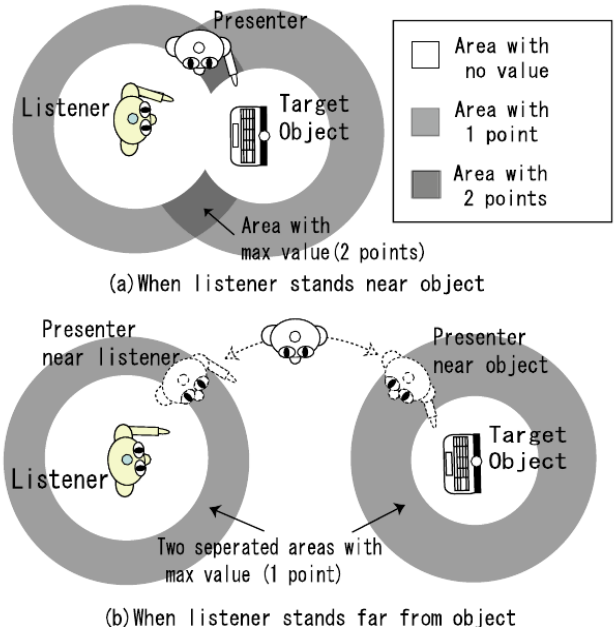


Figure 2.11: Yamaoka's presenter robot [Yamaoka 08], The robot calculates where it has to place itself to present an object. The circles are the O-spaces or interaction regions of the robot.

2.6 Discussion

This chapter has introduced three important psychological concepts of human to human interaction that are the basis of the inspiration for the development of this work and also for many others. The studies on perspective-taking, mental rotation and joint-attention are quite extensive and mature on the psychology research, while in the human-robot interaction context are relatively new and not yet studied in depth.

On the attempt to achieve a good interaction between the robot and the human, authors must implement more and more human reasoning techniques in order to obtain the engagement of both actors on the interaction. Most of the presented authors in this chapter use these techniques either in an explicit or in an implicit way, obtaining smoother interactions or more "natural" behaviors.

Nevertheless, the majority of the presented methods covers one of the three concepts or takes them as separated and non-related techniques. Few approaches use these notions together place the robot in trivial situations, where there are no obstacles, almost no occlusions (or already known) or where the robot does not have to place itself, or if it does, it is in a very simplistic and flat representation where the notion of perspective looses much information.

On motion planning we can find a wide variety of works on computing the path between two points. But there are very few works in literature that attack the problem of choosing a destination position for calculating this path, and even less that covers the concepts mentioned above. Added to this, in Human-Robot interaction there is a need of two things: Real Time on the Real World.

For all these requirements, we need a powerful reasoning system that takes into account its capabilities and limits and that gives a rapid response in a natural and/or social way, achieving a good interaction with the human.

3

Perspective Taking Applied to Motion Planning

3.1 Introduction

Robots sharing the human environment, yet many questions need to be answered on Human-Robot interaction. Motions, actions and tasks have to be adapted to the presence of humans to answer the questions of "what and how" the robot should do in order to interact with humans. In robotics, there are two main areas to answer these questions: "task planning" that tries to determine "what" produce a set of actions to accomplish a task; and "motion planning and control" that answers the question of "how" to perform these actions.

Still, there is another key question to answer, "where an action should take place?". The answer of this question depends mostly on the goal of the task. Where the robot has to place itself in order to accomplish an action, or where it has to place its sensors to help to perform a task, these are examples of problems that are important for any robot but crucial for a robot that interacts with humans.

In motion planning, there is a wide research for finding robot paths from an initial to a final configuration [Latombe 91, Laumond 97, Lavalle 06], but not much that looks for the final configuration, and less that search for a final sensor-based configuration. But this is still not enough, the final configuration has to be chosen in order to perform a task, and finally the task and the configuration must be adapted to the human presence.

As part of the introduction of the human awareness in the selection of the final configuration, the robot has to adopt some "social" abilities. In other words, the robot has to place itself in an "acceptable" manner for the human. One of these social abilities is *perspective taking*: placing itself at the place of another agent (in this case, the human), and more precisely visual perspective-taking. This means not only to model its owns sensors to find the configuration but also to reason about what the other sees.

This chapter will present how the psychological concepts "perspective taking" and "mental rotation" can also be adapted into the behavior of a mobile robot. This adaptation will help to a motion planner to compute different positions in order to move for closely interacting with persons or simply approaching different places.

In the first section, we will present a model of the human that takes into account several properties of interaction as well as the adaptation of the mentioned psychological concepts. In Section 3.3 we discuss a human aware motion planner approach that computes navigation and manipulation paths taking explicitly into account the presence of humans.

Section 3.4 will present the perspective placement planner that introduces social abilities to the robot actions for the purpose of finding sensor based and task based final configurations. Section 3.5 will illustrate simulation results of this planner.

Finally, a discussion concludes this chapter and presents a number of additional features that may be added to this work.

3.2 A Human Model for Interaction

The "social robot" that shares the same space with the human must have a model of the environment, and a model of the human. The environment model should be specific to human environments where in one hand there are obstacles that create frontiers like walls and doors, and on the other hand there are smaller obstacles which are not part of the building, like tables, baskets, shelfs, etc.

For the human model, the robot should take into account not only (and imperatively) the collision avoidance with the person that ensures the human's security but also his comfort and preferences.

Furthermore, the robot has to place itself in a position where it can achieve its task. Object manipulation or object handing are examples of tasks that the robot can perform at a close position to the human/object.

The presented approach model for interaction is based on occupancy grids and grid cost representations of the human and his environment. It is an enhanced model of the originally presented by Sisbot in [Sisbot 08].

3.2.1 Representation of Human's Environment

The environment representation is important for the robot, especially when the robot is supposed to move inside of it. This representation essentially consists of a description of the elements (i.e. furnitures, objects, humans and the robot itself).

Knowing the structure of the elements and their positions in space is essential for motion planning. This information allows computing paths and trajectories that will allow the displacement from one robot configuration to another while ensuring the collision avoidance based on the geometry of the robot and on the requirements of the task.

When the robot has to accomplish a task with a determined object or around a specified place, it is often useless to perform the configuration search in the whole environment, for example in the phrase "put the cup on the table" the robot has to look for a position around the table. Defining the limits of this search is important for the performance and the time response of the algorithms.

Environment's Free Space

The free space definition of the environment depends on the geometrical structure of the robot, the obstacles and on the robot's accessibility. Here, the space representation method starts on dividing the free space in two parts: the 2D and the 3D representation of the environment, for navigation and manipulation task respectively[1].

The 2D model also represents the two-dimensional free configuration space where the robot can pass or place itself in order to accomplish a task.

[1] 2D for placement reasoning and 3D for "posture" reasoning

2D Model: The Navigation Free Space

The 2D navigational model, is constructed by an extended flat method. This means, the 2D projection on the floor of the obstacles and the robot of the environment, with the difference that it is represented in a grid containing different values and not only a representation of the occupancy. This environment grid will be defined as:

$$G = (M_{n,p}, E_1 \dots E_n, f) \tag{3.1}$$

where $M_{n,p}$ is a matrix containing $n*p$ cells represented by $a_{i,j}$, the cost of the coordinate (i, j) in the grid, $E_1 \dots E_n$ is the list of elements (humans and objects) in the environment. The function f calculates the value of each cell according to its coordinate by taking into account a human or an object. This representation of the environment allows the robot to solve the navigation task of navigation, while respecting the constraints of collision-freeness with the obstacles.

The representation of all the obstacles in the environment consists of an occupancy grid defined as:

$$G_{obstacles} = (M_{n,p}, R, f_{obstacle}, Sp, \varepsilon) \tag{3.2}$$

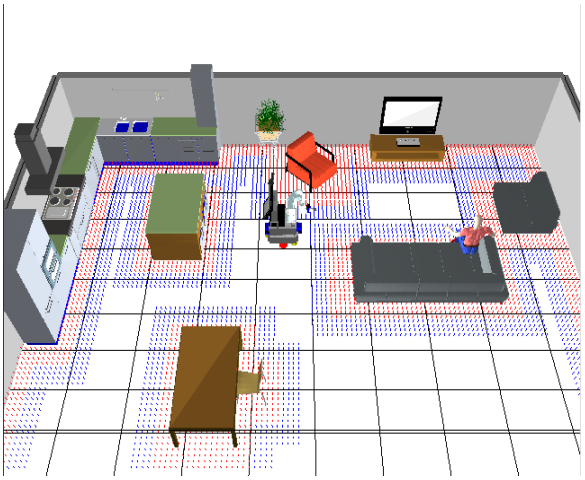
where R is the robot, ε the environment, *Sp* is the *sampling rate* and $f_{obstacle}$ is the cost function denoting the existence of an obstacle for a point in the "Obstacle grid"

The function $f_{obstacle}$ gives three different values depending on the 2D projection of the obstacles on the floor. If the 2D collision test between the robot and an obstacle projection is sure then the function returns an "in collision" value; if the collision occurs only in some robot orientations then the returned value is "in possible collision,Äô,Äô; otherwise the function returns that is "collision free". Expressed in a different form, the cost function of the Obstacle Grid is defined as:

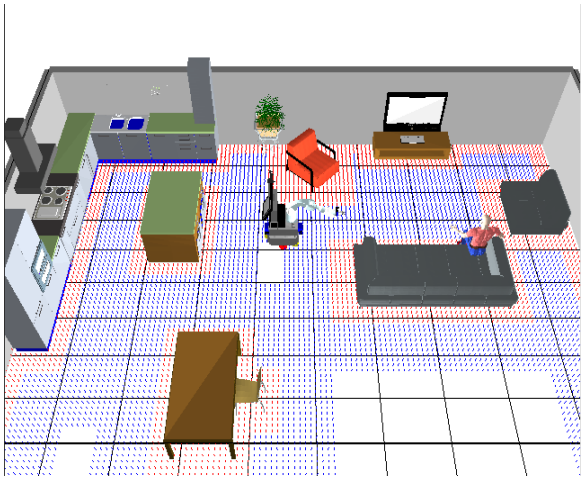
$$f_{obstacle}(R, \varepsilon, i, j) = \begin{cases} -2 & \text{if } \varepsilon_{2D} \bigcap BBR_{min2D} \\ -1 & \text{if } \neg(\varepsilon_{2D} \bigcap BBR_{min2D}) \wedge (\varepsilon_{2D} \bigcap BBR_{2D}) \\ 0 & \text{if } \neg(\varepsilon_{2D} \bigcap BBR_{min2D}) \end{cases} \tag{3.3}$$

where $\bigcap$ is a non-empty intersection, ε_{2D} is the two-dimensional projection of the environment, BBR projection of the bounding box that covers whole current robot configuration, and BBR_{min} is the representation of the base of the robot or the "minimal bounding box". The difference between these two BBR's is conceived in order to separate the robot base structure from the rest of its configuration.

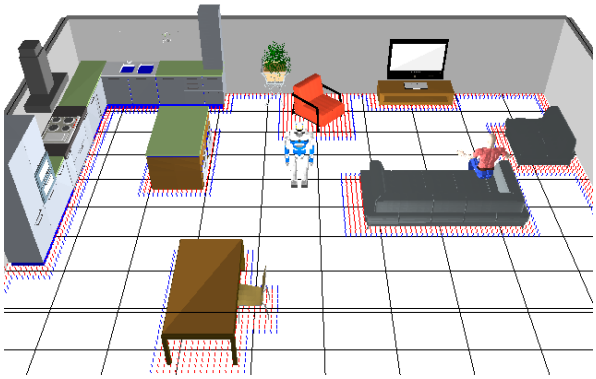
The figure 3.1 illustrates different obstacle grid values depending on the configuration of the extremities of the robot or on its structure.



(a) Robot 1 with its arm retracted



(b) Robot 1 with its arm extended



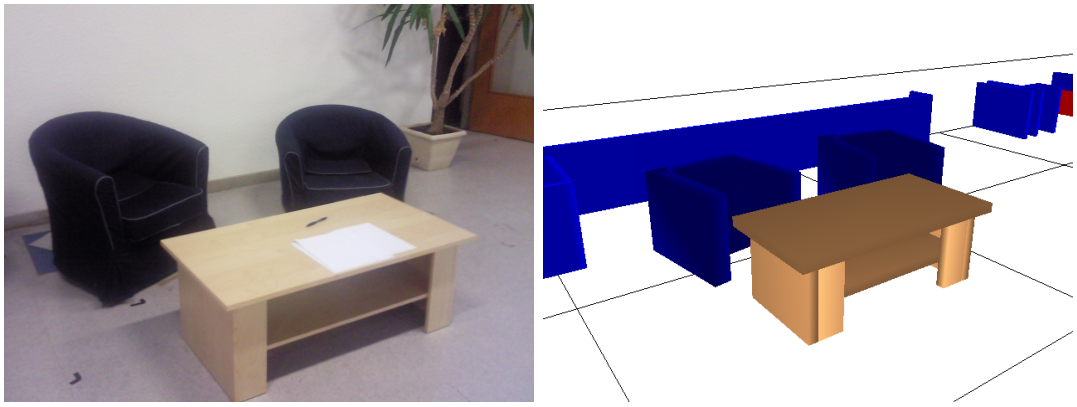
(c) Robot 2 different obstacle grid

Figure 3.1: Obstacle grids return different values depending on the structure of the robot or on the configuration of its extremities. The red zones are the cells "in collision"; the blue area returns "in possible collision"; the clear zones are cells in a collision-free position.

3D Model: Objects 3D structure

The furniture or objects to carry (i.e. bottles, plates), that is common to find on human environments, can have its represented model on 3D polygons. This model will correspond to its real form for different purposes of this work. For example: for manipulation, 3D collision avoidance or even for its perception representation. Figure 3.2 shows the representation of the objects found on a real human environment and the model the robot's representation of this environment.

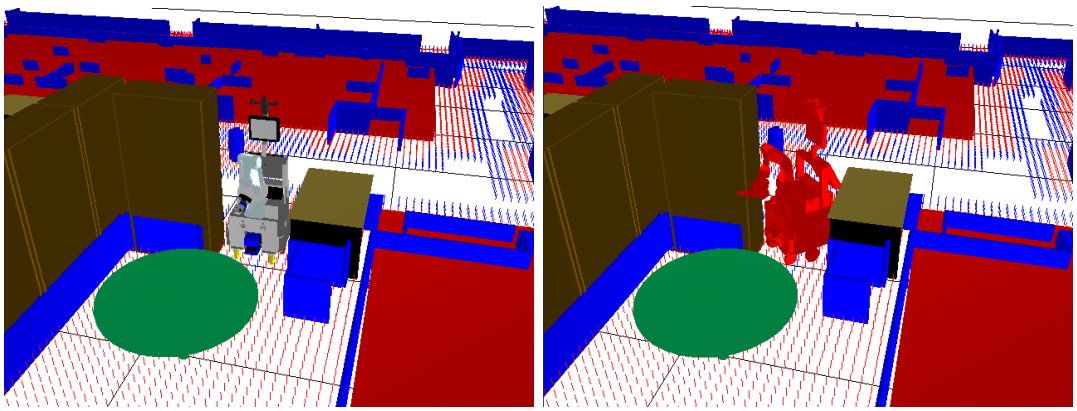
When the status of a cell in the Obstacle Grid is "in possible collision", further collision test are needed to be performed in order to have the certainty that the robot is not in collision. To know this, the robot needs to have the objects' three-dimensional structure representations and test every polygon intersection with the robot structure at the current configuration.



(a) Furniture real model on the environment (b) Representation of the furniture

Figure 3.2: The human environment and its representation on the robot. With this model the robot is able to perform precise motions without collisions.

As we can see on figure 3.3 a robot in a blue cell on the 2D grid (as shown on the figure 3.1), can be in collision with the obstacles on the environment only on some configurations. The robot has to test if a desired configuration in a specific place is collision-free or not.



(a) The robot without collision (b) The robot is in collision on the same position

Figure 3.3: If a robot is in a 2D cell marked as "in possible collision" then it has to perform further collision tests. The figures represent a robot on the same position with different orientations; a) the robot is not in collision and in figure b) is in collision because the arm is in contact with the 3D model of the shelf.

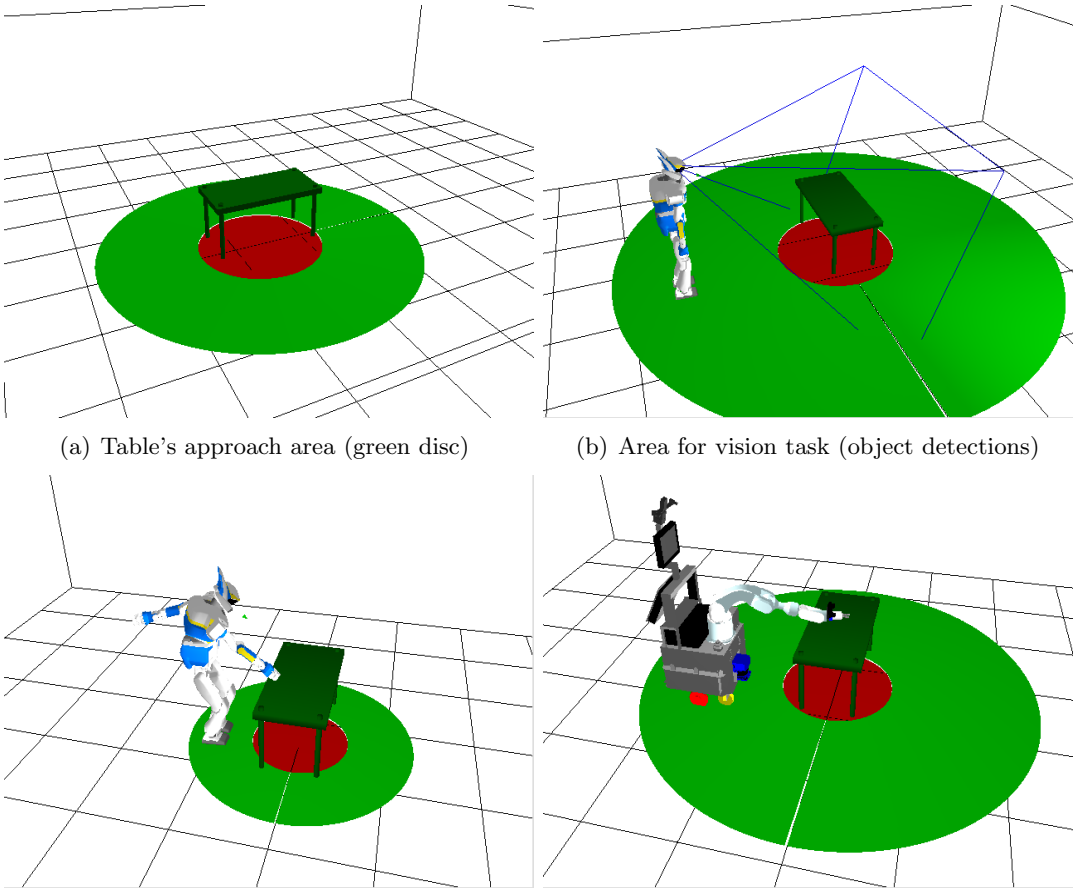
Object's Approach Area

The "Known Objects", objects in the environment like tables, desks, sofas, chairs, etc., considered as fixed obstacles, are also containers of other objects, surfaces where the robot has to perform some task, for example to search for an object or to place/pick an object.

In order to get close to these objects, I have delimited a zone around them where the robot can search its position in order to perform its desired task. The delimitation of the distances

of this area must be conducted following the constraints of the task as well as considering the robot capabilities (i.e. extremities length). The "approach area", as it is called this zone[2], is a proximity distance model of an object for a specific robot. These distances are based on the robot manipulation capabilities of this object and also on its perception limits, depending on the task.

Figures 3.4-b to 3.4-d show approach areas for different tasks and for different robot capabilities. The sizes can vary in the same task type, for example, in the vision task illustrated in figure 3.4-b, the goal could be to look for little objects on the table surface. In this case the area size, even when the limits of the camera are wide, could change to a more reduced area around the table.



(a) Table's approach area (green disc)

(b) Area for vision task (object detections)

(c) Manipulation limits of a humanoid robot

(d) Manipulation limits of a robot with a longer arm

Figure 3.4: Approach area is represented with a green disc. The size of this disc depends on the limits of the robot structure and on the task

[2]In this work we will use the terms "zone" and "area" indistinctly for making reference to delimited regions

3.2.2 The Human Costs Grid Model

Human-human interactions follow implicit rules or protocols that have been found from several studies [Hall 66, Yoda 97, Low 03, Huettenrauch 06]. These implicit protocols can be applied to human-robot interaction, but considering that the second actor of this interaction is not a human. For achieving this type of interaction, it is necessary to gather additional information about the properties of the human-robot interaction in order to acquire more acceptance from the human [Walters 05, Dautenhahn 06].

However, only a limited number of work considers such properties of interaction and often in an ad hoc manner. To integrate these properties in a generic way, it is possible to integrate them on the model of the human as criteria that define the way of interaction.

The method explained here, consists of introducing two additional criteria to the motion planning stage in order to ensure human safety and comfort. These criteria, namely "safety criterion" and "visibility criterion", present two important aspects of robot navigation in a human-robot interaction scenario.

Each criterion is represented by a set of numerical values stored in two kinds of grids, the 2D and the 3D grids. This criterion grids contain a set of cells with various costs derived from the relative positions of humans in the environment, humans' states, their capabilities, and preferences. These costs affect only correspondent cells from the environment's free space explained in Section 3.2.1.

A 2D criterion grid G is defined as:

$$G = (M_{n,p}, H_1 \dots H_n, f) \tag{3.4}$$

where $M_{n,p}$ is a matrix containing $n * p$ cells represented by $a_{i,j}$, the cost of the coordinate (i, j) in the grid, $H_1 \dots H_n$ is the list of humans in the environment. The function f calculates the value of each cell according to its coordinate by taking into account only one human. The matrix M is constructed by the equation:

$$a_{i,j} = max_k(f(H_k, i, j)) \tag{3.5}$$

A human H_i is modelled by $H_i = (St, State_1 \dots State_n)$ where St is the structure and kinematics of the human and $State_i$ is a human state defined by a number of cost parameters. A state is defined by:

$$State_i = (Name, Conf, Param) \tag{3.6}$$

where $Name$ defines the posture state (e.g. $Name = SITTING\ |STANDING$), $Conf$ is the human's configuration in that state (if applicable) and $Param$ represents the data needed to compute costs according to that state.

A 3D grid is an extension of the 2D grid for a three dimension space. The constitution of each type of grid is formed by squared cells of 0.2m for 2D grids, and by cubic cells with 0.05m by side for the 3D grids[3]. We will further explain below the structure of the "safety" and of the "visibility" criteria and their underlying properties.

[3]Further information about the grid model can be found in [Sisbot 08]

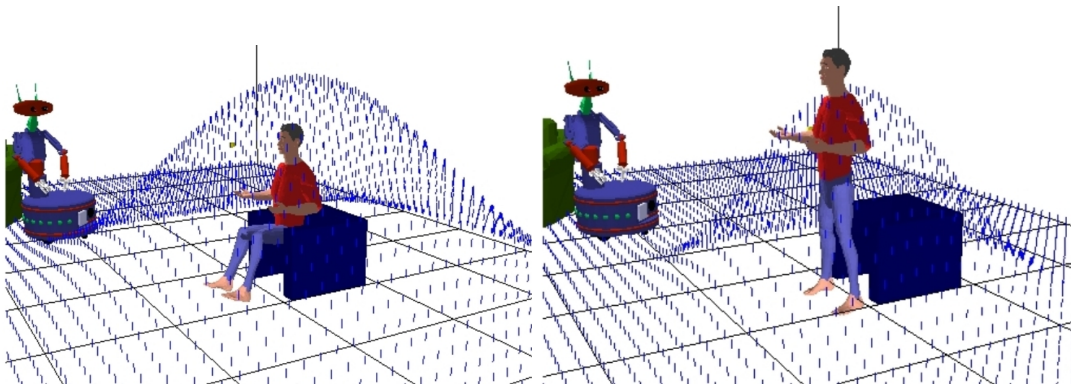
Safety

Safety on the 2D Grid

The first criterion, called "safety criterion", mainly focuses on ensuring the safety of the human by controlling the approximation distance of the robot (the farther the better). However in some cases, as in a close interaction (e.g. handling an object), the robot has to approach the person with whom it wants to interact. Therefore, the distance between the robot and the human is neither uniform nor fixed and depends on the interaction task. The feeling of safety is highly dependent on the human's personality, his physical capabilities and his actual situation; for example, safety differs highly in a sitting position compared to standing. When the human is sitting, his mobility is reduced and he tends to have a low tolerance to the robot getting close. On the contrary, when standing up he has a higher mobility, thus allowing the robot to come closer. These properties are treated in the current system by a "safety grid". This grid contains a human centered Gaussian form of cost distribution. Each coordinate (x, y) in this grid contains a cost inversely proportional to the distance to the human. When the distance between the human and a point in the environment (in the grid) $D(x_i, y_j)$ is greater than the distance of another point$D(x_k, y_l)$, we have $Cost(x_k, y_l) > Cost(x_i, y_j)$. Since the safety concerns lose their importance when the robot is far away from the human, the cost also decreases when getting farther from the human, until some maximal distance at which it becomes negligible.

Figure 3.5 shows a computed safety grid attached to a sitting/standing person. The height of the vertical lines represents the cost associated with each cell. As shown in the figure, human's current state (sitting, standing, etc) plays an important role in the cost of the grid. Also note that this approach allows us to consider other types of human states.

Once this grid is computed, searching for a minimum cost path will result in a motion that avoids moving too close to the human unless it is necessary. However, if the environment is constrained or if the task requires so, the robot is allowed to approach to the human. Only very close proximity of the human is strictly prohibited to avoid collisions.



(a) Sitting (b) Standing

Figure 3.5: A Safety grid is built around every human in the environment. It depends highly on the human's posture. As the person feels less "threatened" when standing, the value and the range of the costs are less important.

Safety on 3D Grid

The notion of safety is the absolute need of any human-robot interaction scenario and it gains a higher importance in manipulation scenario where the robot places itself close proximity of the human.

As farther the robot is from human, safer the interaction is, the safety cost function $f_{safety}(x, y, z, C_H, Pref_H)$ is a decreasing function according to the distance between the human H and object coordinates (x, y, z). The $Pref_H$ contains preferences of the function behavior according to human states like sitting or standing.

The cost of each coordinate (x, y, z) around the human is inversely proportional to the distance to the human. When the distance between the human and a point $D(H, (x_i, y_j, z_k))$ is greater than the distance of another point $D(H, (x_l, y_m, z_n))$, we have $f(x_i, y_j, z_k) > f(x_l, y_m, z_n)$. Since the safety concerns loose their importance when the point is far away from the human, once it is farther from a maximal distance, it becomes null.

The values of the Safety function is illustrated in figure 3.6 with 0.05m between neighboring points. It's clear that from a safety point of view, the farther the robot is placed from human, the safer will the interaction become.

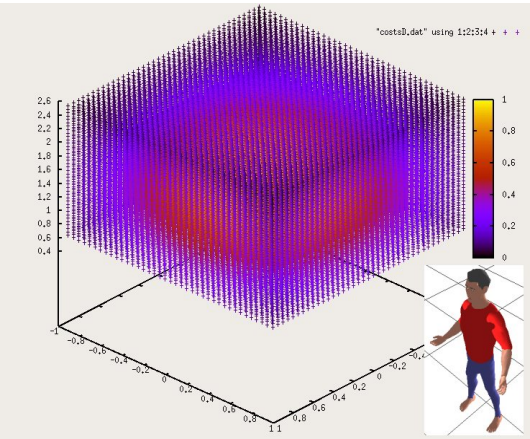


Figure 3.6: The costs of Safety function mapped around the human at 0.05m resolution. This function returns decreasing costs

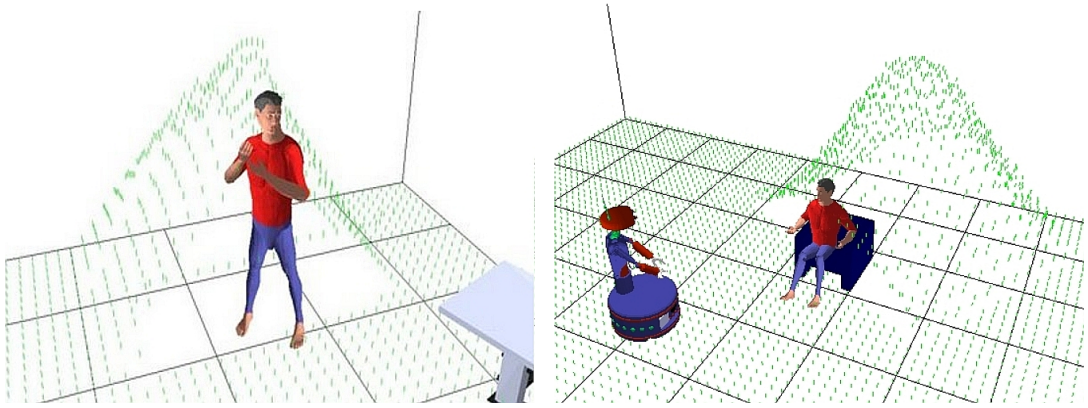
Visibility

Visibility on 2D Grid

The second criterion, called "visibility criterion", aims to improve human comfort during robot's motion. Humans generally feel more comfortable when the robot is in their field of view. This criterion allows the robot to be mostly in the human's field of view during its motions.

The resulting grid, namely "visibility grid", is constructed according to costs reflecting the effort required by the human to get the robot in his field of view. For example, grid points located in a direction for which the human only has to move his eyes have a lower cost than positions requiring him to move his head in order to get the robot in his field of view. Also, when the robot is far away from the human, the effect of the visibility must decrease. The computed visibility costs are shown in figure 3.7. The zone situated in front of the human has very low costs. On the

contrary, the zone situated behind the human has higher costs. Since the grid is attached to the head of the human, the computed costs are updated when the human changes his field of view (turn his head or his direction) during planning and/or execution stage.



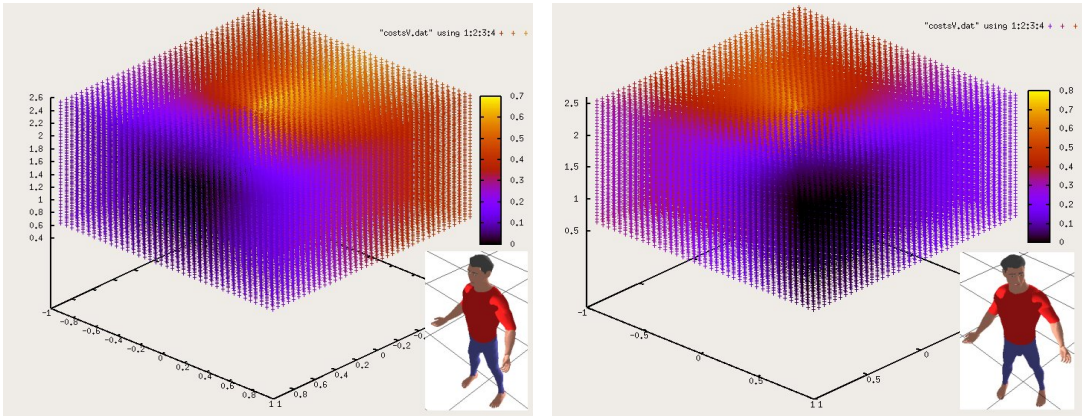
(a) Standing (b) Sitting

Figure 3.7: The visibility grid is computed by taking into account human's field of view. Places that are difficult for the human to see have higher costs.

Visibility on 3D Grid

The visibility of the object is an important property of HR manipulation scenarios. The robot has to choose a place for the object where it will be as visible as possible to the human. We represent this property with a visibility cost function $f_{visibility}(x, y, z, C_H, Pref_H)$. Alone this function represents the effort required by the human head and body to get the object in his field of view. With a given eye motion tolerance, a point (x, y, z) that has a minimum cost is situated directly in front of human's gaze direction. For this property, the $Pref_H$ can contain the eye tolerance for human as well as any preferences or disabilities that he can have.

The values of the Visibility function is shown in figure 3.8 with 0.05m between neighboring points. We can see that points at direction of human's gaze have lower costs. The more the human has to turn his head to see a point, the higher the cost will be.



(a) Looking right in front (b) Looking at his left

Figure 3.8: The costs of Visibility function distributed around the human with 0.05m resolution. The points that are difficult to see have higher costs. The visibility function depends also on the direction of human's gaze.

Arm Comfort

The last property of the placement of the object is human's arm configuration when he tries to reach to the object. The human arm should be in a comfortable configuration when reaching the object. This property also is reflected by a cost function $f_{armComfort}(x, y, z, C_H, Pref_H)$ which returns costs representing how comfortable for human arm to reach at a given point (x, y, z). In this case $Pref_H$ value can contain left/right handiness as well as an other preference of which arm the human prefers.

The inverse kinematics of human arm is solved by IKAN[Tolani 00] algorithm which return a comfortable configuration among other possible ones because ofthe redundancy of the arm structure.

For a given arm configuration, the costs of Arm Comfort property is calculated by

$$f_{armComfort}(x, y, z, C_H, Pref_H) = min(\begin{array}{l} f_{LeftArmComfort}(x, y, z, C_H) + P_{left} \\ f_{RightArmComfort}(x, y, z, C_H) + P_{right} \end{array}) \quad (3.7)$$

where

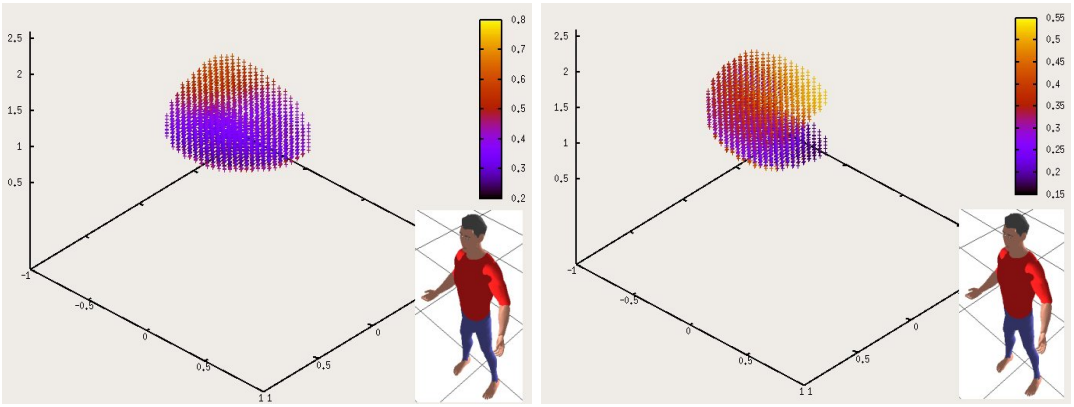
$$f_{Left/Right\ ArmComfort} = \beta_1 f_{displacement} + \beta_2 f_{potential} \quad (3.8)$$

$$f_{displacement} = \sum_{j=1}^{n} (\theta_{rest,j} - \theta_j)^2 \quad (3.9)$$

and where θ_j is a joint angle of the jth joint, n is the number of arm joints, θ_{rest} is angle of the joint in the rest position and P_{left},P_{right} are the penalties coming from left/right handiness.

The minimization of this function will find point where the combination of the joint displacement and arm's potential energy is minimum which is an important property for human arm comfort[Marler 05, Katayama 03, Abdel-Malek 05]. Note that this is not a dynamic minimization as for example minimum-jerk techniques [Broquère 08].

The Arm Comfort functions for left and right arms are illustrated in figures 3.9-a and b. Note that only the accessible and more comfortable point are shown in these figures. All other points are evaluated as not comfortable and their costs are higher.



(a) Left arm confort (b) Right arm confort

Figure 3.9: Arm Comfort function for a left handed person. Even tough the shape of left (a) and right (b) arm functions is the same, a penalty is applied to the right arm thus increasing its cost. Note that only the accessible and more comfortable point are shown. Other points around the human have the highest cost.

Merged Grids

All the grids mentioned above are separated criterions of the human model. In order to take into account every aspect of these grids we have to merge the specified criterions in one unique grid depending on the task problem.

If the robot has a navigation task where it has to pass close to the human or where the robot wants to approach the person, then the 2D grids are fused into the 2D "merged grid". In other case, if the robot has to know where and how to place one of its extremities for a close interaction with the human (i.e. to give an object to the human), then the robot must use the 3D "merged grid".

2D Merged Grid

Once the safety, visibility and hidden zones 2D grids have been computed, they are merged into a single grid in which the robot will search for a minimum cost path. Different methods can be used to merge the grids. A first way can be to compute the overall cost from the weighted sum of the elementary costs:

$$\begin{aligned} f_{merged_{2D}}(x,y) &= w_1 f_{safety_{2D}}(x,y) \\ &+ w_2 f_{visibility_{2D}}(x,y) \end{aligned} \tag{3.10}$$

where (x, y) is a point in the grid, w_1 is the weight of the safety grid and w_2 is the weight of the visibility grid. Another way is to consider the maximum cost values when merging the grids:

$$f_{merged_{2D}}(x,y) = max(f_{safety_{2D}}(x,y), f_{visibility_{2D}}(x,y)) \tag{3.11}$$

3D Merged Grids

The 3D merging method is similar to the 2D merging method but with the parameters of "safety", "visibility" and "arm comfort". The total merged cost of a 3D cell $f_{merged_{3D}}$ is computed as follows:

$$\begin{aligned} f_{merged_{3D}}(x,y,z,C_H,Pref_H) &= w_1 f_{safety_{3D}}(x,y,z,C_H,Pref_H) \\ &+ w_2 f_{visibility_{3D}}(x,y,z,C_H,Pref_H) \\ &+ w_3 f_{armComfort_{3D}}(x,y,z,C_H,Pref_H) \end{aligned} \tag{3.12}$$

3.2.3 The Human's Interaction Area

In addition to share the space with the human in a "proper way", while performing its "domestic" tasks, a *robot companion* has the mission of being into a direct interaction with human. In order to interact with a person, the robot has to find how to place itself in a configuration where it has direct visual contact with this person. This constraint helps to reduce the search space to find such a point.

This region is similar to the "approach area" of an object. The main difference resides on the meaning of "interaction" and more precisely the interaction with human, put differently, in a bidirectional action exchange with communication. A robot cannot perform the interaction if the person doesn't notice its presence. Moreover, the human can do movements that can be dangerous for him when he hasn't notice the presence of a robot out of sight. These actions are examples of how the robot can violate the criterions of comfort and security for the human, explained on this section.

For all these reasons the search of the "Interaction configuration" must take into account the person's perspective. In other words, the robot has to be inside of the person's field of view, where the person can see it. I have represented the human FOV and the whole interaction area by using the flat method (represented as a semicircle area in front of the human).

At the same time, the field of view area can be divided in two sub-areas:

- **Security Area**: is the zone around the human that defines the minimal distance from the person position (Rad_{min}), to approach the person. In this way we can assure not to find a dangerous configuration for the human.
- **Interaction Area**: is the zone around the human where the robot can interact with the person. This area is defined by the radius range $Rad_{range} = [Rad_{min}..Rad_{max}]$. Where Rad_{max} is determined by the sensor capabilities in distance to perceive the elements of interaction (i.e. a person, a bottle), and on Hall's interpersonal distances mentioned in [Huettenrauch 06] depending on desired interaction task (e.g. proximity needed for only visual interaction or handing an object).

A subarea inside the interaction zone is the area under human attentional field of view, called here as "Attentional Area". It is defined as the part of the Interaction zone that is in front of the person in the angle:

$$\alpha_{view} \mid 0^\circ \leq \alpha_{view} \leq 180^\circ \tag{3.13}$$

α is the angle of the attentional field of view or UFOV[4] (Related to each person capabilities. Normal values: $[10^\circ - 30^\circ]$). Additionally to this natural value it can also include the saccade[5] map. Figure 3.10 shows each one of these zones and its subdivisions.

As it is mentioned in the definition of the approach area (see Sec. 3.2.1), the ample radius of the interaction area depends on the robot capabilities and on the nature of the task, so the size of the area can change depending on the task goal as it is illustrated on figure 3.11.

[4]Useful Field Of View: the part of the FOV where the human can extract visual information in a brief glance without head or eye movements, it decreases with the age

[5]fast motion movements

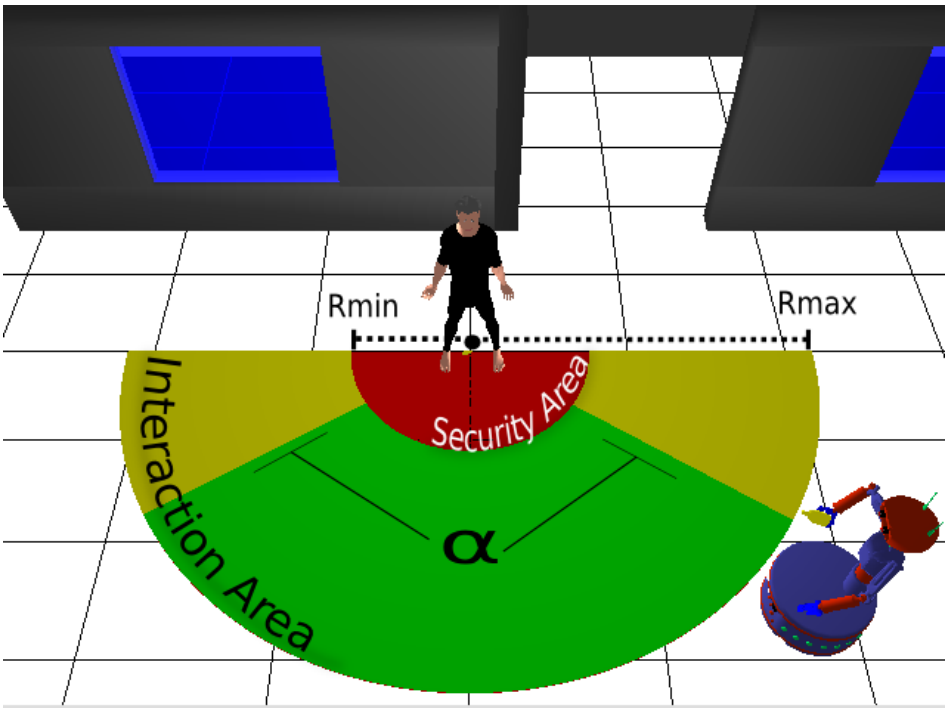
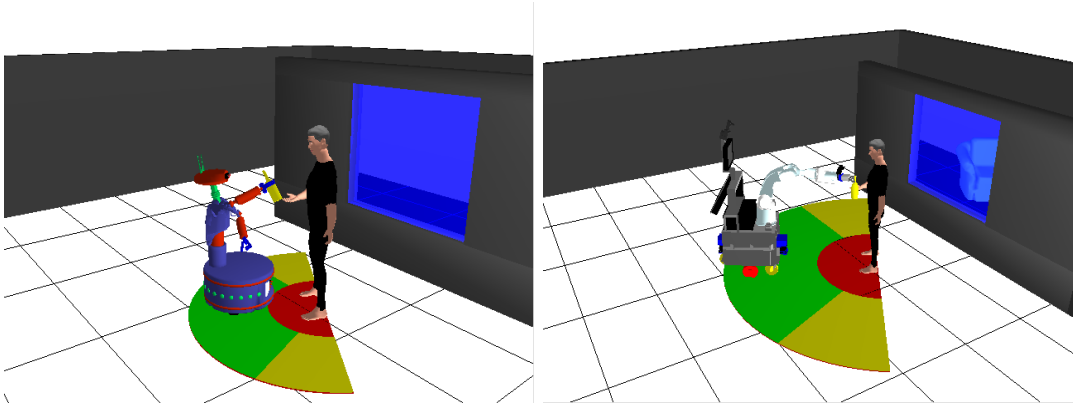


Figure 3.10: The "Interaction Area" inside the human's field of view. The FOV has its center on right in front of human orientation. Here the robot can place itself in order to interact with the human. The area under the human's UFOV or "attentional area" (α) marked with the green band in front of the human; in yellow the rest of interaction zone and; in red the security zone which is excluded from searching a configuration inside.



(a) Robot with short arms for a handing an object task

(b) Robot with different capabilities and different size for the same task

Figure 3.11: The size of the interaction area depends on the task and on the robot capabilities, as also previously illustrated for the "approach area" of objects in Figure 3.4.

3.3 Human Aware Motion Planner

It is necessary for a robot that will "co-exist" with humans, to take into account the way that it interacts with a person or that its presence is in complete harmony with the human's comfort and security. The robot has to perform motion and manipulation actions and should be able to determine where a given task should be achieved, how to place itself relatively to a human, how to approach him, how to hand the object and how to move in a relatively constrained environment in the presence of humans (an apartment for instance). Our goal is to develop a robot that is able to take into account "social constraints" and to synthesize plans compatible with human preferences, acceptable by humans and easily legible in terms of intention.

This section explains the motion planner designed mainly by Akin Sisbot [Sisbot 08] and conceived for the close interaction motions between humans and robots that must be taken by the robot in order to ensure:

- Safe motion, i.e., that does not harm the human,
- Reliable and effective motion, i.e. that achieves the task adequately considering the motion capacities of the robot,
- Socially acceptable motion, i.e. that takes into account a motion model of the human as well as his preferences and needs.

The interaction tasks that imply the robot motion will be divided in two problems: the navigation problem for the tasks that are related with displacement of the whole body of the robot; and the manipulation problem that concerns tasks of motions of extremities for a close interaction with human.

3.3.1 Navigation

The navigation planner uses the 2D merged grid. The weights of the grids to merge can be tuned according to the properties of the task. To find a path between two given positions of the robot, we search for a path in the final grid that minimizes the sum of the costs of the cells linking the cells corresponding to these two positions. The cells corresponding to the obstacles in the environment are labeled as "in-collision" and an "$A^\star$ search is performed to find a minimum-cost collision-free path linking two positions. The computed path is collision-free and also respects the human's care and comfort by taking into account safety, visibility and hidden zones.

Neither the final grid nor 3 criterion grids are constructed explicitly but the values of the cells are calculated for the ones explored during $A^\star$ search. As humans in the environment can change their positions and orientations often, avoiding explicit grid construction gives us the possibility to replan a new path if a change in the environment occurs (i.e. change in human positions, orientations, or states).

3.3.2 Manipulation

The presented approach is based on separating the whole problem of manipulation, e.g. a robot giving an object to the human, into 3 stages. Each of these stages will produce the corresponding result and pass it to the next stage, which consists in computing:

- Spatial coordinates of the point where the object will be handled to the human.

- The path that the object will follow from its resting position to the human,Äôs hand as it were a free flying object.

- The path of the whole body of the robot along with its posture for manipulation.

All these items must be calculated by taking explicitly into account the human partner to maintain his safety and his comfort. Not only the kinematic structure of the human, but also his vision field, his accessibility, his preferences and his state must be reasoned in the planning loop in order to have a safe and comfortable interaction, as we have described on all criteria of the human model design.

In each step of the items stated above, the planner ensures human's safety by avoiding any possible collision between the robot and the human.

Stage 1: Finding Object Exchange Coordinates

One of the key points in the manipulation planning is to decide where the robot, the human and the object meet. In classical motion planners, this decision is made implicitly by only reasoning about robot's and the object's structure. The absence of human is compensated by letting him adapt himself to the robot's motion, thus making the duty of the human more important and the motions of the robot less predictable.

The manipulation planner takes into account the three properties of interaction that will help the robot to find safe and comfortable coordinates of the object inside the 3D grid. This computed place will allow the robot to handle the object to the human. Each property is represented by a cost function $f(x, y, z, C_H, Pref_H)$ for spatial coordinates (x, y, z) ,a given human configuration C_H and his preferences $Pref_H$ when handling an object (e.g. left/right handiness, sitting/standing, etc.). This function calculates the cost of a given point around the human by taking into account his preferences, his accessibility, his vision field and his state.

Stage 2: Finding Robot Path

Even though we found a path for the object (and robot's hand) to follow, it is not enough to produce an acceptable robot motion. With this motion the robot must additionally make its intention clear.

The method consists of finding a path for the robot that will follow the object's motion. The object's motion is computed, as it were a free-flying object. But in reality it is the robot who holds the object and who will make the object follow its path.

To adapt the robot structure to the object's motion, we use the Generalized Inverse Kinematics [Yamane 03][Baerlocher 04] algorithm. Although this method is computationally expensive, it has some advantages:

- **Not dependent on the robot structure**: The Generalized Inverse kinematics method only needs a Jacobian matrix easily obtainable from robot's structure. This property makes this method easily portable from one robot to another.

- **Multiple tasks with priorities**: This method allows us to define additional tasks next to the main task. Therefore the robot not only accomplishes its task but also can take into account additional tasks during its motion.

- **Customizable according to various criteria**:Various costs, potentials or postures can be used as an additional criterion to the main task.

The planner has two tasks with different priorities to find an acceptable posture. In one hand we have a task with higher priority that contains the joints that affect the hand of the robot. This task aims to reach to a given point in object's path. On the other hand a task controls robot's gaze direction (camera joints) containing all the joints that affects to robot's head.

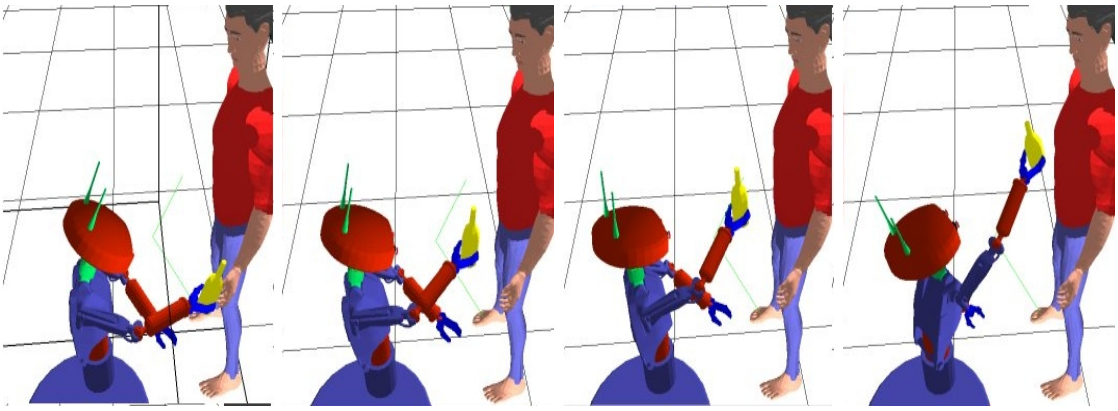
The general formulation[Baerlocher 04] of Generalized Inverse Kinematics with two tasks can be expressed as:

$$\Delta\theta = J_1^{+\lambda_1}\Delta x_1 + [J_2 P_{N(J_1)}]^{+\lambda_2}(\Delta x_2 - J_2(J_1^{+\lambda_1}\Delta x_1))$$

where J_1 and J_2 are the Jacobian matrixes of two tasks, $+\lambda_1$ is the singularity robust pseudo-inverse operator, Δx_1 and Δx_2 are goal points for two tasks, and finally $\Delta\theta$ represents the resulting step in the configuration of the robot.

With this method, the robot's posture is adapted to object's motion. Even tough the first task (motion of robot's arm) is enough for the manipulation scenario, the supplementary task of moving the head helps the robot express its intention clearly, thus makes the interaction more comfortable.

At the end of this stage we obtain a path, shown in figure 3.12 for the robot which is safe, visible and comfortable to the human as we took into account his accessibility, field of view and his preferences.



(a) (b) (c) (d)

Figure 3.12: Calculated path for this manipulation scenario. The robot looks to the object during this motion; with this behavior it shows its intention to its human partner.

3.4 PerSpective Placement (PSP)

As in every robot motion planning, the robot has to find a continuous path between its actual configuration (starting point) and its final configuration (destination point).

The Human Aware Motion Planner (HAMP) needs, for navigation purposes, to receive a precise final configuration where to compute the trajectory to displace itself in order to interact with human, to take an object or simply to observe something. On the other side, for manipulation purposes, the planner needs to know from where start to search configurations in order to hand or take an object in a collision free configuration next to the human where the robot can have an "eye contact" with this person.

I consider that the way of attacking the problem of the robot's placement decision will depend on different factors based on our human and environment model (see Section 3.2):

- The task to accomplish: The positions may differ if the robot has to search on a table, to pick an object on a shelf or to tell the human that the coffee is ready.
- The interaction target: The robot will not approach on the same way if the target is an object or if it is a human, or if the person is sitting or standing.
- The structure of the robot: Slim or thick, long arms or short, camera lens type, etc. The position will depend on the robot structure and on its capabilities.

Between these characteristics, we can observe that the robot's perception on this placement is an important element to take into account. The robot's "egocentric perspective-taking" will serve to estimate its own perception in specified positions and to determine if the task can be accomplished. For example, if the robot has to pick an object on the table, then the first step will be to choose a position where it can perceive the table.

Furthermore, if the intention of task is the interaction with the human, then an addition factor to consider is the Human's perspective. To perform a perspective-taking from the human and choosing a placement inside his field of view where both actors can have, the mentioned, "eye contact", will allow the robot to obtain an interaction in a "natural" or "social" way.

Perspective Placement (PSP) is a planner that produces the "where to go" configuration to the navigation planner, and the configuration where the manipulation planner can search the giving or manipulation positions. These given configurations will respect the constraints of human security and comfort, including the constraints of perception as for the robot itself as for the human.

Explained differently, **PSP** plans robot configurations taking into account a more general aspect than simply "visual" constraints, by also considering where it is possible to perform motion planning tasks. In a more general formalization, it can be introduced in the motion planning *CSpace* as a set of configurations that allow to satisfy a property. This property can be obtained by the geometric computation of different aspects like perception, object manipulation, feasible path. These aspects gain importance when the human is present on the environment, because the robot has to adapt all its actions to take his presence into account.

In this section it is described the method that I have defined for PSP, which is the main contribution of this work, this method will choose a configuration from a set of positions. These positions are obtained inspired on motion planning sampling techniques [Latombe 91], based on efficient estimation of a sample. In the first part, we will explain the sampling method followed

by the description of the evaluation parameters of each sample expressed by costs and qualities. We finish by explaining additional evaluation parameters based on task goals.

3.4.1 Position Points Generation

In the object modeling problem (see Section 2.2.3), the search for configurations or sensor positions is performed by testing places around the object that is being modeled, and with the sensor oriented toward this object. The proposed approach is based on this principle, the main difference is that the robot will not try to model its objective; here the robot will search for a place where it can perceive the object or the human whom it is going to interact.

Once the approaching and/or interaction areas defined, a set of points inside the target's zone will be generated in order to discretize the search. Either surrounding the object or in front of the human, these points represent 2D Cartesian coordinates of the robot center position on the environment. The points are contained on a circular or semicircular grid of n layers by m segments.

Inspired on the user studies on human robot interaction [Dautenhahn 06, Huettenrauch 06], on motion planning techniques, and on the computer graphics approaches (Sec. 2.2.2), we consider that each position should follows the next six basic properties:

- **Collision Free**: Robot in this position must not be in collision neither with objects and persons nor with itself.
- **Sensor Oriented**: Selected sensors must be oriented towards the target (an object or a human) in order to perceive it.
- **Secure and Comfortable for the human**: Following grid properties of the human model criteria of security and comfort (see Section 3.2).
- **Minimal Cost**: Robot should find a position that minimizes the cost based on point distance from the robot position and proximity of humans on the environment.
- **Maximal Perception**: In sensor's acquisition, the target has to be perceived as much as possible.
- **Human Aware**: if the target is a human, the robot has to treat him as a person and not as any other object. The robot has to place itself in a position where it is possible to have a mutual perception, and based on user studies [Huettenrauch 06] of robot-human spatial placement, the robot has to find spatial formation with the person in a human acceptable way.

On the configuration search, we avoid to test all points that are inside obstacle zones on the environment grid representation. A second collision test is performed once the robot is placed on a desired point, validating like this the first property of a point.

Orienting the sensor to the target depends mainly on the robot structure, and also on which part of its body it has attached the sensor. Then, to find a robot configuration where the objective is inside its sensors FOV, the robot can move several body parts in order to find the desired configuration.

Finally, to validate the rest of the properties of the points, we have to determine the visual perception of the sensor (in our case a camera) and assigning costs to each point to select the one that accomplish the better with social criterions as also to reduce the robot path distance.

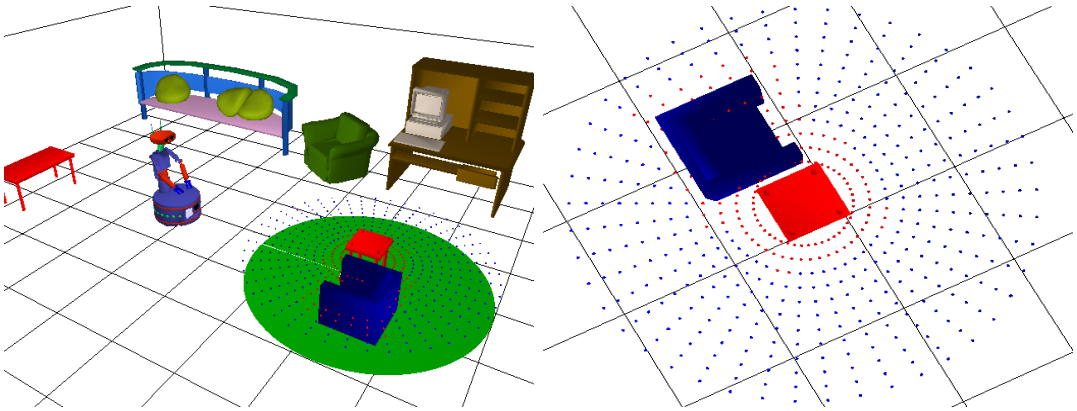
Approaching objects

On the methodology of perspective placement for an object, an array of points is generated all around it, inside the objects approaching area. In this manner, the robot can place itself all around object's position. Nevertheless, robot has to take into account its own perspective to look for the best placement configuration. Maximal and minimal ranges of the approaching areas are defined based only on collision avoidance and robot sensor limits.

Static Objects

Objects of the environment like table, sofas, desk, etc. Their position is often the same, even when we can move them to change their place. It is for this reason that this kind of objects is considered as static objects. Another characteristic of these objects is their 3D model, that is already known and that the robot can determine its structure, and the way it looks.

These models considered as obstacles for the navigation planner, can be the targets of our task (i.e. placing an object on a table), then the robot has to find positions in proximity of this object. In the figure 3.13 we can observe an example of a robot inside the approaching area around a table considered as known object, and the points generated to approach. The points are contained on a circular grid of n layers by m segments.



(a) Points of the target object inside its approach area

(b) Top view, the points are possible positions of the robot base center

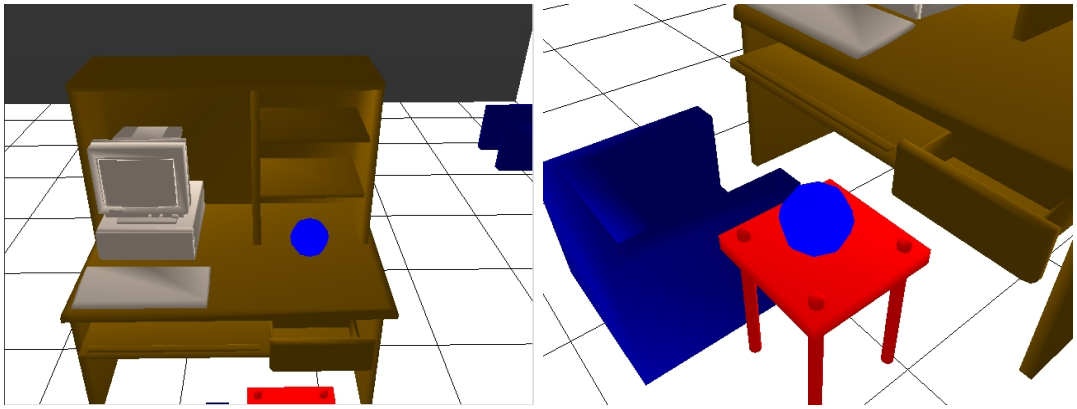
Figure 3.13: For objects on the environment the approach area is discretized by a circular grid around the target to search for a valid position where it can see the object (little red table next to the blue couch)

Movable Objects

Sometimes there are objects that can be manipulated in the environment that are small. They are not always at the same place and they can be hidden by another object. Here the object can be unperceivable to the robot, and it will need additional information to estimate the object position.

On the sentence "the cup is on the desk", the person is giving the robot important information that will help the robot search the "cup,Äù somewhere on the surface of the known object "desk".

In this case, our approach to this problem, as an extension of the previous one, is to create a simple geometrical object, called "search sphere", that could be set on possible places like the one referenced by human or on surfaces of known objects (e.g. on the top of a table). Discovering the referred object can be interpreted as seeing the search sphere.



(a) Search object on a desk (b) Search object on a table

Figure 3.14: Different search sphere predicted positions a) hidden by the same obstacle that holds the object (the desk) b) placed far from the robot.

Another example of how this approach can be useful is that sometimes algorithms or sensors are capable to distinguish objects and their estimated positions from far. The search sphere then, can be placed on object inexact location.

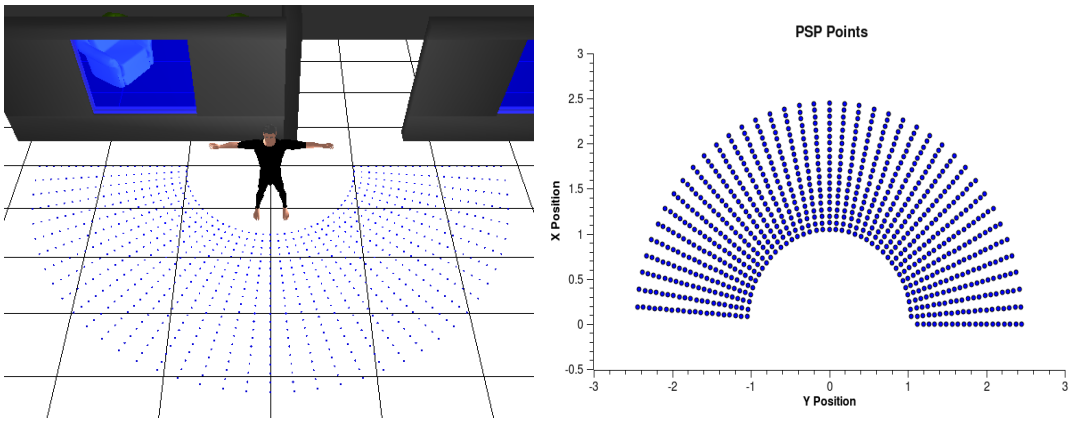
Due to the search sphere's shape and position, its approaching area and its costs functions are placed around the sphere as it is in for known objects. Maximal and minimal approaching ranges are defined by the specifications of the task. At the end, the robot will search between the generated points inside its approach area.

Approaching Humans

Approaching humans is not the same thing as approaching objects, the robot in this case has to take into account the human's perspective by placing itself inside the human's FOV, and with this obtain a mutual perception or "eye contact", crucial for obtaining smoother interactions.

The generation of possible positions of the robot will be similar to the methodology used to approach objects. In this case, the points will be generated inside the human's interaction area. The point,Äôs position will depend on the person's position, orientation (the direction of his gaze), his field of view, his preferences and his own capabilities of reaching the objects around him.

Different from objects, the points are contained in a semi-circular grid of n layers by m segments, as shown in figure 3.15. The layers and segments are homogeneously distributed through the interaction area, distance between layers is $\frac{Rad_{max}-Rad_{min}}{n}$ and, as the grid is in a circular form, the distance between segments will vary depending on the layer. For example, in a grid of 51 segments and 30 layers the minimal distance between segments is 7.5cm and the maximal is 25.1cm. The number of layers can depend on the size of the interaction area, for using the same distribution of layers. If the interaction area is in a fixed size then the point can be used to test the task goal achievement[6].



(a) Human and the points in its FOV

(b) Points generated around human, positions of the robot center

Figure 3.15: a) Person with all the points generated on the Interaction area. b) A zoom of the generated points, the robot will test determined points at this positions oriented to the center of the circular formation.

3.4.2 Determining Visual Perception

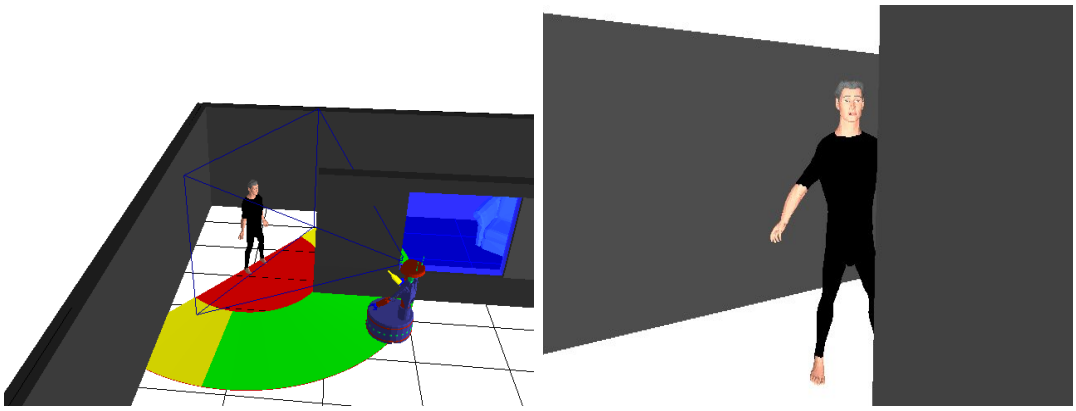
The robot needs to estimate what it is going to perceive in the different generated positions, in other words the robot needs to perform an "egocentric perspective-taking" to "imagine" its own perception on each of the configuration points. To achieve this, the robot has to have a model of its sensors capabilities in order to know WHAT and HOW it is going to perceive. In the presented work we are going to deal with the model of camera sensors.

To determine what the camera perceives, we use 2D perspective projection of the 3D environment. This projection is obtained from the sensor's position when the robot is placed in the desired configuration point. The obtained result is the matrix $MatP$ where the value of the position (x, y) represents one point in the target's projection image in sensor's field of view. A 2D projection is illustrated in the figure 3.16.

We define *Projection* Pr as the quantity of projection of a target element El (object or human) on the environment represented in $MatP$. Pr is obtained by:

$$Pr(El) = \Sigma MatP(x, y) \mid (x, y) \epsilon El \tag{3.14}$$

[6]We will see how to test the task goal achievement in Section 3.4.5



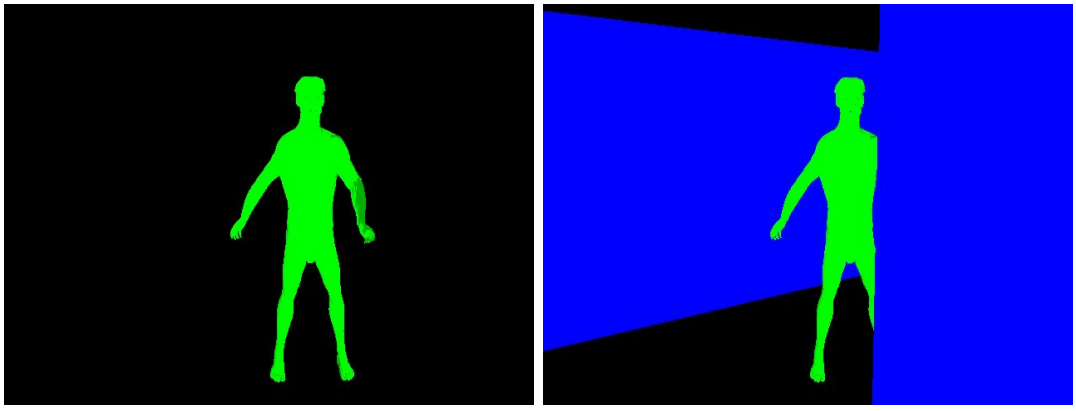
(a) Scenario (b) Perspective

Figure 3.16: In this scenario is illustrated that even when a robot is placed inside the interaction area of the human, it doesn't guarantees that the robot will perceive entirely the person as we can observe on the figure b).

The expected projection of an element that is not reflected on the matrix $MatP$ is known as Pr_{hidden} and it can be obtained with:

$$Pr_{hidden}(El) = Pr_{free}(El) - Pr_{visible}(El) \tag{3.15}$$

where $Pr_{visible}$ is the projection that considers visual obstructions (the "real" projection). On the other hand, Pr_{free} is the relative projection obtained without considering objects in the environment (as it should look without visual obstacles). In figure 3.17 we can observe the difference between free and visible relative projections.



(a) Free projection (b) Visible projection

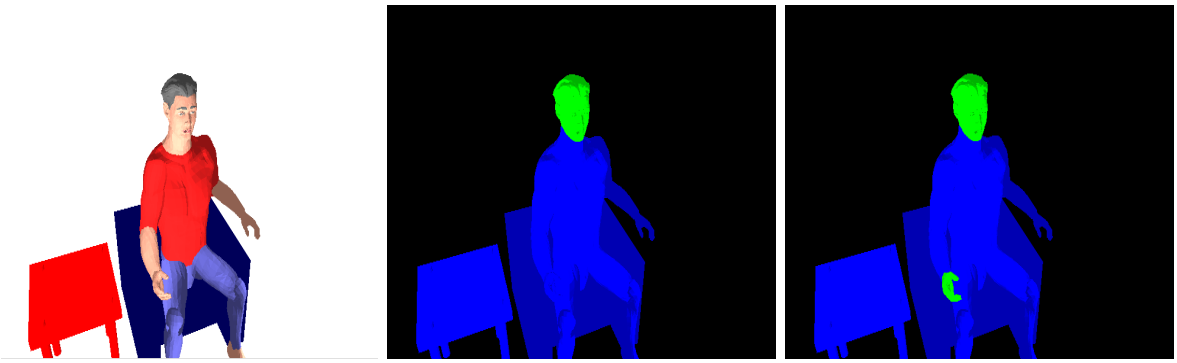
Figure 3.17: Relative projections. Here the person is the target and differs from other elements on the environment. a) Free relative projection b) Visible relative projection

Objective Ob quality visibility percentage defined by $Watch$ is determined by:

$$Watch(Ob) = \frac{Pr_{visible}(Ob)}{Pr_{free}(Ob)} \tag{3.16}$$

Finally, candidate configuration points for perspective placement may be filtered by: $Watch(Ob) \geq \mu$ where μ is a threshold that corresponds to a desired percentage.

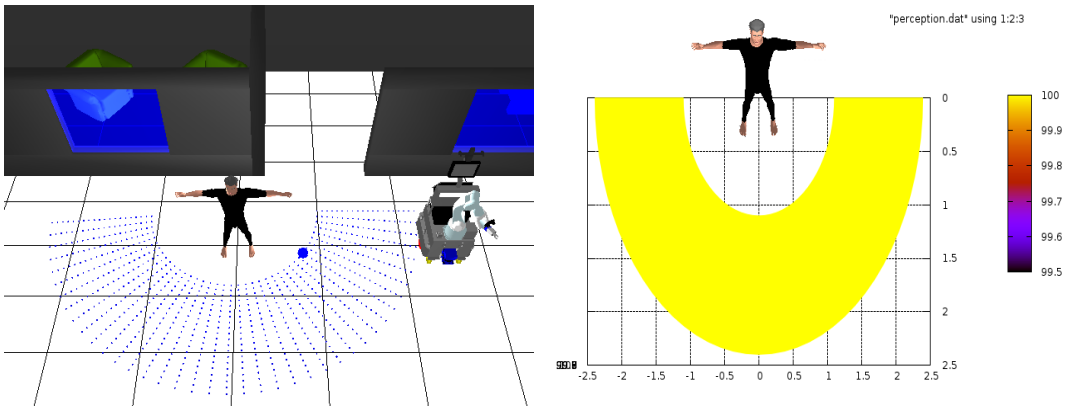
The perception elements can vary depending on the task, for example if we want to see only a part of the person like the head or head and hands, then we can only mark that our interest are only determined bodies, are shown on the figure 3.18



(a) Perspective of the robot (b) Searching for the human head (c) Searching for multiple human parts

Figure 3.18: The robot can search for the whole body or for single body parts of the human, depending on the task of interaction and on the perception algorithms.

On a scenario like the one shown in figure 3.19-a, that is free of visual occlusions, all the generated points have the maximal value of perception as shown in figure 3.19-b.

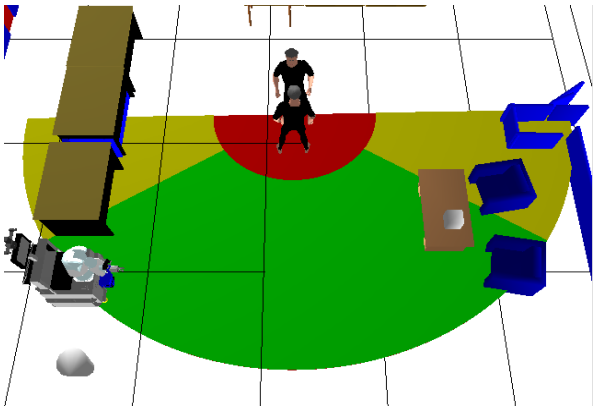


(a) Scenario 1: A Free Interaction area (b) Maximal quality values all around

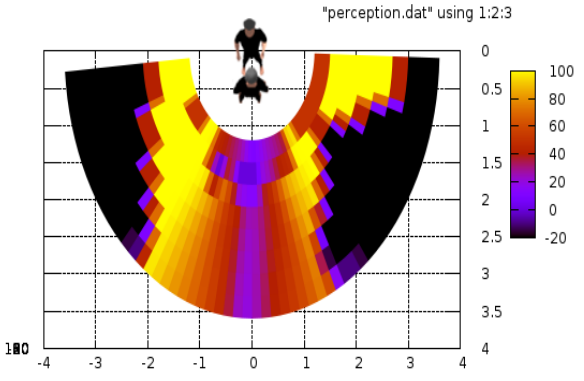
Figure 3.19: Scenario without visual occlusions, all points generated around the human have a total perception of the target. No matter where the robot place itself it is going to perceive entirely the human and viceversa

Otherwise, in a scenario like shown on figure 3.20, where there are two human talking one in front of the other, each point will contain a different quality values of perception. In addition, the obstacles prevent the robot to test different positions.

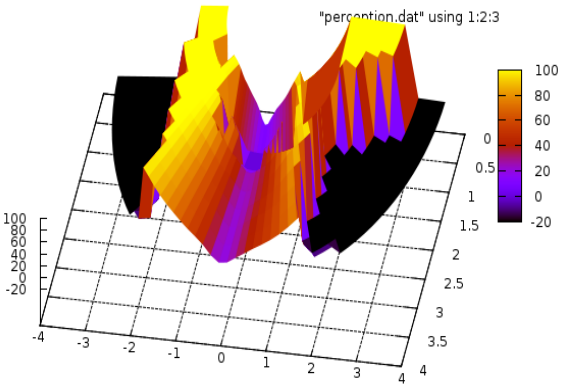
The perception can be different depending on the model of the cameras. To know how the perception is going to be, it is necessary to have the parameters of the sensors, that is the field-



(a) Scenario 2: "Two men talking". Here the target human is the one who is on the top of the image.



(b) Perception values, the values of perception change as it comes out from the visual obstruction caused by the person in front of the other target

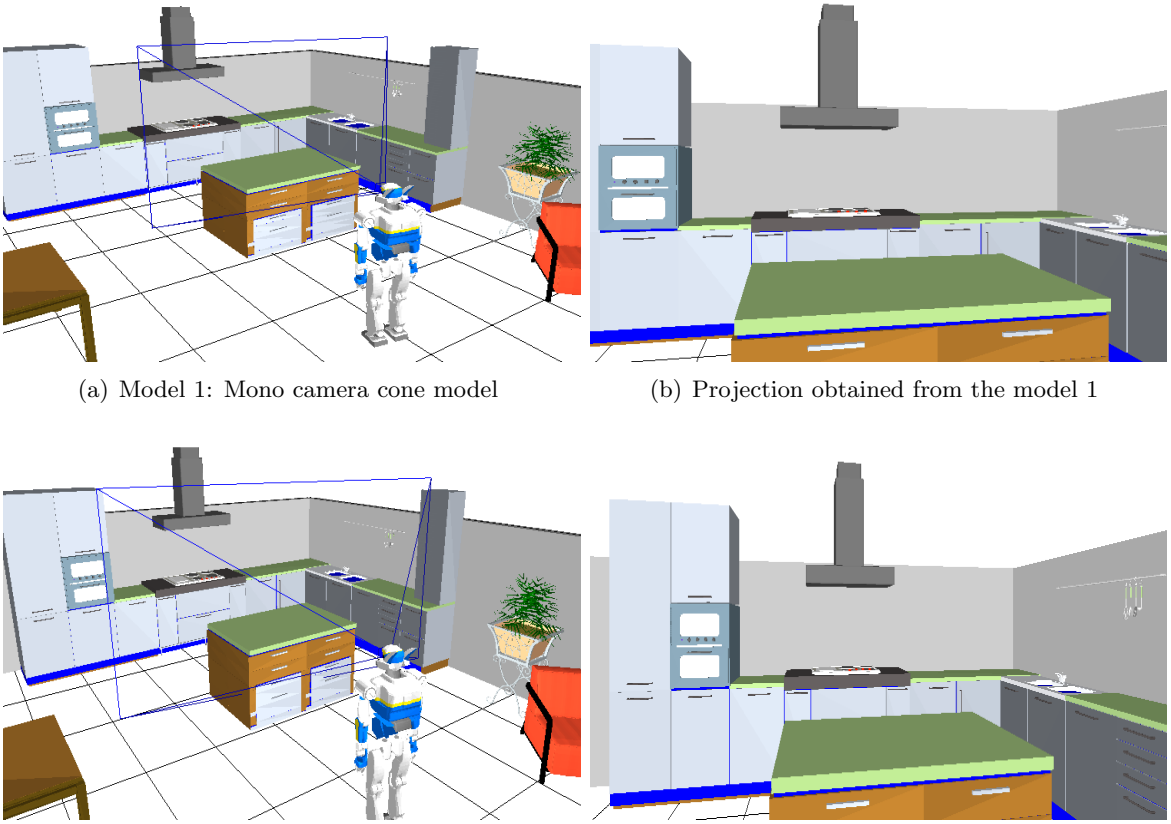


(c) Perception values (3D view)

Figure 3.20: Scenario with visual occlusions and with points on collision. Different perception values are assigned to each point, black zones on b) or c) are points on collision with obstacles on the environment.

of-view of the cameras. [7] The model consist of the vertical angle of view, the horizontal angle of view and it may consider the maximal distance of perception. It can be represented, as mentioned and implicit illustrated before, as a cone. The estimated projection will differ if the camera model changes, as it is shown on the image 3.21.

[7] It may also depend on the distance that the algorithms of perception are able to treat the information of the images.



(a) Model 1: Mono camera cone model

(b) Projection obtained from the model 1

(c) Model 2: Stereo camera cone model

(d) Projection obtained from the model 2

Figure 3.21: Different camera models provide different relative projections, model 1 and 2 varies on its cone horizontal angle, model 2 simulates the perception of two cameras inside the robot's head, while the model 1 obtains a smaller field of view from a mono camera model.

3.4.3 Determining Costs

The quality of the perception is an important aspect to know, but sometimes the value of a point will be the same as many other points, as for instance in figure 3.16 where there are no visual obstructions. Even when the perception value of some positions is the same, there are other aspects that can make the difference between them. For example: distance, it is not the same to choose one point at one meter than a point at five meters. For this purpose, the robot has to measure the cost of achieving a configuration point.

Costs for each point are obtained, as we have introduced before, based on different criteria obtained from minimizing robot path distance while taking into account human comfort. The basic cost evaluations are divided as follows:

- **Distace cost** ($Cost_{distance}$): distance from robot's actual position to the desired point position.
- **Safety and Comfort** - *humans on environment* ($Cost_{mergedgrid_{2D}}$): is the cost of the point from the computed 2D merged grid obtained from the human model based on human safety and comfort (See section 3.2), humans on the environment can affect the position of the robot.

Moreover, if the task of the robot is to interact with the human, then additional considerations must be taken into account:

- **Frontal Cost** ($Cost_{frontal}$): is the distance from the front of the human gaze orientation, to incite a $face\text{-}to\text{-}face$ human like position, and to encourage the robot to enter on the human visual zone with faster response time to events.
- **Attentional Cost** ($Cost_{atten}(x, y)$): is a fixed cost depending if the point is in the attentional area (defined by the attentional field of view) or not.
- **Safety and Comfort** - *human state* ($Cost_{mergedgrid_{2D}}$): Same cost than the basic one. To notice that is for the intention of respecting the human target state (Sitting or standing) and his proximity preferences.

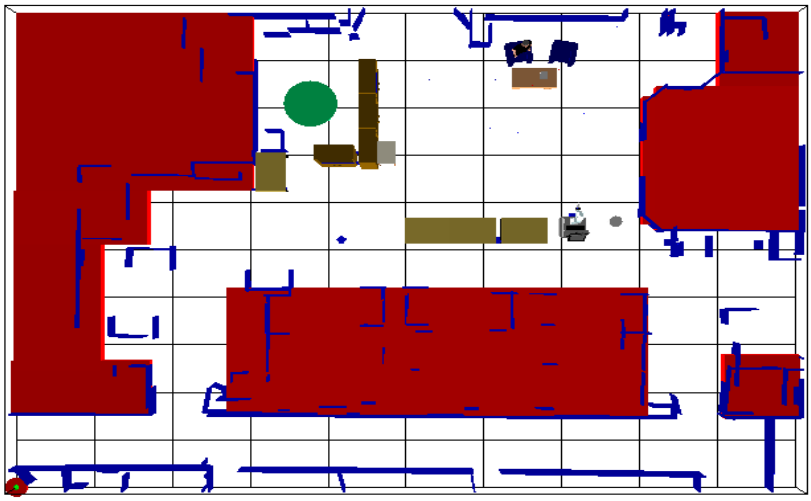
Each one of these costs contributes to help the robot to choose a configuration position depending on the task, the distance and on the human preferences. We consider that these costs have different priorities. In this work, these priorities are expressed as weights and their initial values have manually adapted from series of experiments[8]. These weights provide the flexibility of adapting costs to different human preferences.

Distance cost:

Distance is an important cost to analyze for optimizing execution time. This will also help to avoid the target person to get bored and loses interest on interacting with the robot.

Navigable zone obtained from the $G_{obstacle}$ and another grid containing the incremental distance values will be generated to obtain coordinates with distance values. This technique is inspired from a gradient method from motion planning literature. On this step each cell of the grid will have an assigned cost of distance including the distance of avoiding obstacles. In the "Grande Salle" environment shown in figure 3.22 the increasing costs obtained from the generated wave expansion, this grid will cover the whole navigable zone.

[8]This model is intended for further extensions as discussed on last section of this chapter.

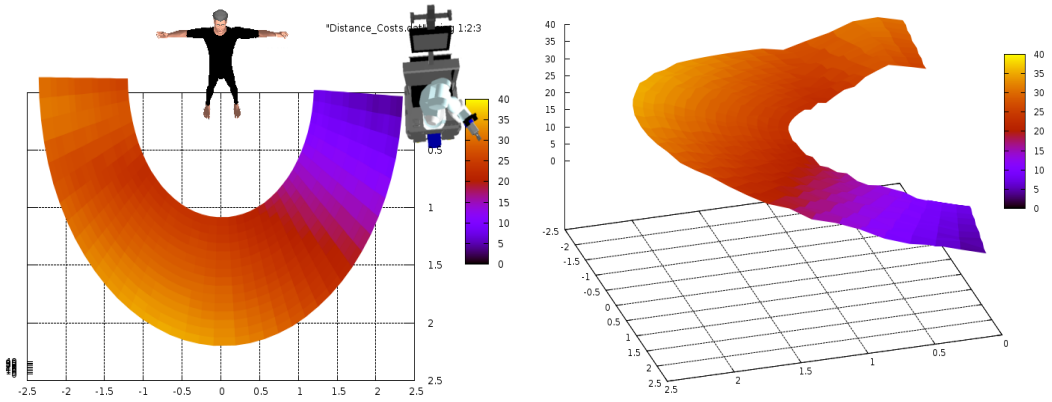


(a) Grande Salle environment

(b) Wave Expansion Grid, incremental costs from the position of the robot

Figure 3.22: The robot position (almost on the middle) shown in a) generates the minimum cost on the grid as shown in figure b and the distance costs will be incrementally growing all around until arriving to the environment limits. Black zones represent obstacles of the environment

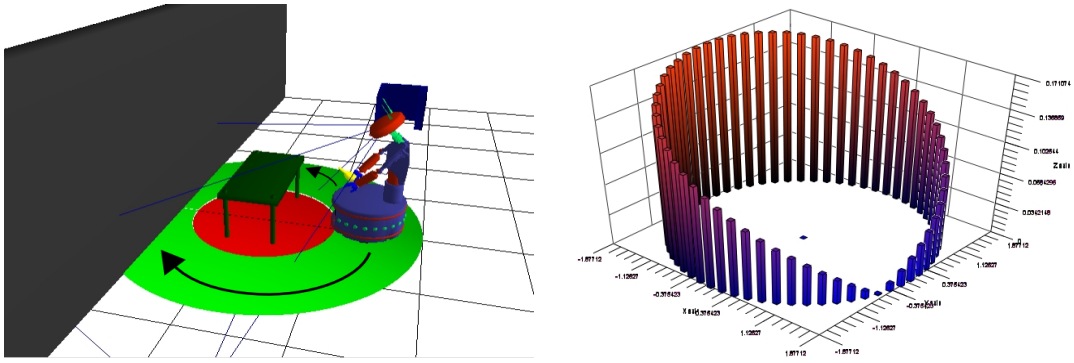
This method is executed once and before the generation of points, like this each point will take its corresponding distance cost value directly from its position. In figure 3.23 we can appreciate how this technique can help on measuring the distance to choose the position around a human.



(a) (b)

Figure 3.23: Distance costs increases in the way it gets farther from the current robot position

On the other side, for approaching an object, the generation of all the points is made around the object and not only the front, as done by human. The costs increment, as shown in Figure 3.24-b, rises from the robot position based manly on distance.



(a) Approach Area (b) Incremental costs around the "approach area"

Figure 3.24: For objects the area of interaction is called approach area, and is defined illustrated as the green band that turns the object all around. The costs will depend on the robot current position.

Frontal and Attentional costs:

In order to ensure a good interaction with human, the robot has to take advantage of all the characteristics of the human field-of-view. The farthest an event occurs from the center of the frontal gaze direction, the slower the human reaction time is, [Mizuhara 99] as illustrated on figure 3.25. Independently of this, one of the preferred Kendon's F-Formations on human-robot interaction is the "Vis-à-vis"[9] formation [Huettenrauch 06]. These two reasons take us to conclude that the robot should try to take preferentially a frontal position when it has to interact with humans. The formation between both agents can also depend on the task to perform, but for this work we have only expressed frontal formation as a cost.

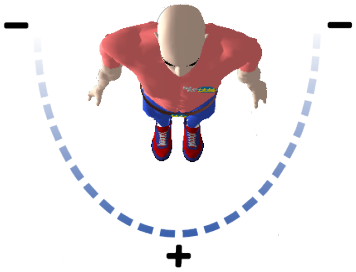
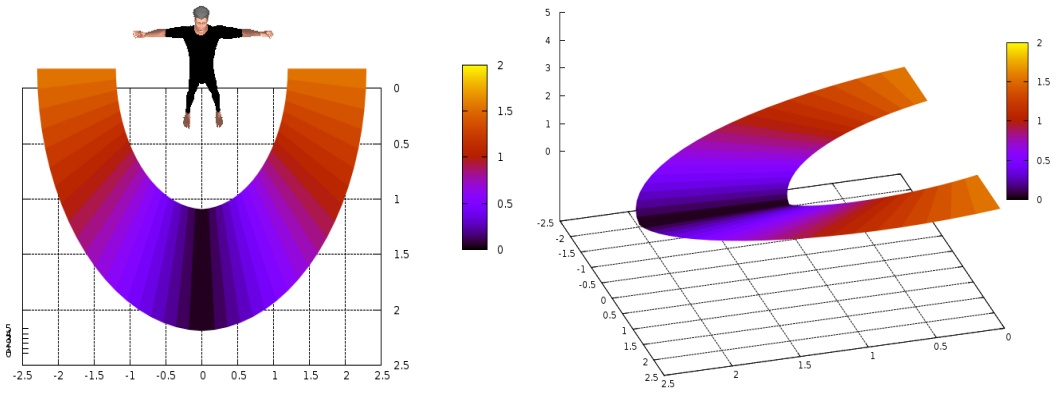


Figure 3.25: The farther an event occurs from the center of the frontal gaze direction, the slower the human reaction time is. The circular dashed line on the image shows the gradual decrement of the response time on the human vision.

To stir up the robot to take a frontal position, additional costs are introduced on the interaction area that gradually increase in the way the point get farther from the front of person's orientation. This cost is illustrated in figure 3.26.



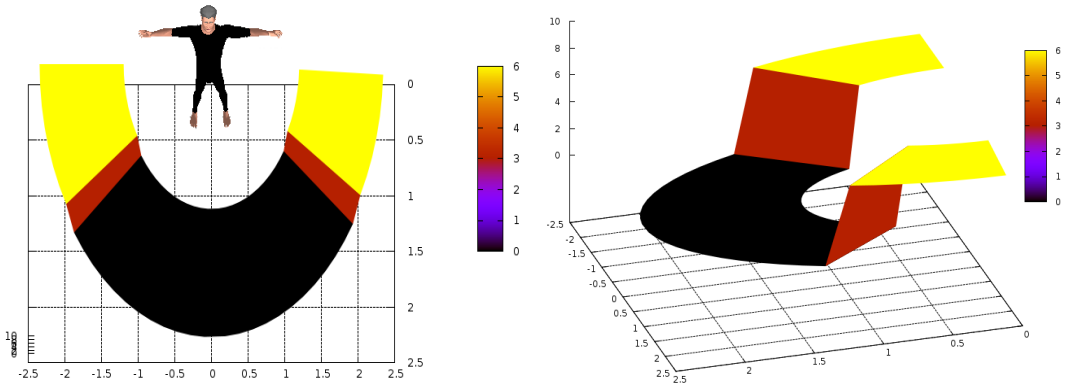
(a) (b)

Figure 3.26: Frontal costs increases in the way it gets farther from the face-to-face position

[9] a face-to-face formation in a communication task

Human field of view has a very wide range and, depending on the person, it can reach until 180 degrees of its horizontal angle. Nevertheless, attentional FOV is a restricted cone angle as shown in [Müller 05], where the human centers his attention, and sometimes without taking into account events that are not inside this particular area. That is the reason why the robot must try to be inside this zone, to call the human's attention avoiding to being ignored or to take by surprise the person. In this work, it is represented this area by the attentional area of the human model interaction area.

With the purpose of pushing the robot to be in the attentional area, the existing costs that are outside this zone will be boosted by "Δ" [10]. This intensification can be perceived in figure 3.27



(a) (b)

Figure 3.27: Cost of the point intensifies if it is out of the human attentional FOV

Safety and Comfort costs:

Security and comfort must be guaranteed in the last configuration of the robot, and not only for the human that is the target of the interaction, but also we have to take into account all the humans on the environment. Consequently, the generated costs of the human model (for navigation purposes) are considered as well.

The costs of "visibility" and "safety" (See Section 3.2) play an important role in the decision of the configuration point. For example, a robot will not choose to put an object on a table just behind a person that is on the environment, the human can step back and crash with the robot or even the person can be scare for being suddenly surprised. In the figure 3.28 we can observe the "aerobics room" scenario where two persons are in different parts and in a close proximity, the robot has to take into account every person on that share same place.

As illustrated in figure 3.29, the costs of being behind another human is much more higher than the other points independently on the distance that the robot has to follow. The priority of the cost generated by the 2D grids, is on the top of the list of costs.

Another aspect that we can obtain from the 2D grids is the possibility of taking into account the human state. With this information the robot knows that the human usually is less tolerant of a very close proximity of the robot if he is sitting; while he is standing the proximity threshold is lower. The cost of the grid then, increases depending on human target state.

[10] We have chosen the value of 1.3 to intensify the frontal value of 30%

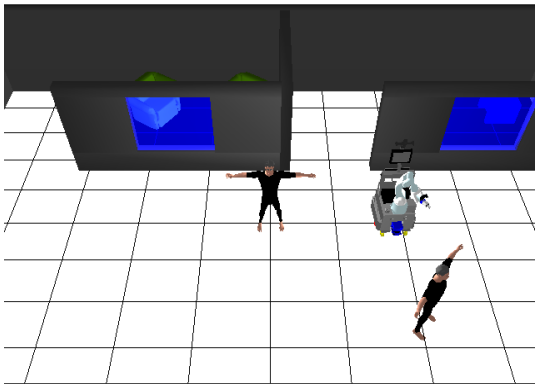
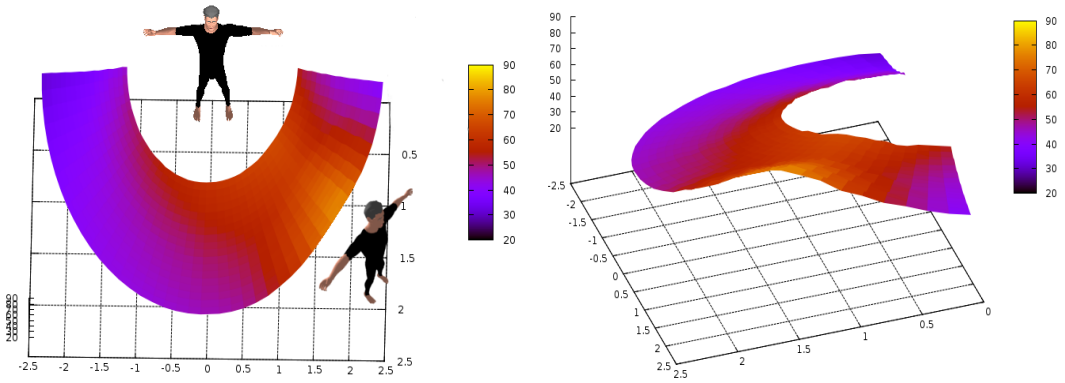


Figure 3.28: Scenario 3: the aerobics room, the robot has to take into account all the human in the environment and not only the one who it is going to interact.



(a) (b)

Figure 3.29: Human model costs prevents the robot for choosing a point to interact with the human of the interaction area, on a position that is just behind a second human. In this manner, comfort and security are guaranteed for any human on the environment.

Total cost:

At the final step, we can obtain the total cost of each point basically by adding all the costs that are related with the target. The values of all the costs are in the range of $[0..1]\epsilon R$. The elements of the addition for obtaining the costs of a target object are calculated as:

$$\begin{aligned} Cost_{total\,object}(x,y) &= (\lambda_{distance} Cost_{distance}(x,y)) \\ &+ (\lambda_{mergedgrid} Cost_{mergedgrid_{2D}}(x,y)) \end{aligned} \tag{3.17}$$

Where λ_x are the weight given to the x criterion and where:

$$\sum_{i=1}^{n} \lambda_i = 1$$

The computation of the total cost on the configuration points around objects varies from the sum of the total cost of the configuration points of the target human, where each point is calculated as:

$$\begin{aligned} Cost_{totalhuman}(x,y) &= ((\lambda_{distance} Cost_{distance}(x,y)) \\ &+ (\lambda_{frontal} Cost_{frontal}(x,y)) \\ &+ (\lambda_{mergedgrid} Cost_{mergedgrid_{2D}}(x,y)))\Delta_{atten} \end{aligned} \tag{3.18}$$

Including the frontal and attentional cost. λ_x value will depend on human preferences and it may be used on a learning phase that adapt the cost evaluation. In this work, we will use values that prioritize the costs as:

1. Safety and comfort: the most important aspects to respect in human-robot interaction is the safety and comfort.
2. Distance: The robot will normally go to the shortest distance, as in a "natural behavior"
3. Human perception: The robot will be close to the human attention to have an easier detection, but is not imperative reach the center.

An example of calculated costs is shown in figure 7.2.

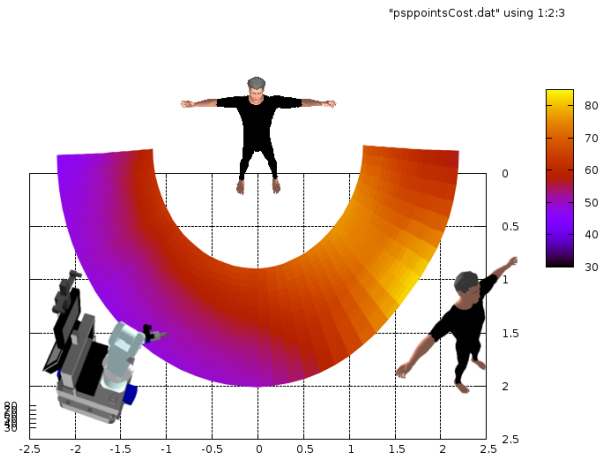


Figure 3.30: Computed costs of the points. a) Points in all around the field of view area, those points out of the attentional area have the highest cost. b) Points on the interaction area, lower costs are due to robot proximity and that are closer to front.

3.4.4 Position Evaluation and Selection

As it has been introduced before, to find the configuration where the robot can perceive its target, it is necessary to delimit the search space. The Interaction or approaching areas added to its discretization on a grid made the search of this point feasible (overall in terms of time).

The Evaluation

After the generation of all the points, we have to define a methodology to obtain the point that will be the one that the robot will choose. This methodology will vary between the quality/cost of each configuration and the computational time.

For evaluating a point we need to know two criterions: the quality of perception and the total cost of the point. It is necessary to obtain the maximum perception in the minimum cost. The evaluation value will be in terms of utility:

$$u_{x,y} = \alpha Q_{x,y} + (1 - \beta C_{x,y}) \tag{3.19}$$

Where u is the utility, Q the quality, obtained by the $Watch$ function of the Section 3.4.2. and C the total cost of the point x, y on the grid of generated configuration points. And where:

$$\begin{aligned} \alpha &= (1/Max_q)\Delta_1 \\ \beta &= (1/Max_c)\Delta_2 \\ &| \qquad\qquad \Delta_1 + \Delta_2 = 1 \end{aligned}$$

PSP is intended to find the best perspective of the target so that, the weight given to the perception part of the evaluation should be bigger. Nevertheless, as the planner is conceived for human-robot interaction, it cannot ignore the human security and comfort. For these reasons, the evaluation has to follow the next constraints:

$$\Delta_1 >= \Delta_2 \mid \Delta_1 > 0, \Delta_2 > 0$$

For our purposes we have defined $\Delta_1 = 0.6$ and $\Delta_2 = 0.4$.

The Search Methods

In this work we have implemented and tested three different point-search methods to choose the configuration that is best adapted for the interaction. The selection criterion varies between picking either the one that furnishes the best utility (quality of perception and cost) and selecting the least cost.

Best Configuration Search In this method the robot covers all the grid of points, evaluating each point by its utility. The algorithm of the methodology is as follows:

Algorithm 1 Best Configuration Search general algorithm

```
1: listPoints ← GeneratePointsWithCost()
2: while p ← GetNextPoint(listPoints) do
3:    p.quality ← GetQuality(p)
4:    if p.utility > 0 then
5:       p.utility ← GetUtility(p.cost,p.quality)
6:       if p.utility > bestpoint.utility then
7:          bestpoint ← p
8:       else
9:          if p.utility = bestpoint.utility & p.cost < bestpoint.cost then
10:            bestpoint ← p
11:         end if
12:      end if
13:   end if
14: end while
15: RETURN bestpoint
```

The costs are computed while the generation of the configuration points in the function *GeneratePointsWithCost()*, and assigned to the list l. In order to get the best utility with lowest cost, the algorithm compare first the utility and then in case of having the same utility it chooses the one that have the least cost.

The *GetQuality(p)* tests the configuration at point p and returns the quality of the perception or a negative value if the it is a not valid configuration. This function is explained on the next algorithm 2.

Algorithm 2 The test made by the GetQuality() function to each generated point. It returns the perception quality value or a negative value if it is not valid

```
1: if ¬inCollission(p) then
2:    if CanLookAtObjective(p) then
3:       qual ← p GetPerspectiveQuality(p)
4:       if qual > minPerception then
5:          RETURN qual
6:       else
7:          RETURN −3
8:       end if
9:    else
10:      RETURN −2
11:   end if
12: end if
13: RETURN −1
```

This function will be used by all the presented methods and it returns a different negative value depending of the fail reason. The function *GetPerspectiveQuality(p)* obtains the perception quality value of the configuration at point p, as explained in section 3.4.2.

This method is exhaustive the whole algorithm depends on the layers and segments defined. In other words its complexity is in $O(n)$ where n is the number of points generated.

Cost Based Search In this method, the points are sorted following their costs and it stops on the first one that accomplishes the quality minimal values. The algorithm 3 explains this method.

Algorithm 3 The Cost based Search general algorithm. It searches the in the list of generated points, sorted by cost, the first acceptable

```
1: listPoints ← GeneratePointsWithCost()
2: SortListofPoints(listPoints)
3: while p ← GetNextOrderedPoint(listPoints) do
4:    if GetUtility(p)> 0 then
5:       RETURN p
6:    end if
7: end while
8: RETURN null
```

Even when this method is far from searching the best point of the grid, it allows the robot to give a rapid answer to the needs of interaction. Its response time will differ depending on the number of points tested, in the worst case it will take the same time that the Best Configuration Search Method. The *SortListofPoints(l)* function sorts ascending by cost all the configuration points on the list l.

Random & Gradient Search (RGS) This is an adaptation of the Shotgun Hill climbing method to search the maximal utility configuration point. The search will starts at a random point and then will perform and local maxima (in gradient ascent) search towards the best neighbor value. After this, it will obtain non visited random points from the grid, compare the value of the obtained point, and if it is better utility than the last best point, it performs again the hill climbing method for getting the next local maxima. The process continues until δ points without detecting a better utility. The general algorithm is illustrated in 4.

Algorithm 4 Random & Gradient Search general algorithm

```
1: listPoints ← GeneratePointsWithCost()
2: while p ← GetNextRandomPoint(l) ∧ iter < δ  do
3:    p.utility ← GetUtility(p)
4:    if p.utility > 0 then
5:       if p.utility > bestpoint.utility then
6:          bestLocal ← HillClimb(p, l)
7:          bestpoint ← bestLocal
8:          iter ← 0
9:       end if
10:   end if
11:   iter ← iter + 1
12: end while
13: RETURN bestpoint
```

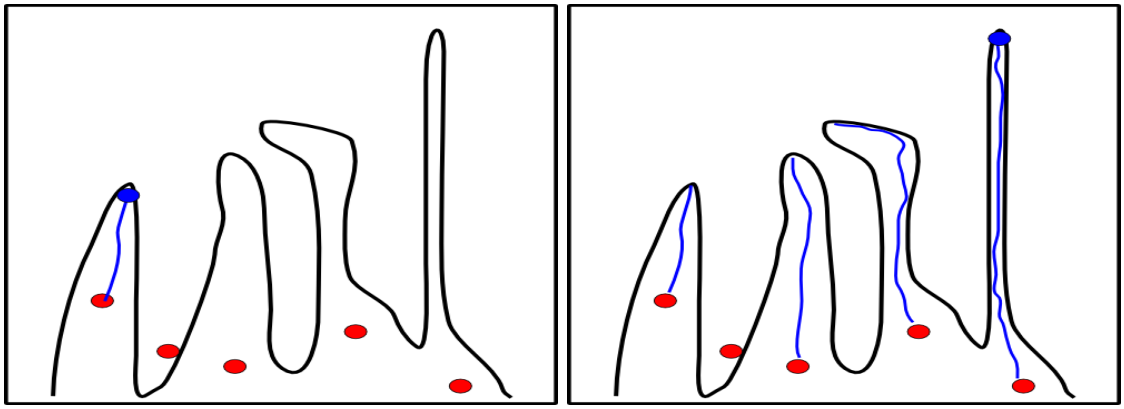
Here, the *HillClimb(p,l)* function is the one that test the neighbors and takes the best until there is on the local maximal utility.

A variant of this method is that for each obtained random point it will perform a *HillClimb()*, with this variant there are more opportunities to find the global maxima. Another difference with the previous method is that, this method repeats its process for fixed δ points.

Algorithm 5 Random & Gradient Search 2 general algorithm

```
1: listPoints ← GeneratePointsWithCost()
2: while p ← GetNextRandomPoint(l) ∧ iter < δ do
3:    p.utility ← GetUtility(p)
4:    if p.utility > 0 then
5:       bestLocal ← HillClimb(p, l)
6:       if bestLocal.utility > bestpoint.utility then
7:          bestpoint ← bestLocal
8:       end if
9:    end if
10:   iter ← iter + 1
11: end while
12: RETURN bestpoint
```

Figure 3.31 illustrates the same function with the same random points. The variant of the random and gradient algorithm can find the global maxima where the first method wouldn't be able to find. Nevertheless, the RGS2 will perform more point exploration than the simple RGS, which will be reflected on time.



(a) RGS (b) RGS2

Figure 3.31: The RGS2 is able to find better utility values on the same function with the same random points.

3.4.5 Reasoning about the Task Goal

As we have shown, approaching a target cannot be done by only selecting predetermined or random positions around this target. To make a good decision, the robot has to reason about the space where it is and also, the robot has to know the intention of the task and its well accomplishment. For example, the simple task "go to the table" can imply some actions that are not explicitly mentioned like go to the table and then look at the table. After this sentence is given, if the robot only goes to the table area without knowing the goal of the task then the chosen position can not be adapted to perform the next task that is "to look". Yet another question can appear, "to look? What for?" the task goals can be chained one after the other and so on.

However, not all goals are bound and not all binds are about space. In the "fetch and carry" scenario a phrase like "bring me the bottle" will imply two goal sections: the "bottle" section goals and the "me" section. The first section, refers to the task that are involved by the bottle, i.e. supposing that the bottle is on a table and the robot that knows that the bottle somewhere on this table, the robot has then to perform a sequence of actions: Go to the table, look for the bottle and pick the bottle. On the second goal section, we have the task related to the human that is asking for the bottle, with actions like: Go to the human, look at the human and give the bottle. These actions are goals that affect the selection of the position of the robot.

As we can see, the information obtained from the goal of a task can be very useful for resolving the problem of making a good choice of the destination point. The way that this problem has been treated along of this chapter, is by reducing or increasing the size of the *approach* or *interaction areas* based on the limits of the robot. Nevertheless, in some cases this information is not enough to know if the robot will accomplish its task or the manner that the robot performs its actions is comfortable for the human.

In order to obtain more information from the task, it is necessary to know its next mission on the same goal section. In other words, the robot has to obtain all the actions to be performed on the same place. On the examples mentioned above, we can observe that the egocentric perspective task (look for on/look at) is in aid support of the displacement task (go to), and that generally is followed of a manipulation task (give, take, pick)[11]. Based on this, I have divided the *composed tasks* as follows:

For objects:

Goto-Look it means to find a place to move where the robot can look in an ample range from the target object.

Goto-Pick Refers to select a position where the robot has to displace itself in order to look and then pick a movable object or to take something from the surface of an object.

And for interaction with human:

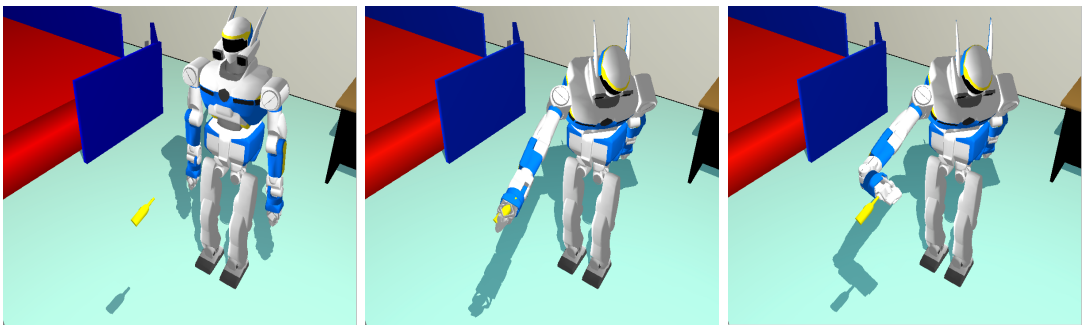
Goto-Talk The equivalent to look for the objects, but taking into account that it is going to interact with a human, informing the human, etc.

Goto-Give Give or take an object from an area inside the human's reachable range and preferences.

The composed tasks Goto-Look and Goto-Talk are examples of what has been explained along this chapter (Finding a configuration where the robot can perceive the target). The other

[11] On the context of the "fetch and carry" scenario on home environments

tasks are intended to find a position where it can not only perceive the target but also obtain a valid configuration to perform the task goals. To validate these positions we need to know the structure of the robot as its kinematic capabilities in order to find acceptable configurations that can accomplish the goals. For this intention, the manipulation planner must be used in order to obtain the maximal joint configuration on each position. For example, in figure 3.32 a scenario is shown where the robot has to take a bottle. By simple distance measure it is possible to determine if the robot is capable of reaching the bottle position. However, due to its structure, the robot is not capable of achieving a configuration where it is able to take the bottle (from this position).



(a) Robot initial position to take the bottle

(b) Achieving the target position

(c) The robot is not able to grasp the object, the grasp configuration is not reachable

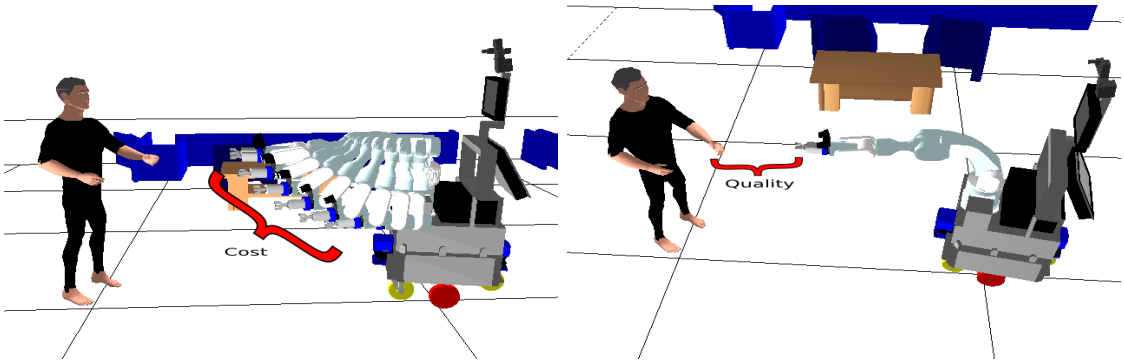
Figure 3.32: Added to the distance model a motion test must be performed in order to achieve a position that validates the task goal as better as possible. For this, it is necessary to obtain a kinematic configuration where the robot is able to accomplish this task.

Other example for considering the kinematic model can be conceived by an obstacle that can prevent the manipulator to take the object, even if the robot is able to see it (i.e. an object in an opened box).

In this section we aim to add an evaluation of the task accomplishment, in order to gather more information for supporting the configuration search. This evaluation is possible to conceptualize as an additional cost or as a quality of the position, based on the obtained configuration of manipulation. In this case we are going to consider the task achievement as both cost and quality, in other words, an utility. The cost of the task will represent the distance of achieving the goal position, and the quality will measure how close is the maximal configuration reached of the manipulator extremity to the desired goal position, the closer the manipulator is, the bigger the quality would be. The utility will be obtained in the general formula of the utility used on this chapter:

$$u_{taski} = \alpha Q_{taski}(q_f) + (1 - \beta C_{taski}(q_f, q_i)) \tag{3.20}$$

Where the quality of the task Q_{task}, of the task i on the maximal reachable configuration q_f, represents the proximity to the goal achievement, and the cost C_{task} refers to the configuration distance to accomplish q_f from the initial configuration q_i. For the experiments the values of the α and β are equivalent and $\alpha + \beta = 1$. Figure 3.33 shows the representation of the evaluation of the cost and the quality of the task goal achievement.



(a) Task goal cost: distance between initial and final configuration

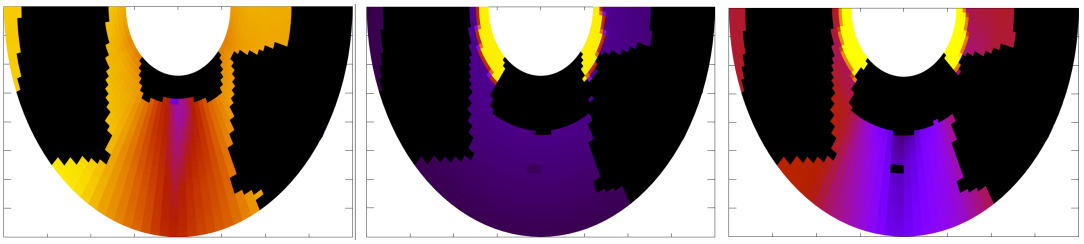
(b) Task goal quality: The distance value of the task achievement

Figure 3.33: Task goal utility evaluation by measuring the cost based on the configuration difference and the quality based on the proximity for accomplishing the task.

Once reached the end of the process of all the task goal utilities we can evaluate the total utility of a complete task by the average of these utilities, expressed as:

$$U_{TASK} = \frac{\sum_{i=1}^{n} U_{task i} \lambda_{task i}}{n} \tag{3.21}$$

Where a complete $TASK$ as GoTo-Look/Talk or GoTo-Give/Pick will be formed by an ordered sequence of $task$ as $TASK = \{task_1, task_2, ..., task_n\}$. Figure 3.34 illustrates the difference of each task goal utility and their resulted addition on the "Two Men Talking" scenario previously presented in figure 3.20 .



(a) Utility of GoTo-Look/Talk task

(b) Utility of GoTo-Give/Pick task

(c) Utility addition

Figure 3.34: "Two Men Talking" scenario of figure 3.20. Resulted utility function for different task goals, and that together form a different form on the interaction area.

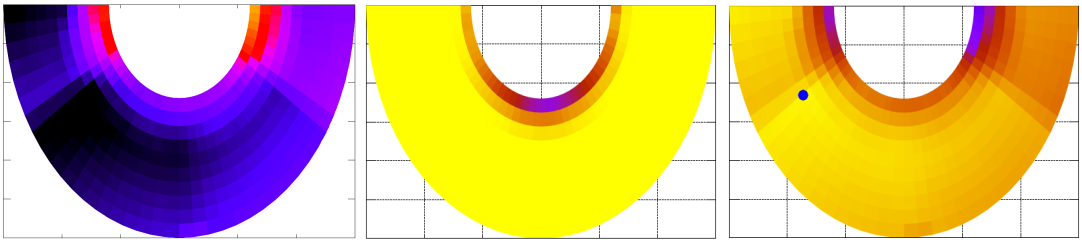
3.5 Simulation Results

The perspective placement system is implemented in C and integrated and tested within the Move3D [Siméon 01] software platform developed at LAAS-CNRS.

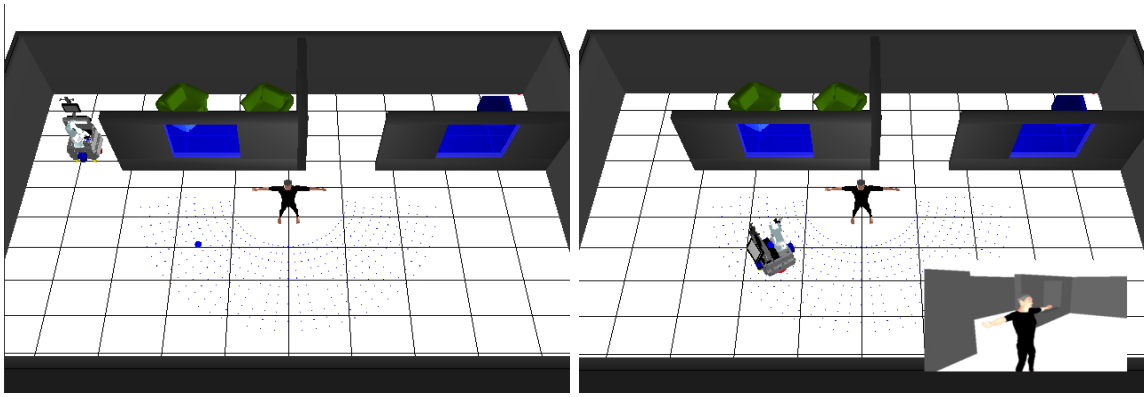
This section presents the different resulted configurations of the placement planner in different scenarios. The scenarios will vary in robot start position, human state, environment configuration, task to accomplish and configuration search method. The task goals of the robot will change between Goto-Talk and Goto-Give and will show that the configuration will change depending on its goal. PSP will be demonstrated on 2 different environments with different situations, which are: "Aerobics room", "Grande Salle".

3.5.1 Environment 1: The Aerobics Room

On this environment we show different robot placements when the robot has to interact with human in a Goto-Talk task, in two situations: alone with the human to whom the robot is going to interact, starting from different positions; and with four men on the same room with different postures of the second human.



(a) Costs (b) Quality (c) Utility



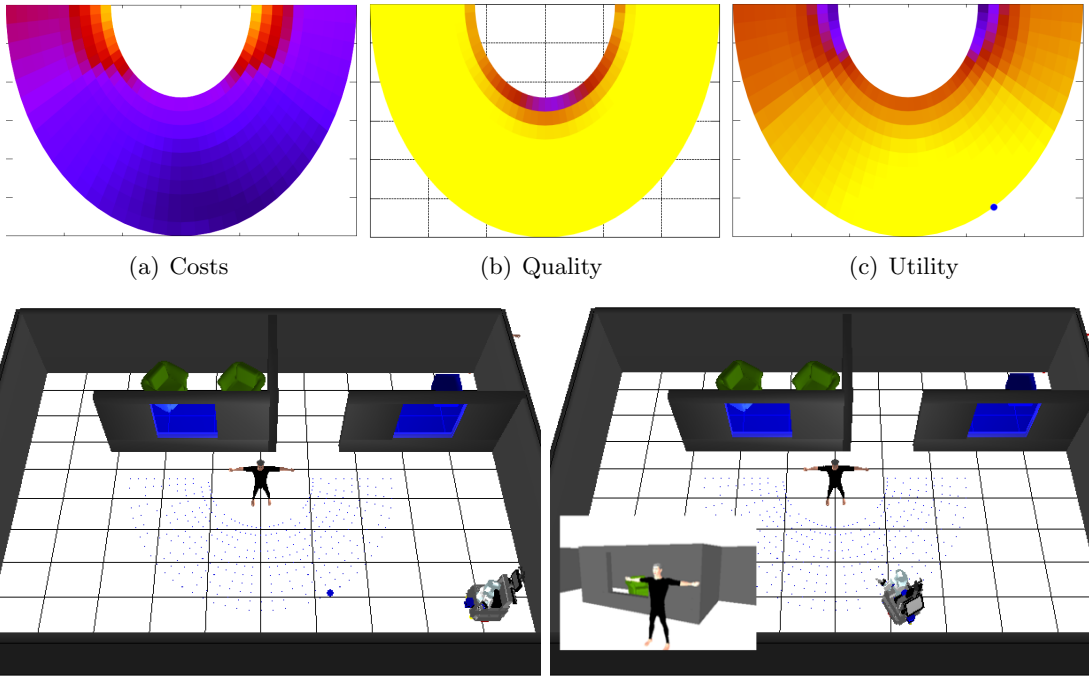
(d) Initial configuration on the robot position. Final marked on the blue sphere

(e) Final configuration and its perspective

Figure 3.35: Scenario 1-a: Human alone, robot coming from back. robot searches for the best position inside humans interaction area where there is no obstacle.

In Figure 3.35 the robot is coming from the back, it chooses a position on the limits of the visual attention that is not far from its current position. On the other hand, Figure 3.36 shows the result obtained from an initial position on the opposite side of the room. The robot selects a position that is best adapted to its current location, all this respecting the constraints imposed by the human presence. Blue point on the utility images, mark the positions that have obtained

the maximal utility on the interaction area. Costs and perception qualities of this area are also illustrated.



(a) Costs (b) Quality (c) Utility

(d) Initial configuration on the robot position. Final position on the blue sphere

(e) Final configuration and its perspective

Figure 3.36: Scenario 1-b: Human alone, robot coming from his front-left side. The robot adapts its final position depending on the initial configuration placement

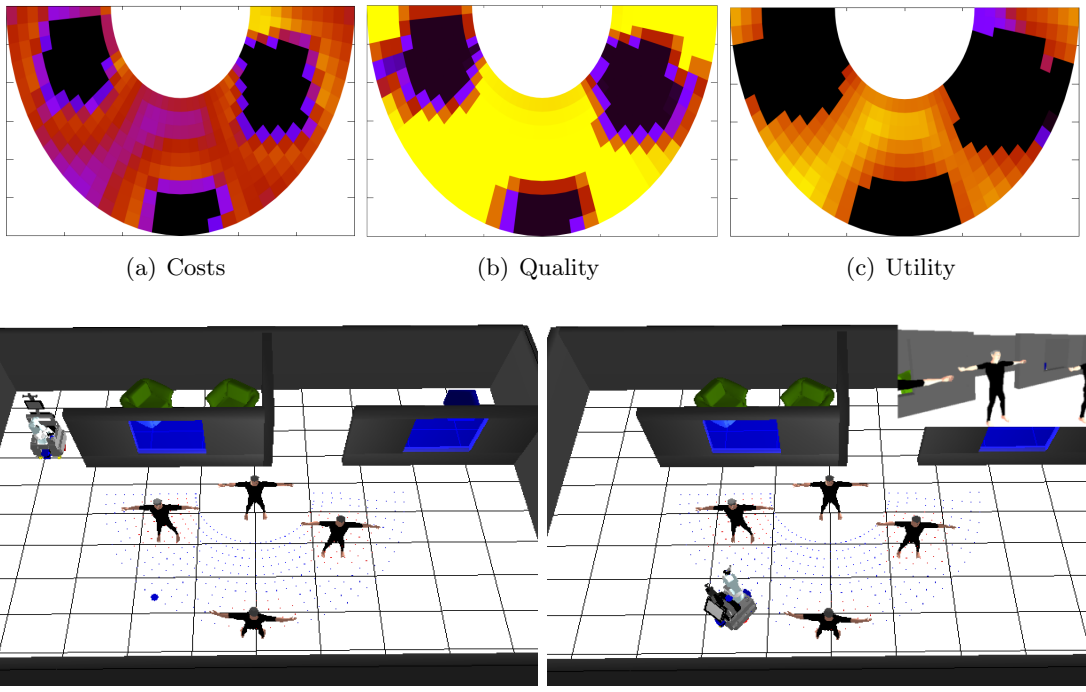
On the two previous scenarios, the person is alone and there is no constraint other than getting in to the interaction zone coming from different parts. Yet, when there are more than one human on the environment, and especially when they are close to the destination area, the robot has to consider the security and comfort of each of them. On figure 7.3 we can observe how the destination point obtained on the Scenario 1-a, is modified by the presence of additional persons on the environment. In addition, the position found is not only avoiding collisions with the humans on the interaction zone, but also considering their comfort as much as it is possible.

3.5.2 Environment 2: The grande salle

As we can observe on the scenarios of the previous environment, the utility of the selected point is influenced mostly by their costs and the collision avoidance, mainly because there are almost no occlusions of the target.

Figures 3.38 and 3.39 give examples of how PSP can find a configuration for known object (a furniture) in two different situations. In the first case with a person is blocking the obstacle and causing a visual obstruction, here the robot finds a position on the right side of the person, by avoiding it in order to perceive the table. Note that the robot computes a final configuration where it avoids not only collisions with human or obstacles but also avoiding visual obstructions.

On the second situation an extra person is occluding the table from the position chosen on



(a) Costs (b) Quality (c) Utility

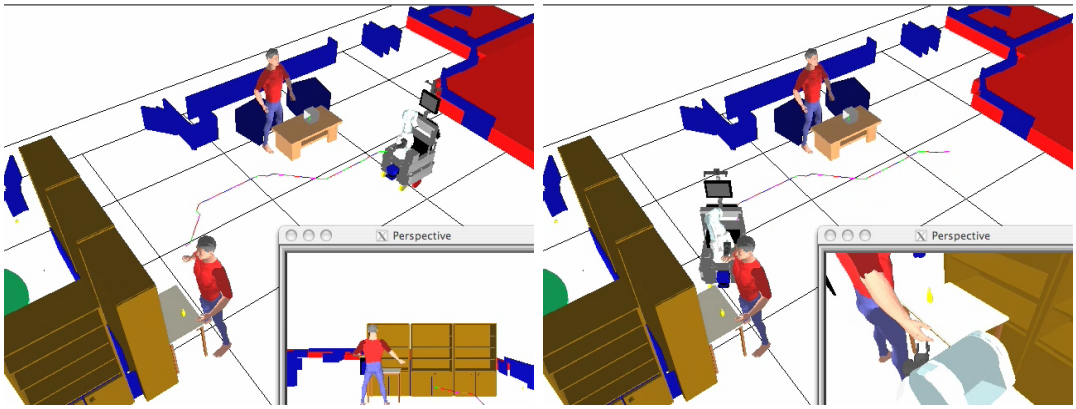
(d) Initial configuration on the robot position. Final position on the blue sphere

(e) Final configuration and its perspective

Figure 3.37: Scenario 1-c: The robot has to talk with the human on the middle, every person on the environment modifies the costs and the quality of the position. The robot finds the zone that respects the constraints of each one of the human on the environment.

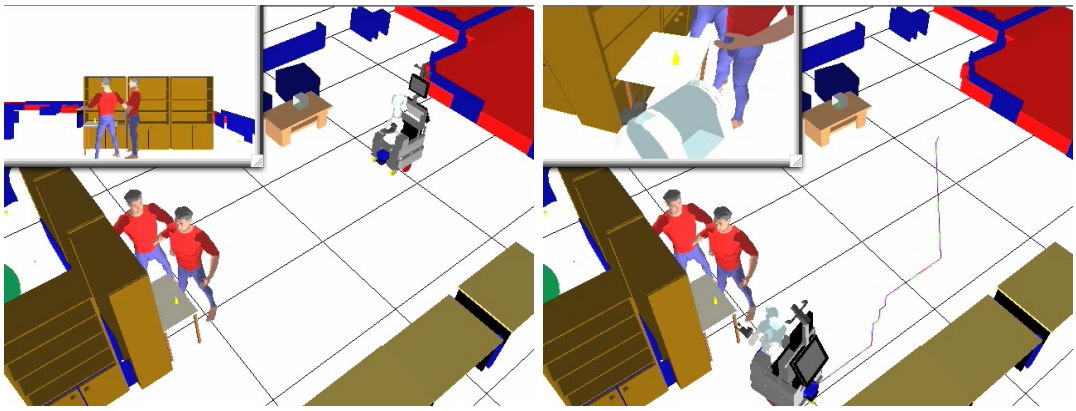
the first case, here the robot computes a configuration on the other side of table where none of the persons is present on the environment are visually blocking this table.

Results obtained on closer interaction (smaller area) shows that the interaction position depends on the human state (Sitting or Standing). As illustrated on figure 3.40, the robot chooses the farthest close position if the human is sitting for two reasons: first, the robot structure prevents the robot to perceive the human while he is sitting, cause by hiding the human with the robot's own arm; second, the information acquired from the security and comfort criterion entails that the robot has to be farther when the human is sitting.



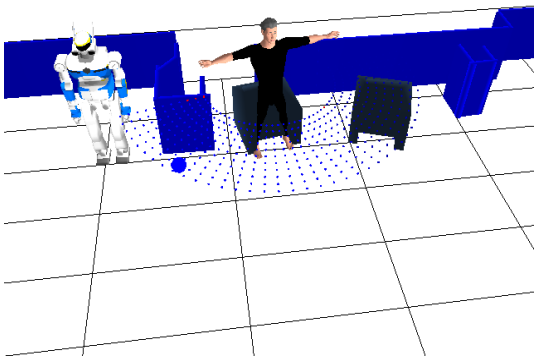
(a) Initial configuration (b) Final configuration

Figure 3.38: Obstacles are not only those that cause collisions with the robot on some configurations, but also those that prevent the target to be seen by the robot. In this scenario the robot intends to approach to look on the table, where one person is occluding the table, the robot takes the closest position from its initial position, but from where it is possible for the robot to perceive the table.

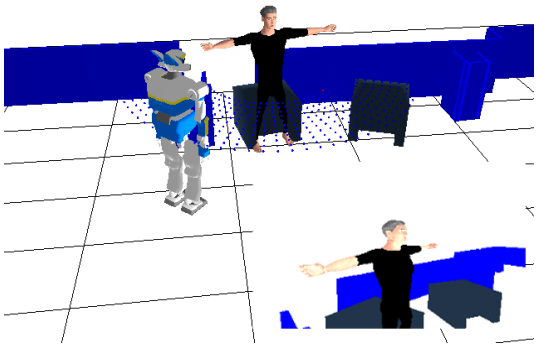


(a) Initial configuration (b) Final configuration

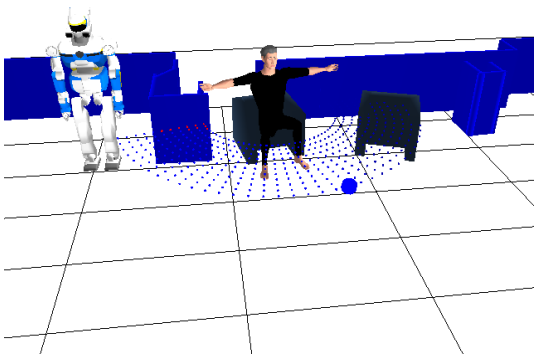
Figure 3.39: Approaching to look on the table from the same position and where there are two persons hiding the table. The robot finds another position that offers better perception of the table even if it is farther than the one chosen on the previous scenario.



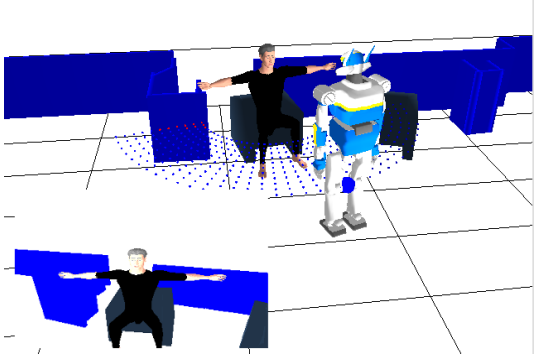
(a) Close interaction with human standing



(b) Final configuration and perspective (standing)



(c) Close interaction with human sitting



(d) Final configuration and perspective (sitting)

Figure 3.40: The human preferences change depending on its state, the configurations of interaction are adapted depending on the this state.

3.5.3 Performance Measures on Search Methods

As introduced before, we have tested 3 different search methods get the best point. For each point search method, there are some common parts that must be completed, all of them are dependent of the number of points n that are part of the discretization of the areas.

- *Costs computation:* This is strongly linked with the grid generation, it depends also on the definition of the grid but with the difference that the distance cost is computed once for all points.
- *Point quality test:* Each point-validation test takes about 0.03 seconds of CPU time, depending on the object size and on its occupancy on the desired projection matrix.

The total computation time will depend mostly on the search method by the number of points tested. We will illustrate the methods in two different scenarios shown in figure 3.41.and in figure 3.42

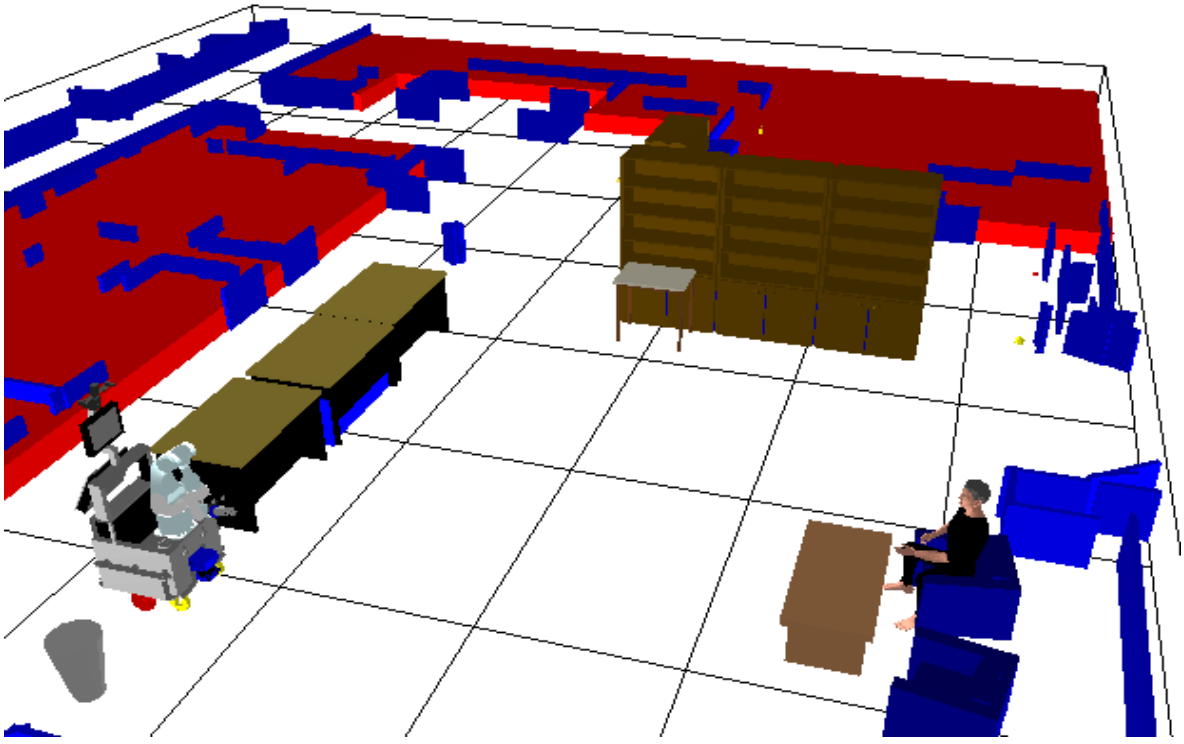


Figure 3.41: a) Scenario 1: Man sitting. The first scenario, it shows one of the simplest cases where there is almost no occlusion of the human target that is sitting on the couch, near to the small table.

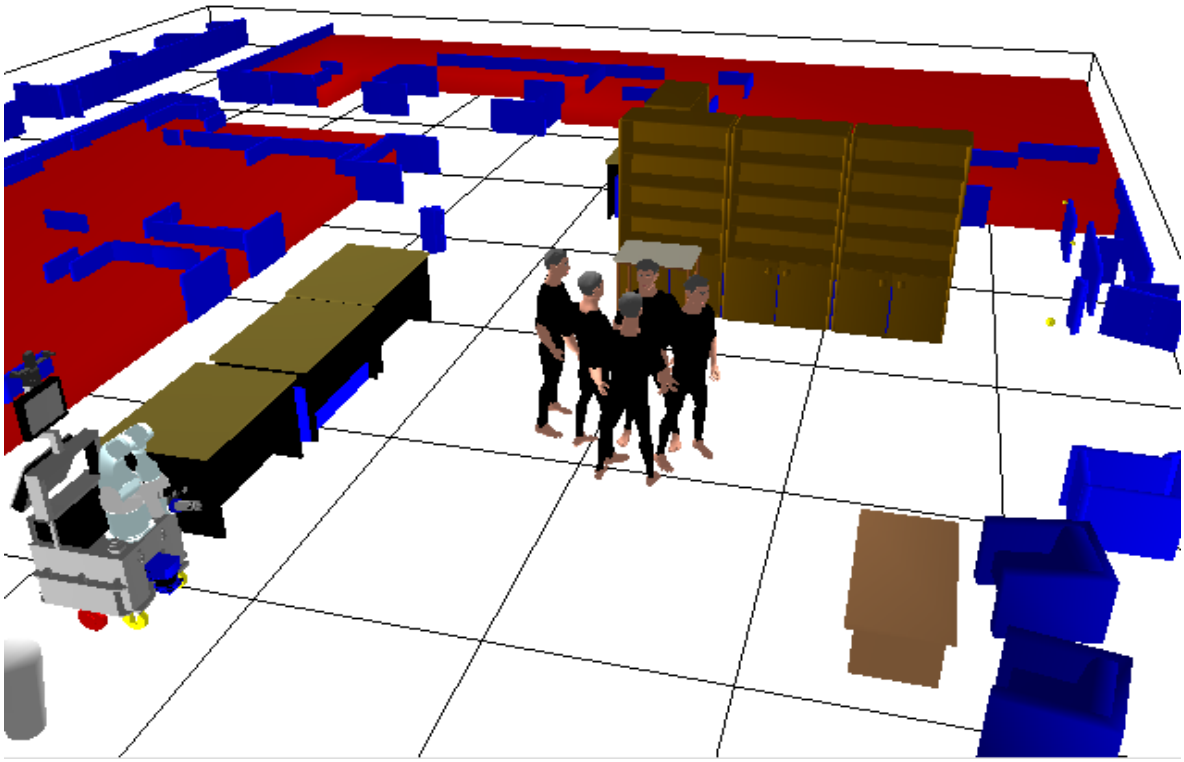


Figure 3.42: b) Scenario 2: Man in a crowd: Two opposite testbeds for our search methods. The second scenario, it is presented a very hard case, where the robot has to "GoTo-Talk" to the human surrounded by four person.

Best Configuration Search (BCS)

The best configuration search obtains the higher utility from those created points on the approach or the interaction areas. The table 3.1 shows the CPU time taken for different grid resolutions. As we can observe there is almost no gain on utility from a grid resolution of 11x8 (88 points) for the scenario 1 with almost no visual obstacles.

Table 3.1: CPU time of the BCS method from Scenario 1

matrix	Points Generated	Tested	Valid	Cpu Time (s)	UTILITY
11x3	33	19	19	1.41	1.45
13x3	39	25	24	1.79	1.45
15x3	45	27	27	2.01	1.45
17x3	51	31	30	2.2	1.45
11x5	55	34	34	2.5	1.46
11x8	88	63	61	4.42	1.47
11x12	132	94	91	1.13	1.47
18x8	152	105	103	7.46	1.47
11x16	176	125	123	8.9	1.47
33x8	264	187	184	13.38	1.47
40X11	451	321	317	22.76	1.47
40X18	738	537	525	38.06	1.47
49x22	1078	789	779	61.18	1.47
49x50	2450	1820	1775	139.53	1.47
91x50	4550	3378	3305	239.39	1.47

Table 3.2 presents the cpu time and utility given from the scenario 2. Here we can observe that also the grid of 88 points give enough information in a still acceptable computation time.

This method is slow in terms of interaction but assures to find the best position for interacting with the person or approaching to an object. Figure 3.43 shows examples of all the configurations generated inside the interaction area on different scenarios. These configurations are tested and ordered following their costs as well as their quality of perception. As we can observe, configurations that are in collision with the second human on the environment are not tested.

Cost Based Search (CBS)

The inner works of this method is very fast; it is very useful for reducing the planning time for interaction. The algorithm gives an user acceptable configuration for interaction mostly when there are few collision obstacles or visual obstructions. Tables 3.3, 3.4 and 3.5 show that this method gives a low utility in comparison with the BCS but its very fast, for the two scenarios.

Tables 3.4 and 3.5 are from the same scenario (scenario 2) but the second with a higher threshold for the quality (30% and 50% respectively).

The time of this method is very fast, it is far from searching the best point but it guaranties an interaction time. Normally when the quality of the perception is the same on all positions (or almost the same) as in the figure 3.35 the BCS and CBS will find the same position but CBS on less time.

Table 3.2: CPU time of the BCS method from Scenario 2

matrix	Points Generated	Tested	Valid	Cpu Time (s)	UTILITY
11x3	33	21	10	1.44	1.238
13x3	39	24	12	1.72	1.202
15x3	45	29	15	2.08	1.189
17x3	51	32	15	2.11	1.244
11x5	55	32	21	2.82	1.238
11x8	88	51	40	5.16	1.370
11x12	132	74	58	7.41	1.345
18x8	152	91	73	9.4	1.352
11x16	176	100	82	10.53	1.375
33x8	264	161	128	16.61	1.374
40X11	451	275	220	28.28	1.385
40X18	738	446	374	47.97	1.382
49x22	1078	651	551	70.4	1.379
49x50	2450	1470	1271	161.71	1.380
91x50	4550	2735	2357	306.79	1.386

Table 3.3: CPU time of the CBS method from Scenario 1

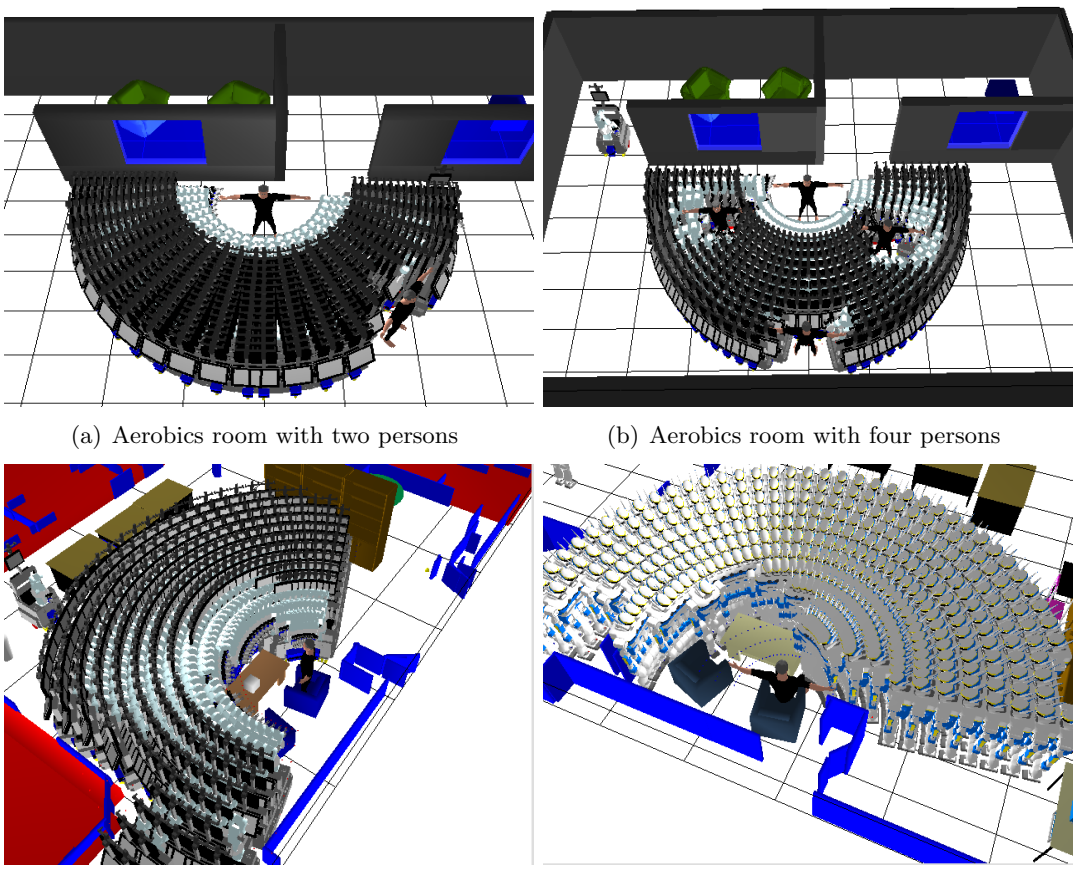
Points Generated	tested	CPU time (s)	UTILITY
33	1	0.07	1.43
55	1	0.09	1.43
88	1	0.08	1.43
2450	1	0.5	1.43

Table 3.4: CPU time of the CBS method from Scenario 2 with 30 percent in the quality of perception of the target

Points Generated	tested	CPU time (s)	UTILITY
33	1	0.09	1.18
55	2	0.15	1.19
88	3	0.21	1.18
2450	5	0.74	1.19

Table 3.5: CPU time of the CBS method from Scenario 2 with 50 percent in the quality of perception of the target for 2450 points

	CBS Sce.2		
Points Generated	tested	CPU time (s)	UTILITY
2450	597	31.9	1.23



(a) Aerobics room with two persons

(b) Aerobics room with four persons

(c) Grande salle, constrainted environment around a human on the couch

(d) Grande salle, with a different robot structure

Figure 3.43: Robots compute all its possible (and oriented) configurations around the human, avoiding to collide with humans or other obstacles on the environment.

Random & Gradient Search (RGS-RGS2)

This search method its executed on less time than the BCS, slower than CBS but, as the utility function grows gradually, RGS has more probabilities of finding the best position point or at least a local best. On the table 3.6 we show the results for 2450 (49x50)[12] points with 5,10,20 values of δ[13].

Table 3.6: CPU time of the RGS method from Scenario 1

Points Generated	delta	Tested	Cpu Time	UTILITY
2450	5	18	1.11	1.44
		33	1.79	1.45
		29	1.62	1.46
		51	2.64	1.46
		58	3.34	1.46
		25	1.34	1.46
		34	1.93	1.44
		37	1.93	1.46
		25	1.42	1.44
		22	1.16	1.46
	10	47	2.6	1.46
		25	1.37	1.47
		42	2.34	1.45
		14	0.79	1.47
		67	3.67	1.47
		22	1.21	1.46
		37	2.01	1.46
		37	1.97	1.46
		38	2.03	1.47
		28	1.56	1.46
	20	40	2.09	1.46
		74	4.07	1.46
		89	4.74	1.46
		85	4.6	1.47
		125	6.66	1.46
		36	1.96	1.46
		48	2.59	1.47
		66	3.5	1.47
		63	3.27	1.46
		41	2.16	1.47

While BCS takes around 130sec. on finding the best configuration, and the best utility or a close approximation obtained by RGS is in an average of 3.56sec. For the scenario 2 the RGS method arrives little less than half of the times to the best utility, as it is shown on table 3.7, with

[12] We have chosen this resolution in order to show clearly the difference in the time spent between methods

[13] the iteration counter of non utility improvement for RGS and the counter of random point generated for RGS2

values of δ 10,20 and 50.

Table 3.7: CPU time of the RGS method from Scenario 2

Points Generated	delta	Tested	Valid	Cpu Time (s)	UTILITY
2450	10	30	27	3.41	1.25
		25	23	2.84	1.27
		19	18	2.5	1.17
		32	28	3.66	1.38
		26	24	3.21	1.38
		46	43	5.13	1.38
		44	41	4.99	1.27
		31	28	3.66	1.38
		29	25	3	1.38
		41	38	5.41	1.38
	20	26	23	2.36	1.38
		32	32	4.04	1.21
		39	37	4.04	1.27
		26	24	2.57	1.27
		43	40	4.48	1.38
		32	30	3.81	1.21
		40	31	3.05	1.38
		21	18	1.91	1.28
		45	43	5.02	1.2
		41	39	4.45	1.27
	50	78	73	7.83	1.27
		34	31	3.54	1.38
		81	75	8.13	1.27
		74	61	6.59	1.36
		18	18	2.26	1.21
		26	22	2.38	1.28
		26	22	2.54	1.21
		77	68	7.35	1.38
		49	43	4.95	1.27
		74	61	6.74	1.28

On the other hand RGS2, takes little more time but its rate of success increases as illustrated in tables 3.8 of from the scenario 1 and 3.9 from the scenario 2.

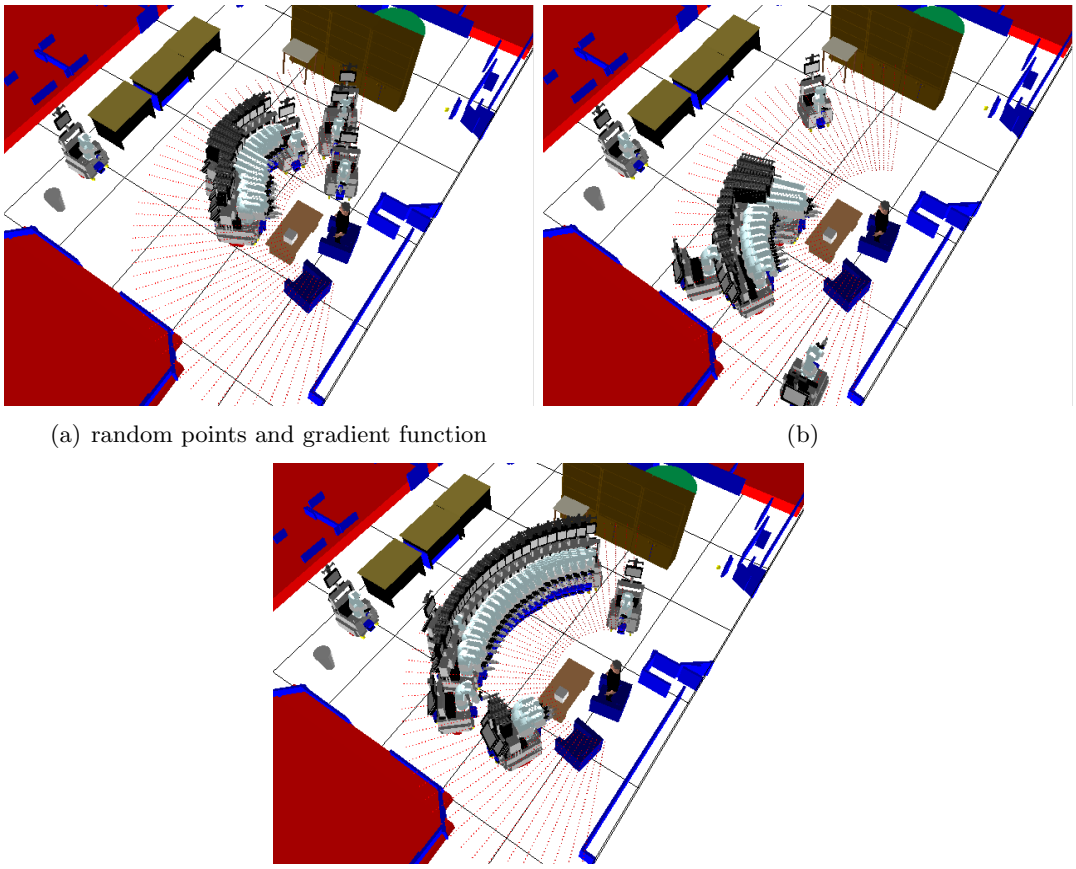
We can observe in figure 3.44, the some examples of the configurations generated on this method.

Table 3.8: CPU time of the RGS2 method from Scenario 1

Points Generated	delta	Tested	Valid	Cpu Time (s)	UTILITY
2450	5	175	175	9.86	1.46
		134	133	7.57	1.45
		141	138	7.27	1.47
		226	226	12.6	1.47
		203	195	10.17	1.47
		206	206	11.08	1.46
		257	249	13.6	1.47
		123	123	6.72	1.46
		158	158	8.33	1.47
		149	148	7.86	1.47

Table 3.9: CPU time of the RGS2 method from Scenario 2

Points Generated	delta	Tested	Valid	Cpu Time (s)	UTILITY
2450	15	244	236	26.2	1.27
		258	252	27.99	1.27
		287	271	28.3	1.38
		260	249	28.86	1.38
		267	255	32.44	1.36
		193	176	24.18	1.27
		234	219	33.53	1.36
		224	213	31.72	1.38
		258	249	37.64	1.28
		260	249	39.92	1.27
	30	437	416	52.47	1.27
		477	454	57.37	1.38
		448	414	53.87	1.38
		430	413	64.48	1.28
		486	463	70.77	1.38
		469	443	66.37	1.38
		456	440	67.59	1.38
		420	400	60.91	1.38
		414	392	59.75	1.38
		421	399	59.83	1.38



(a) random points and gradient function

(b)

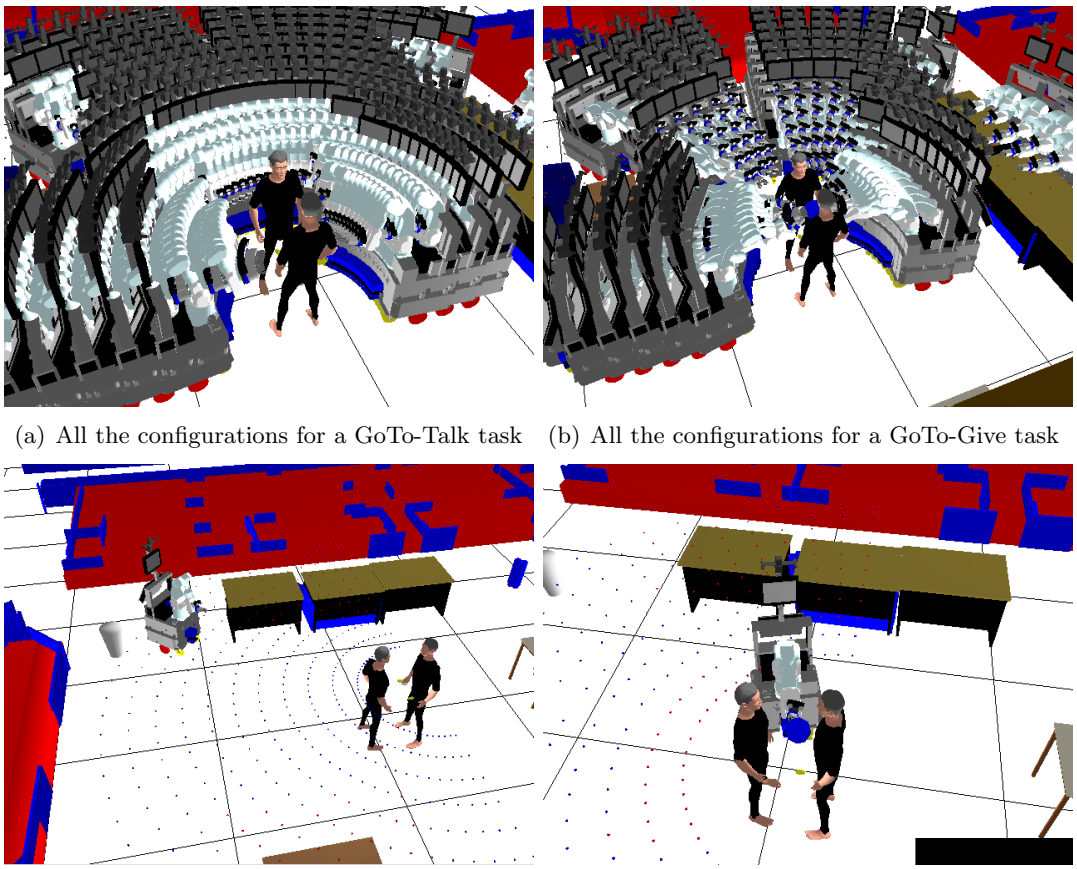
(c) The robot finds the local maxima that in this case is the global maxima

Figure 3.44: The robot search in its neighborhood to find to improve as possible the randomly obtained configuration.

3.5.4 Task Goal

The task goal as an additional utility modifies the optimal position towards the task goal accomplishment. Figure 3.45 shows the first scenario, the difference between the optimal configuration found for a Goto-Talk/Look task and the one obtained for a Goto-Give task, when two persons are talking. The reached positions are fewer on the second task, due to the configuration that a robot has to perform for a "give" task. The final position for a "Talk/Look" task has almost not changed from its original position, while in the second task the robot has to stand in a close proximity to the human.

On the second scenario shown on the figure 3.46, we can also observe the perspective of the two different tasks. The perception of the human is not highly occluded, but an obstacle prevents the robot to approach the human from the front. Even when some configurations seems possible to be achieved from the other side of the little table (a closer position from the initial configuration), the robot finds that the most feasible placement for a GoTo-Give task is obtained by avoiding the table, and getting closer of the human.



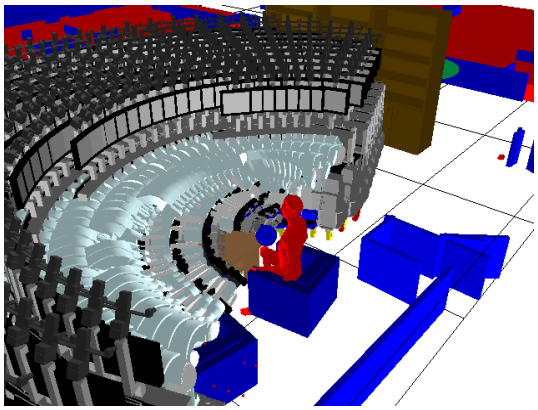
(a) All the configurations for a GoTo-Talk task

(b) All the configurations for a GoTo-Give task

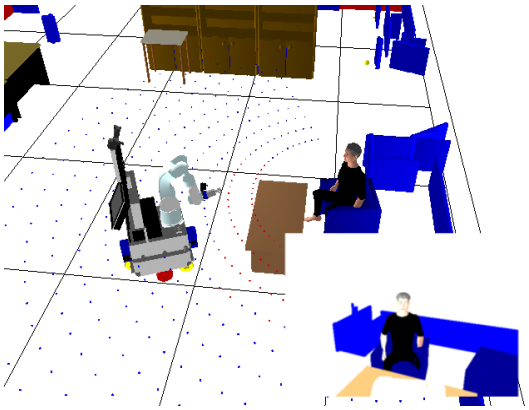
(c) Final configuration found for a GoTo-Talk task

(d) Final configuration found for a GoTo-Give task

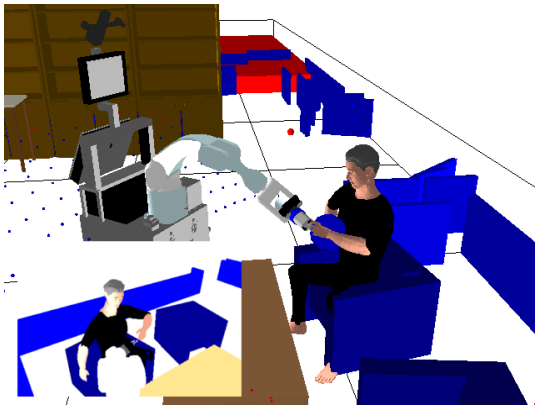
Figure 3.45: Scenario of two person talking on the Grande Salle environment: The number of possible configurations for GoTo-Talk task is higher than for a GoTo-Give. The final position for the Give task is closer because the utility increases while the robot gets closer to the exchange point position, with the grasp extremity.



(a) All the configurations for a GoTo-Give task



(b) Final configuration found for a GoTo-Talk task



(c) Final configuration found for a GoTo-Give task

Figure 3.46: Scenario of a person waiting for his/her coffee: the robot finds the "optimal" position of the give task in a close position of the human even when there are configurations that can achieve the task from the other side of the table.

3.6 Discussion

In this chapter I have introduced the PerSpective Placement as a manner of implementing the "Perspective Taking" and the "mental rotation" in software architecture for helping to a motion planner, in a Human-Robot interaction framework.

The planner was presented as the complementary (and necessary) partner of the Human Aware Motion Planner. Both planners work together in a perfect symbiosis where, not only they exchange data as most of the systems do, but also they share a whole description of the environment.

The PSP, as I call it here, is intended to be a general framework to compute "perspective scoped" configurations. It works as a generator of destination points for navigation planners; or of configurations, where the robot can hand and/or manipulate objects for manipulation planners. All this, focused on human-robot interaction. I have illustrated a number of simulation results to demonstrate how both planners act in different scenarios.

One of the best contributions of the "PerSpective Placement" on motion planning is the adaptation of the social criterions (i.e. security and comfort) to the search of the placement configuration. All this accomplishes the task of perceiving the target while it is minimizing the trajectory distance.

Nevertheless, this architecture can be improved in several ways, for example minimizing the point testing by different search methods, a learning process for adapting the weights for the costs or quality, and also it can be extended by searching 3D sensor positions, as we describe below.

3.6.1 Weight Adapting Learning Process

Seeing as no apparent justification or general reasoning is given to the choice over the values considered in the weights affecting the utility function in this chapter, this is, the quality and cost variables; we consider there is a lot to be done in this direction. An interesting approach for these weights is, rather than to be chosen by the developer or adjusted to a fixed configuration, to be dynamic and learned over several task executions, so that it is possible to obtain a higher adaptability to the environment and to have a further refined human-robot interaction. The utility function maximization through learning algorithms has largely been the object of study [Wang 08][Perez 08] and therefore various options are available as solutions to pertinent weight choices through machine learning, for example reinforcement learning.

In our case we can observe Δ_y, in the quality and cost variables and even more specific to the cost variable the λ_x weights can be calibrated as to give higher performance. Reducing the possible values of the weights without affecting the performance of the robot we can create a search space in which the complexity allows us to converge easily to a solution. Sutton & Barto [Sutton 98]propose an approach using TD (temporal difference) learning in which based on a given initialization of the weights (as done currently) and measuring according to the utility function we can progressively maximize through several executions the value of the mentioned utility function, incrementally and even dynamically, this is, if the user had a certain weight configuration in which the utility was maximized but later on his/her optimal weight configuration was to change, the robot would learn the new configuration as well.

3.6.2 Further task-goal postures

The approach presented in this chapter, deals basically with 2D positions of the sensor. Even when we test 3D collision and when we modify the robot configuration to orient the sensor to the

target, the points that are generated are positions on the floor, where the robot has to reach this point by placing its center position (x, y) in these two dimensional coordinates.

With the addition of final postures, we could deal to the modification of the position of all (or some of) the body parts of the robot, this without moving the robot position on the environment. The modification is performed to reach a configuration where the robot can perceive a target, without displacement.

In simple words, that would allow the robot to minimize the displacement by just changing the place of the body that holds the perception sensor. For example, moving the head of a Humanoid robot, by leaning the rest of the body towards a closer position to the target.

This, can give place to another task goal "GoTo-Check", means for example to go to look inside another object (i.e. a box), or to look in all the surface of the object.

Due to the actual morphology of our robot Jido, where its arm holds a pair stereo camera, this extension would be more advantageous. The robot can move this long arm in order to look for a posture where it can have a better perception of the target. It can be used also for testing if the position is apt for modeling objects as shown in some NBV approaches (See Sect. 2.2.3).

For resolving this problem, our approach could be having points in a 3D shape, like a sphere or cylinder and as we can show in our early results illustrated on the figure

4 Robot Implementation and Results

4.1 Introduction

The integration of algorithms designed for human-robot interaction raises some new problems and some unresolved problems on robotics. For example, the HRI approaches are very dependent on detection and localization, not only of the robot in the environment but also of the human. People detection becomes mandatory to ensure human security and also for the very meaning of the interaction.

Another aspect that is necessary to take into account is that, for interacting with people, the robot has to have a robust multi-component system in software as well as in hardware design levels. This system has to provide flexibility, extensibility and efficiency.

This chapter presents the integration of the perspective placement planner (PSP) into the multi-modular architecture inside our mobile robotic platform, as well as a brief description of human detection modules that play an important role on the good performance of our system.

4.2 General Architecture

The Perspective Planner for motion planning is integrated to our robotic platform Jido. The software architecture of this robot is based on the *LAAS Architecture* [Alami 98] having multiple *Genom*(called now OpenGenom) [Fleury 97] modules. This architecture, provides a great level of modularity and generality that eases the programming load of integration. It was originally conceived as a three-layered architecture (functional, decisional and execution), although, for better adaptation to HRI, the LAAS architecture was revised and transformed into a two-layered architecture:

- **Functional Layer:** This Layer, also called the "low layer", contains the whole perception and action functions of the robot. Control loops and data interpretation are encapsulated into Genom modules. The modules in this layer have direct access to robot's hardware components (i.e. sensors and motors) and offer services controllable via requests. Each module communicates by publishing on "posters" (its own attributed memory block).
- **Decisional Layer:** The indicated layer, named "high layer", contains components that provide decision capabilities to the robot. A task planner, a supervisor and a fact database are situated in this layer.

In the figure 4.1 we can appreciate the completeness and modularity of the the whole system.

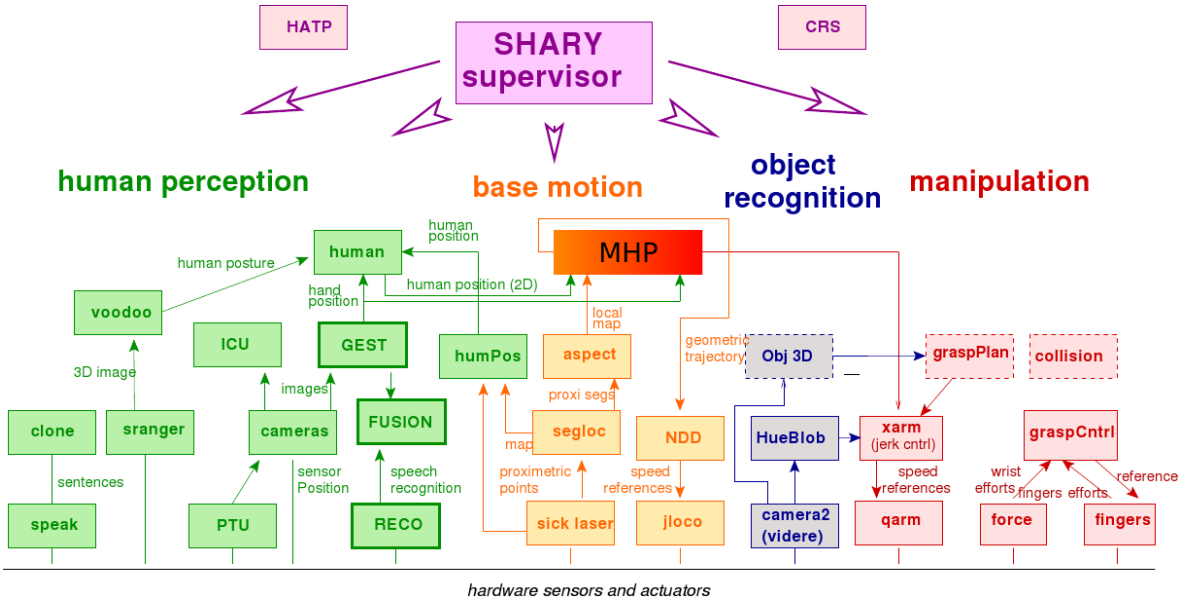


Figure 4.1: The whole architecture. Modules in the superior part (SHARY, HATP, CRS) belong to the decisional layer. In the lower part are shown all the modules of the functional layer (categorized on task types).

As we have introduce before, the system has been carried into our robot *Jido* equipped with three Pentium IV processors, one Laptop with an Intel core-duo processor, two 2D SICK laser scanner (front and rear), a six degree of freedom Mitsubishi PA-10 manipulator, 1 pan-tilt stereo camera, 1 color stereo cameras on the end axis of the manipulator, 8 sonar sensors, and three finger tactile hand. Jido is illustrated on the figure 4.2

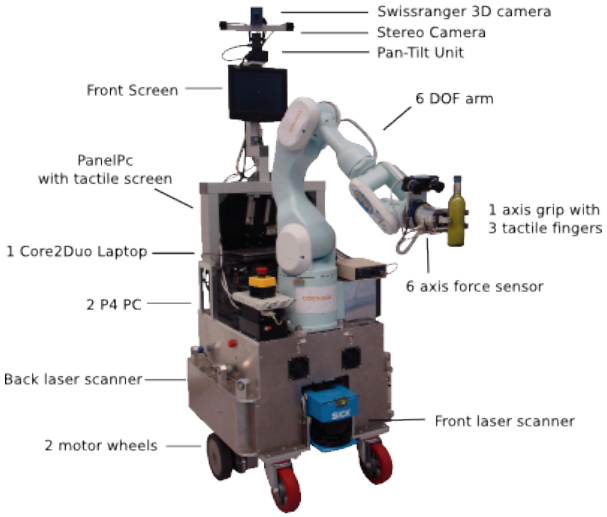


Figure 4.2: The robotic platform Jido where the system is integrated

Jido is conceived to be a "home-helper", due to its various sensors and capabilities of interaction. For now, it is the most complete, autonomous robotic platform on LAAS-CNRS.

4.3 Motion in Human Presence (MHP) Module

The PSP and the HAMP planners are integrated into the LAAS architecture as a single Genom module. As both of the systems rely on Move3D and share common/similar world representations and their interaction is very close (but independent), they are implemented and represented in the architecture as one module called MHP, Motion in Human Presence. The fact that these two systems are merged in one module provides, not only computation and programming advantages, but also, it allows to share the same environment representation including robot and human models, thus avoids a possible risk of their mismatch.

MHP module is situated on the functional layer. Nevertheless, the PSP system can be also considered as a link between these two layers. Supervisor can control the MHP planning at any time by a number of requests

Figure 4.3 illustrates the MHP module and its components. The requests generated and sent by supervisor are too abstracts for the motion planner to execute. PSP behaves as an intermediate level between the supervisor and the HAMP planner. It transforms the high level requests of the supervisor to more concretes commands for the planner.

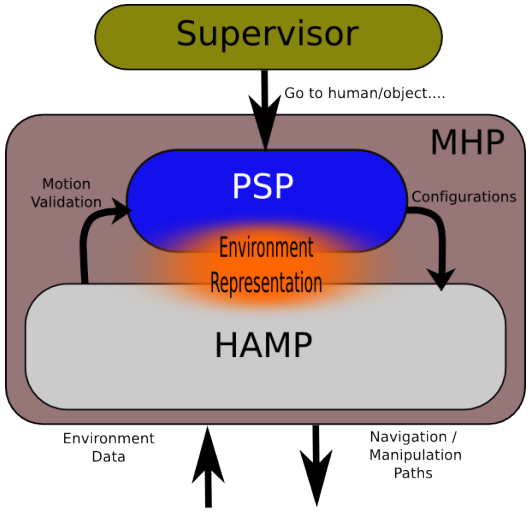


Figure 4.3: The internal architecture of the MHP module

As an example, the supervisor sends the request GO TO THE HUMAN X, yet this request stays too abstract for the HAMP, because it needs to have a robot goal position (x,y,θ) to plan a path. That is why PSP reasons on the task asked by the supervisor and computes a goal position for the motion planner.

After receiving the interpreted orders from PSP, the motion planner rely on its own algorithms and internal structures to generates robot's paths.

4.4 Human Detection Modules

Detecting humans is necessary for a robotic system that involves interaction with humans. There are different methods depending on robot's sensor capabilities.

With camera and laser, the information can be used to detect more precisely humans in robot's proximity [Kleinehagenbrock 02]. In absence of cameras, the laser can be used, in some situations, to detect leg-like shapes [Xavier 05]. After the detection, tracking[Shulz 01, Baba 06] must be launched in order to follow the human motions and detect motion patterns.

Many human detection modules can be found on the LAAS architecture (most of them based on machine vision approaches). These modules are mostly described on [Fontmarty 07], the most important for the MHP module are:

- **HumPos** Human Detection & Tracking Module, using mainly laser data and matching information with the ICU/ISY module.

- **GEST** Face and hands detection, based on stereo vision information.

- **ICU** Face and body recognition and tracking, using mono-camera to recognize faces and match them with information from a database.

4.4.1 HumPos - Human Detection & Tracking Module

I have encountered the problem of robust human detection in every step of its motion and in a farther distance than camera detection. For this reason, It was necessary to develop this module passing a more reliable information about human position and orientation to the MHP module (PSP - HAMP).

This module provides human detection and tracking services based on laser and camera data. HumPos provides a list of humans in the environment to the MHP module. This list contains positions and orientations of the detected humans associated with a confidence index and an identifier.

The general algorithm consists of two phases: detection and tracking. On the first phase lie two types of detection (laser and visual), checking for correspondences. Finally, the algorithm assigns orientations either towards the robot, if person's face is detected, or in his motion direction acquired by the tracking phase.

Algorithm 6 General algorithm on HumPos Module

```
1: if a person P detected by laser then
2:   if visualdetection detects P then
3:     Direction_P ← looking at the robot (body towards the robot)
4:   else
5:     if P is moving then
6:       Direction_P ← motion direction
7:     else
8:       if P already in list then
9:         Direction_P ← Old_Direction_P
10:      else
11:        Direction_P ← looking at the robot (body toward the robot)
12:      end if
13:    end if
14:  end if
15: end if
```

Figure 4.4 shows graphically the overall mechanism of human detection and tracking.

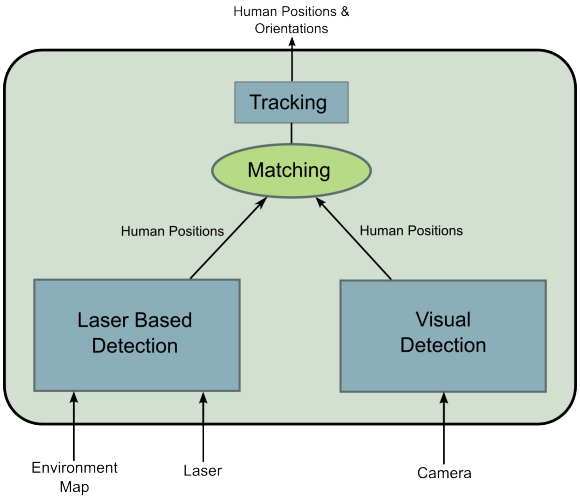
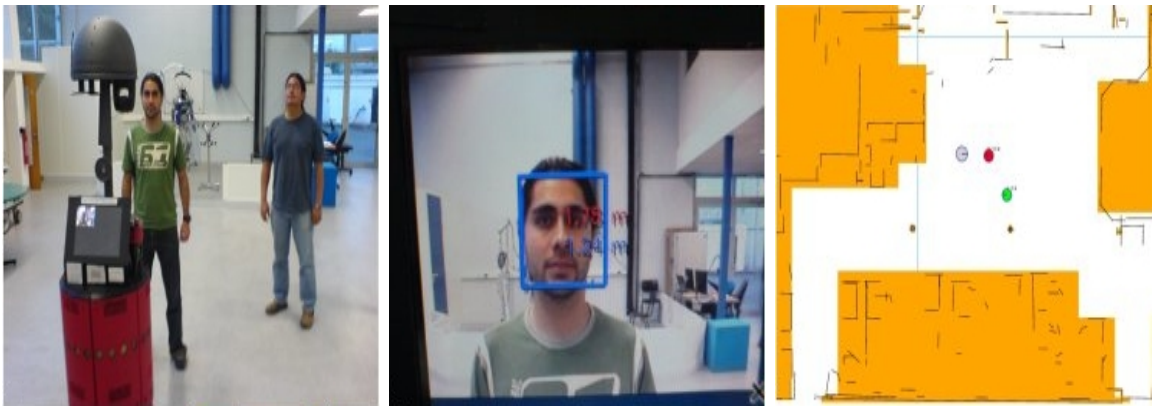


Figure 4.4: The Human Detection process combines laser and visual data to detect and track humans.

In laser-based detection, supported on a map of the environment, the static obstacles are filtered from the sensor obtained data. Resulting points are then used to detect leg-like shapes (a leg or pair of legs) according to their geometry and neighborhood. This process produces a list of detected humans with their positions and an attached confidence index.

On the other hand, the visual data coming from the camera are used to detect people in the near proximity of the robot (∼[1.5 - 2.0]mts) (figure 4.5-b) by the visual face detection module, explained in more detail in [Brèthes 05]. The visual detection process provides a list of humans with their estimate distance to the robot. This, using camera's position and orientation linked to face size metrics.



(a) Laser leg detection (b) Face detection (c) HumPos Data Fusion

Figure 4.5: Scenario where two persons have been detected based on laser data. One of these persons is also detected using a vision-based face detection. Once the data fusion is performed, the person detected by the camera has high confidence value (marked in red) while the other person is marked with a lower probability.

These two lists are then matched in the final stage to produce only one list of humans (with their confidence index) with corresponding positions (figure 4.5-c).

Finally detected humans are tracked by the tracking stage and here, based on it's motion, an orientation is assigned to detected elements. This supposing that people naturally face to their walking direction (if and only if a face is not detected, that implies that the person is looking at the robot).

The tracking stage, use an algorithm based on Gaussian sum filters. Here one human is represented by several particles (possible positions), and each of these particles has a probability of being a good estimation. After the particle generation, a Kalman filter is applied to each particle in order to predict and correct the states of the particles.

At each sensor acquisition the tracking algorithm has to perform the following steps:

- **Prediction:** Predict possible states of humans (generation of particles).
- **Data association:** associate predictions (particles) to measurements.
- **Correction:** correct the predicted states of humans using the measurements.
- **Pruning:** select particles that will be kept.

There are two types of particles: S and G. A particle in state S represents a human that is not moving (Static), a particle in state G represents a human that is moving. These two types will be the bases for the human motion on the prediction step.

Prediction step:

It is necessary to have a model in order to predict the movement of a person in its environment. As the walking movement of a human is generally "erratic" in "constrained spaces", it is needed to make several predictions corresponding to the main possible trajectories. As we are interested in tracking people in a limited space, it is important to consider that a person will often change

direction or even stop at any moment. For this purpose i have defined four main behaviors in human walking that will help on the prediction of his next position in the environment: *start* for a person who was static and suddenly started to walk; *go* for a person that moves in one direction; *change direction* of the original walking direction and *stop*. These behaviors can succeed two previous status *walking* and *stopped*.

We consider that a person who is walking (G) can be modeled as:

- **GF:** Go forward with the same velocity and same direction.
- **GL:** keep its velocity but change its direction of $\pi/3$ to the left.
- **GR:** keep its velocity but change its direction of $\pi/3$ to the right.
- **GS:** Simply stop.

The models for a person who is not moving (S) can be represented as:

- **SF:** Start walking in the direction th=0.
- **SR:** Start walking in the direction th=$-\pi/2$.
- **SB:** Start walking in the direction th=π.
- **SL:** Start walking in the direction th=$\pi/2$.
- **SS:** Stay where it is.

At the end of the prediction step, each human is represented by n particles that are predictions of possible states of the human. Each particle has a state (predicted by one of the models), a state covariance matrix and a probability ρ to be a good estimation. On the first sensor acquisition all the detected humans are in a state S.

The table 4.1, illustrates all the probabilities for all different particle types and states.

Table 4.1: Table of prediction probabilities depending on previous human state

Type / State	Stop / Stay (S)	Forward (F)	Turn Right (R)	Turn Left (L)	Backwards (B)
S	0.9	0.03	0.025	0.025	0.02
G	0.02	0.8	0.09	0.09	0.00

Data association step:

For performing the association for each particle of the human, first it is necessary to find the likelihood λ between each human h previously detected and the new sensor measurement M.

$$\lambda_{h|M} = \sum_{i=0}^{n} (\lambda_{p_i|M} * \rho_{p_i})$$

where the p_i is the the prediction particle i of the human h.

Once the likelihood of each human on the tracking list is obtained the next step is to select the associations $h|M$ for which there is at least one particle in the gate of the measurement. This is done by calculating the Mahalanobis distance, between the measurement and the prediction. Only the associations with a Mahalanobis distance inferior to s threshold of the value 2 will be kept. This means that only associations that are probable will be taken.

Then the associations human-measurement are achieved; first, by associating the couples $h|M$ with the highest $\lambda_{h|M}$. Then, the measurements that have been associated with no human are considered as new humans, they are added in the tracking list. When one of the "humans" on the list has not been detected, its detection status change between the following detection states:

- **OCCLUDED:** Not detected caused by another object or human.
- **LOST:** Not detected for a moment.
- **DISMISSED:** Definitively lost and marked for erasement.

Initially, a human that has not been detected is considered as LOST, and it can change to OCCLUDED if it accomplishes three criteria:

1. The human must have more than 10% chances of being occluded.
2. It should be constantly detected more than 75% of the total time from its first appearance.
3. It has been missed less than 15 sensor acquisitions in a row ($\approx$ 3.3 seconds)

Otherwise, if the non-detection time exceeds the threshold, it is considered as DISMISSED and tagged for being erased from the track list. The third criterion is because the human motion cannot be properly predicted if it has been occluded for a long time.

Correction step:
The correction step of the Kalman filter is applied to every particle of a human that has been associated to a measurement.

$$X_{corr} = X_{pred} + K(Y_{mes} - Y_{pred})$$

where X_{pred} is the predicted state, X_{corr} is the corrected state, Y_{pred} is the predicted measure and Y_{mes} is the real measure. K is the Kalman gain. For each particle, ρ is updated by $\rho = \rho * \lambda$.

Pruning step:
In this step a selection of n particles that have the highest ρ is done the rest of non interesting particles should be eliminated. Between this selection the particle that will represent the human on the tracking list will be the particle with the highest probability ρ.

On figure 4.6 we can appreciate a human that passes in front of the laser sensor. First, a detection of the legs allows getting the position of the person and it is added to the human tracking list. After this, the HumPos tracking phase follows and matches the human detected with the one on the list, this step gives an orientation to this person in the natural walking direction.

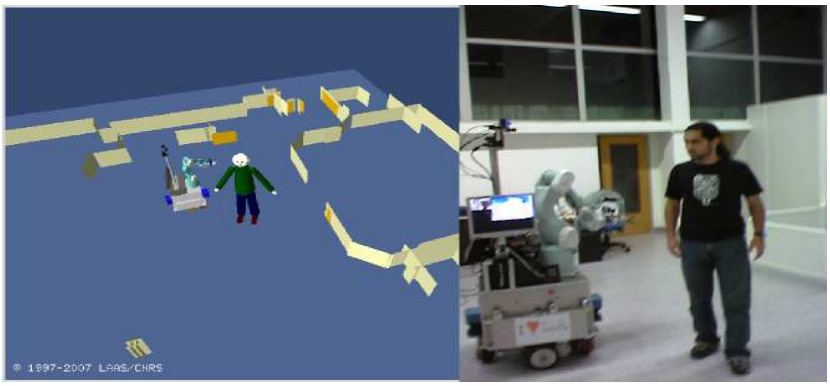


Figure 4.6: The Human Tracking process gives orientation to the human on the persons moving direction

4.4.2 The Gest module

The GEST module was developped by Brice Burger and it can be found in [Fontmarty 07]. The goal of this module is to provide a 3D human hand tracking from the video stream of a stereoscopic system. Actually, the hand is modeled by a 3D deformable ellipsoid. The template is fitted through the estimation of its space coordinates (x, y, z), its size (ax, ay, az), and its orientation (θ, ϕ, ψ). All these parameters are included in the state vector $x_k = (x_k, y_k, z_k, ax_k, ay_k, az_k, \theta_k, \phi_k, \psi_k)$ related to the $k-th$ frame.

As the robot's evolution takes place into dynamic and cluttered environments, several hypotheses must be handled at each instant concerning the tracker parameters to be estimated. Therefore the tracker is based on the I-Condensation algorithm [Isard 98].

With regard to the dynamics model $p(x_k|x_{k-1})$, the hand motion is difficult to characterize over time. It is supposed that the state vector entries evolve according to mutually independent random walk models, *viz.* $p(x_k|x_{k-1}) = \mathcal{N}(x_k|x_{k-1}, \Delta)$, where $\mathcal{N}(.|\eta, \Delta)$ is a Gaussian distribution with mean η and covariance

$$\Delta = diag(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_{ax}^2, \sigma_{ay}^2, \sigma_{az}^2, \sigma_\theta^2, \sigma_\phi^2, \sigma_\psi^2).$$

In order to evaluate the 3D particles after their generation, the projection on the stereo images has to be done. The corresponding ellipses are obtained by a common quadric projection [Menezes 05].

As follow a characterization of both importance and measurement functions involved in the tracker.

Measurement function The measurement function is based on skin color probability images and is inspired from [Thayananthan 03].

Each ellipse e - which is a projection of one particle - is given a likelihood $p(z, e)$ that depends on the average of skin color probabilities around the template corresponding to e. The pixels in the image are partitioned into a set of object pixels O, and a set of background pixels B. Assuming pixel-wise independence, the likelihood can be factored as

$$p(z, e) = \prod_{o \in O} (p(Ps(o), e)) \times \prod_{b \in B} (1 - p(Ps(b), e))$$

where $Ps(k)$ is the skin color probability at pixel location k.

The likelihood of the particle x is given by the merge of the two corresponding projected ellipses likelihood.

Importance function The importance function $\pi(.)$ is defined by a Gaussian mixture from the triangulated 3D skin blobs. Let N be the number of detected 3D skin blobs and

$$b_i = (x_i, y_i, z_i, ax_i, ay_i, az_i, \theta_i, \phi_i, \psi_i), i \in \{1..N\}$$

the ellipsoid descriptions corresponding to each such region. An importance function $\pi(.)$ at location $\mathbf{x} = (x, y, z, ax, ay, az, \theta, \phi, \psi)$ follows, as the Gaussian mixture proposal

$$\pi(\mathbf{x}, z) = \sum_{i=1}^{N} \frac{1}{N} \mathcal{N}(\mathbf{x}|b_i, \Delta).$$

Snapshots of a typical sequence is shown on figure 4.7.

Figure 4.7: Hand being tracked by the Gest module : the ellipses are the projection of the 3D state vector

4.4.3 The ICU module

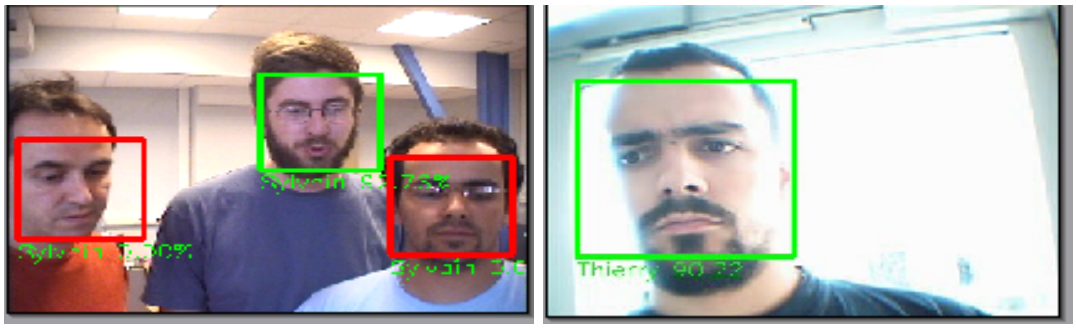
The ICU module was developed and improved by Mathias Fontmary and Thierry Germa and can also be found in [Fontmarty 07]. This module aims to track human upper-body (torso and head) in a video stream in order for the robot to be sure that it can interact with the person in front of it and to be able to follow somebody. The ICU module is composed of three main parts :

User face detector This detector is based on the well-known window scanning technique introduced in [Viola 01]. This classifier covers a range of $\pm 45°$ out-of-plane rotation of the user's face. It is used to switch between different modalities, but also to feed the face recognition part of the module.

User face recognition This part is based on the eigenface as described in [Turk 91]. It aims to classify facial regions $\mathcal{F}$ segmented from the face detector into either one class C_t out of the set $\{C_t\}_{1\leq t\leq M}$ of M tutors of the robot. Each new detected face $\mathcal{F}(j)$, written as a $nm \times 1$ vector, is reconstructed in $\mathcal{F}_{r,t}$ by projecting it into $B(C_t)$. $\mathcal{F}$ is linked to the class C_t by its error norm.

User Tracking The tracking part is based on the I-Condensation algorithm also used in Gest. The followed template is fit with its location $[u_k, v_k]^{'}$, and its scale s_k, so that $\mathbf{x}_k = [u_k, v_k, s_k]$. In the human upper-body tracker, multi-patches of distinct color distribution related to the head and the torso are considered.

Moreover, taking into account the recognition step, the importance function related to the tracked class C_l becomes, with $\mathbf{b}_j$ the centroid of the j^{th} extracted face



(a) (b)

Figure 4.8: Snapshots of detected/recognized faces with associated probabilities. The target is Sylvain (resp. Thierry) for the first (resp. last) frame.

Figure 4.8 shows some snapshots of recognized tutors' faces where the detector marked in red color. The detected faces but only those in green color are recognized from the previously learned faces.

Figure 4.9 involves occlusion of the target by another person crossing the field of view. The combination of multiple cues based likelihood and face recognition allows to keep track of the region of interest even after a complete occlusion.

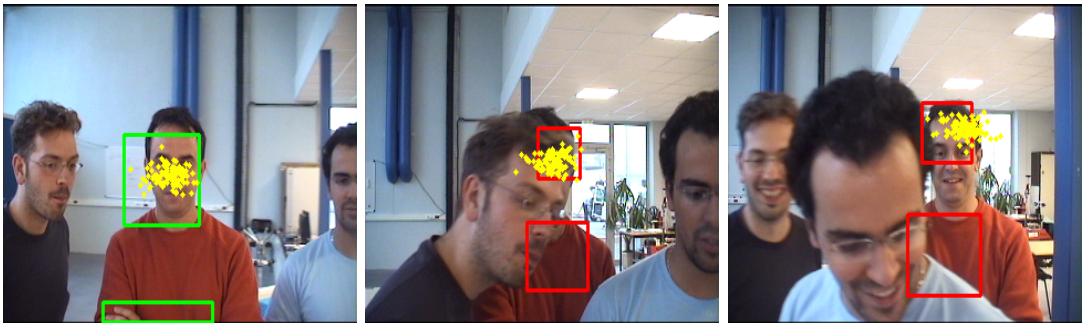


Figure 4.9: Tracking scenario involving full occlusions between persons. Target recovery.

4.5 Real World Results

In figure 7.4, a scenario is shown with a person interacting with the robot. The person indicates the robot to go and pick up an object on the table and bring it back to him ("bring me the yellow bottle"). The robot makes the respective plan with the sequence of different tasks. PSP module finds a valid configuration to execute each task that implies navigation motion of the robot.

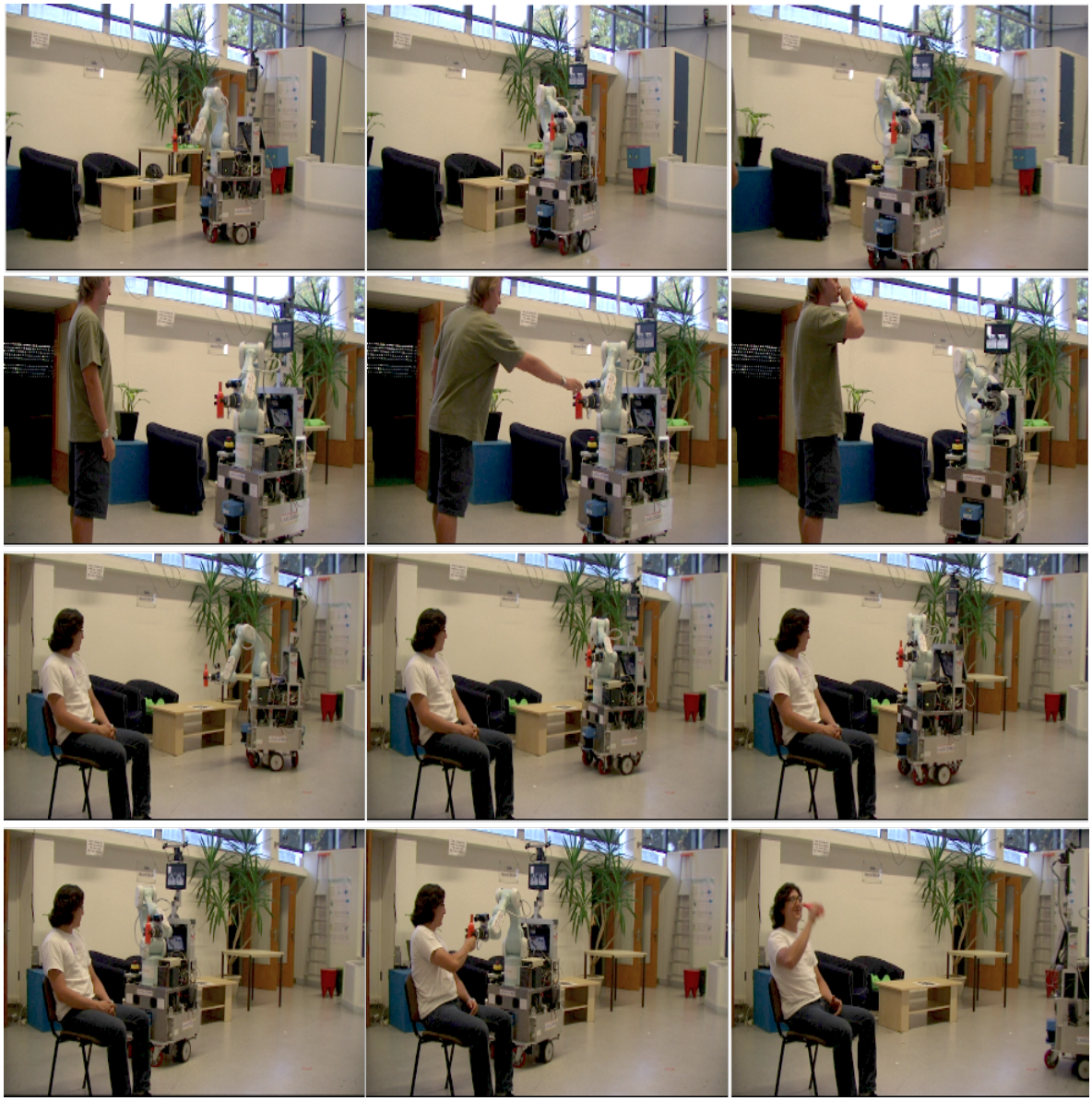
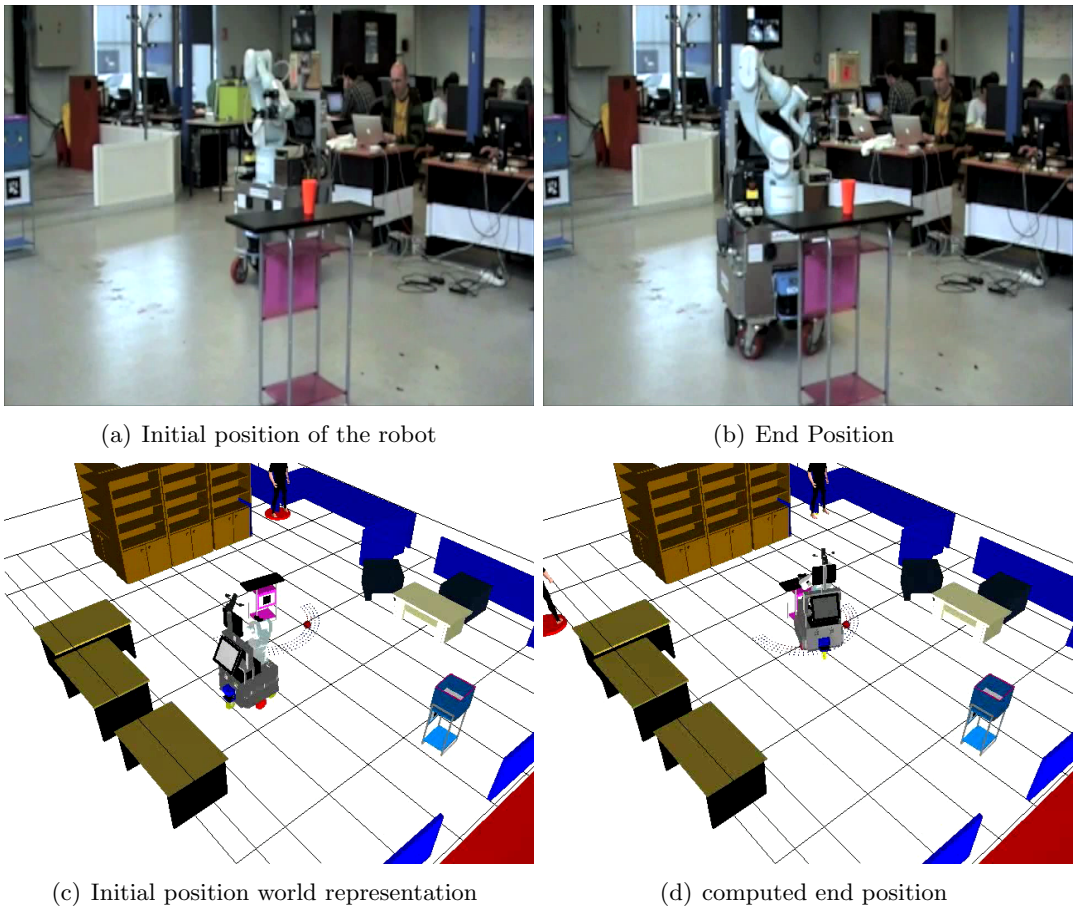


Figure 4.10: A whole "Fetch and carry" scenario, sequence where a robot has GoTo-Give task to a person that is standing and to a person that is sitting. We can see the difference on the approaching behavior of the robot. On the second case the robot takes into account that the human is in a sitting position but also the closest position to the robot current placement.

The first configuration is placed to see the table where the bottle is located, maximum and

minimum distances are set according to arm capabilities for grasping the bottle. Second configuration is found to get close to human and give him the bottle.

As described above, the MHP module creates a 3D representation of the world according to its sensed data and updating the state of the world on Move3D system where the HAMP and PSP are integrated. This allows computing a position based on real data. In the figure 4.11 is illustrated how this world is represented on the Move3D system.



(a) Initial position of the robot

(b) End Position

(c) Initial position world representation

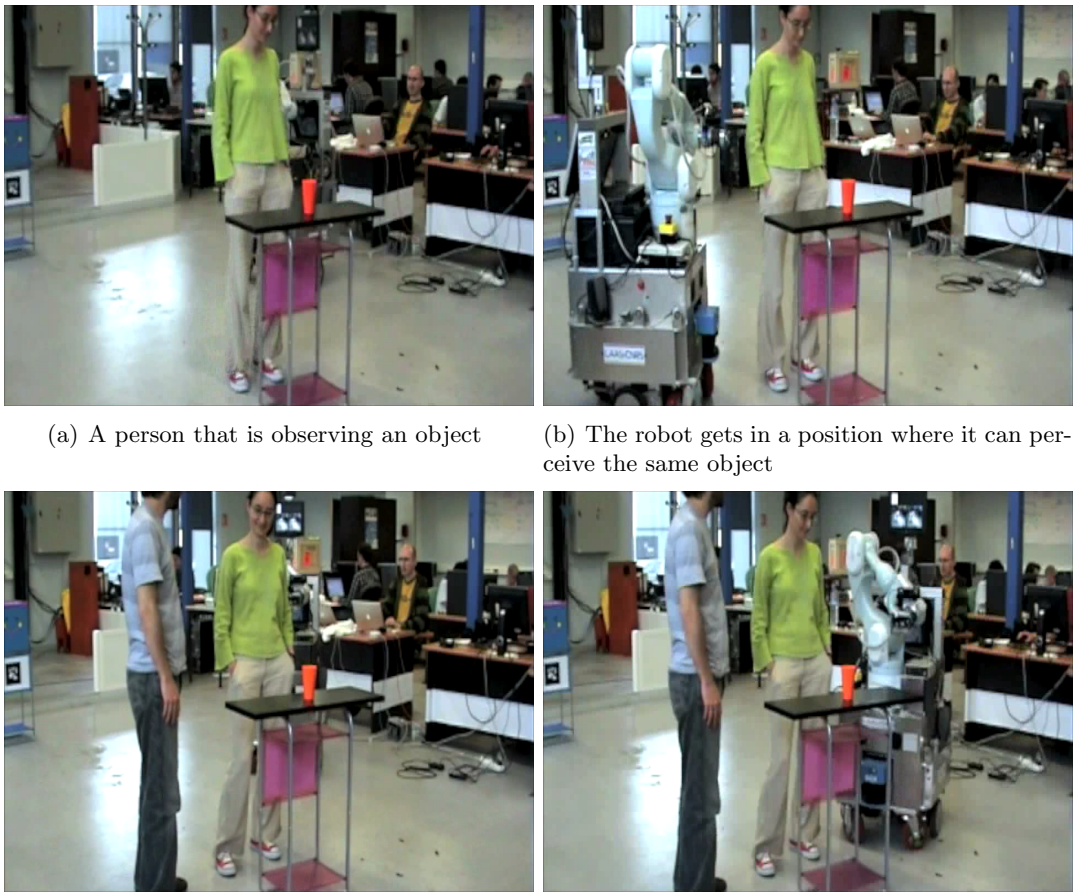
(d) computed end position

Figure 4.11: Scenario for a GoTo-Pick task, the robot approaches the object in a normal situation where there is no collision/visual obstacles nor humans on the proximity of the trajectory or the final target.

Obtaining a configuration where the robot can perceive an object where there are no visual obstacles is relatively an easy problem to resolve, the problem comes when the robot has to obtain a good position where one or more obstructions for the robot perception are present. On the figure 4.12 we can appreciate the case where the robot is capable to find different configurations to avoid visual obstacles, depending on what the robot senses of the environment.

As the representation of the human obtained by the human detection modules on the system architecture, what is sensed is also represented on the robot representation in real time.

Another example is illustrated in Figure 4.14, where not only visual occlusions are taken into account, but also the robot avoids collision with known obstacles on the environment verifying the human preferences, trying to get little farther from the human when he/she is sitting.



(a) A person that is observing an object

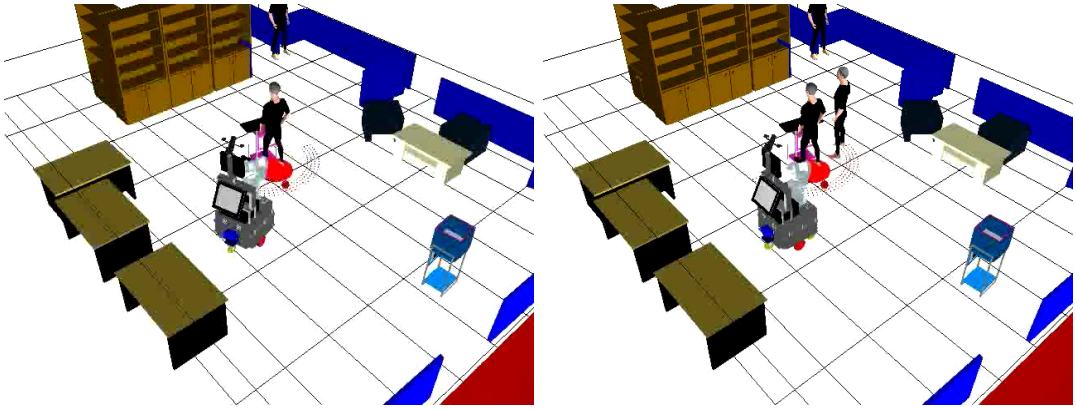
(b) The robot gets in a position where it can perceive the same object

(c) Another scenario where person are perceiving an object

(d) The robot finds a new position where it can achieved its task

Figure 4.12: The robot finds safe, comfortable and "understandable" positions where it can achieve its task. In the first scenario only one person is preventing the robot to place itself to see the table, the robot finds a position where it can see the table treating the person as a human and not as a normal visual obstacle. On the other scenario the second person occupies the position that the robot found on the first scenario, and occluding the target from the visual perception. The robot adapts the point utility and changes its position.

Here we can also observe that an obstacle (a table) prevents the robot from several positions close to the human to bring him the bottle.



(a) World representation for 1 occlusion (b) World representation 2 occlusions

Figure 4.13: The representation of the MHP module environment and of the human detected using HUMPOS module. This is the representation of the scenario presented on figure 4.12. The human detection modules updates the representation of the world in real time based on the robot sensors data.

(a) Initial position human sitting (b) End Position

(c) Initial position (d) End Position

(e) Giving the object (f) Taking the object

Figure 4.14: PSP to hand an object to a human sitting in front of a table a) and c) Initial positions, b) and d) Final configurations looking at the human. The robot is perceiving human by cameras on top. The robot finds its final configuration to complete the tasks of giving e) or taking f) the object.

4.6 Discussion

In this chapter it is presented the integration of the Perspective Placement planner into a mobile robotic platform. This planner is encapsulated in a Genom module, called MHP, among the motion planner in the LAAS architecture.

As illustrated in the general architecture, this module plays an important role in the interaction task related with motions. It is clear that this module lays on the human detection and self-localization modules, which allow update the environment representation inside the MHP module.

The system was evaluated in the frame of the FP6 Cogniron Project [Cogniron 08] and also was presented in the FET conference [FET 09] as a demonstration stand, with "normal" people. We can observe that the results help on the accomplishment of the task. We are planning to perform different user studies for better adapting the system behavior.

5 Perspective Taking on H-R Joint Attention

5.1 Introduction

Human Robot Interaction also brings new challenges to the geometric reasoning and space sharing. The robot should not only reason on its own capacities but also consider the actual situation by looking from human's eyes, thus "putting itself to human's perspective".

In humans, the "visual perspective taking" ability begins to appear by 24 months of age [Moll 06] and is used to determine if another person can see an object or not. In this chapter, I propose a geometric reasoning mechanism that employs psychological abilities of "perspective taking" and "mental rotation" in order to reason what the human sees, what the robot sees and where the robot should focus to share human's attention.

In the first section, it is shown the adaptation of robot mechanisms to egocentric perspective taking (presented in the Chapter 3) for modeling the human perspective. These mechanisms, represented here as geometric tools, are used to achieve a bidirectional visual attention on the same object, by understanding the human perception.

In section two, this geometric reasoning mechanism is demonstrated with HRP-2 humanoid robot in a human-robot face-to-face interaction context.

At the final section we will discuss the chapter as mention the perspectives of this work.

5.2 Geometric Tools for Shared Attention

Attention sharing requires psychological notions of perspective taking and mental rotation taken into account in robot's reasoning. As it has been mentioned in previous sections, perspective taking is the general notion of taking another person's point of view to acquire an accurate representation of that person's knowledge. In the context of this work, I am interested in visual perspective taking where the robot should place itself to "human's place" to determine what he is actually seeing.

5.2.1 Perspective Taking of the human

To closely interact with human, the robot has to represent in some way the shared space and the objects that belong to this space. Added to this, the robot has to understand the human perception of this space by performing a "perspective taking" of the human.

In order to find out what human is seeing, I proposed to attach a virtual camera into "human's eyes" in its model within Move3D [Siméon 01] simulation and planning environment. The attached camera will move as the configuration of the human model changes. To identify what this camera perceives, we use 2D perspective projection of the 3D environment.

This projection is obtained from an image taken from the human's eyes point of view (and not robot's).

The obtained result is the matrix $MatP_i$ where the value of the position (x_i, y_i) represents one point in the projection image of the object i inside human's field of view. A 2D projection of the scenario shown on Figure 5.1 is illustrated in the Figure 5.2.

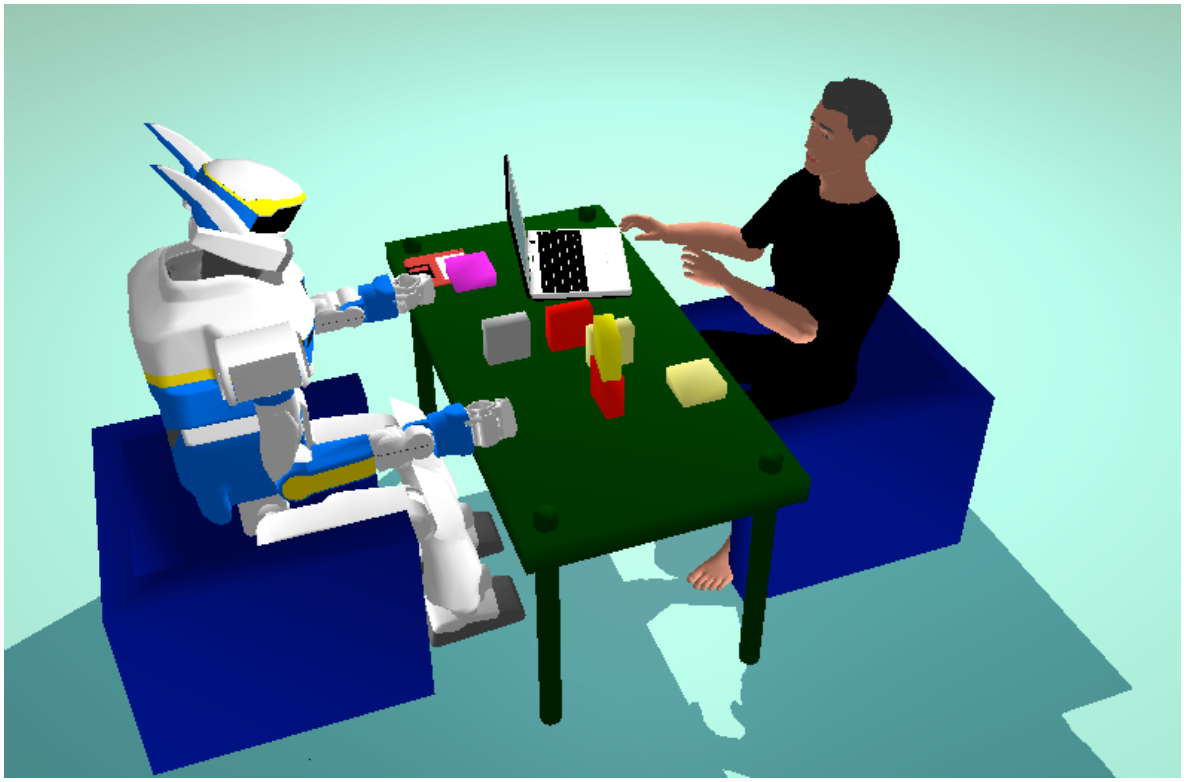


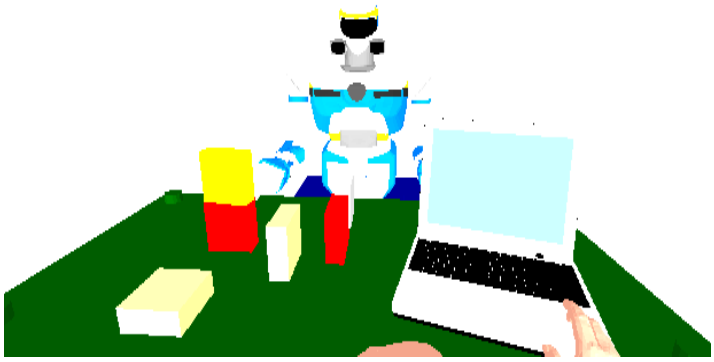
Figure 5.1: A scenario where the human and the robot are sitting face to face

The 2D projection image, which is the result of this *mental rotation* process in order to perform *perspective taking* of the human, represents the points of the environment that the human is actually seeing.

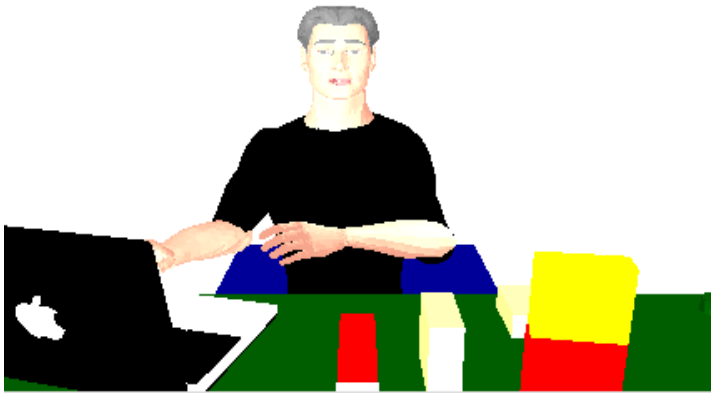
Even though the information on visible and invisible points is interesting, for a HRI scenario the most important information that can be extracted from this image is which objects, humans, obstacles or robots are actually seen by the human. This image will be used as an input of perspective taking mechanism that computes the visibility of each body in this image.

The acquiring process of the human perspective projection on the robot's representation is obtained on a similar procedure for obtaining the robot's *egocentric perspective taking* explained in section 3.4.2. The difference is that here, the robot has to reason not only on its own perspective but also on human's, having with this two perception models (camera position, aperture angles, etc).

Figure 5.3 illustrates the difference between desired and visible relative projections of the robot



(a) Human perspective



(b) Robot perspective

Figure 5.2: Computed perception of the robot (top) and human (bottom). The perception depends on the sensor capabilities and configurations

from human's point of view. As the perspective taking system reasons the visibility by taking into account everything in the 3D environment, including the human himself, human's hand causes a small visual occlusion on the laptop (figure 5.3-c and -d).

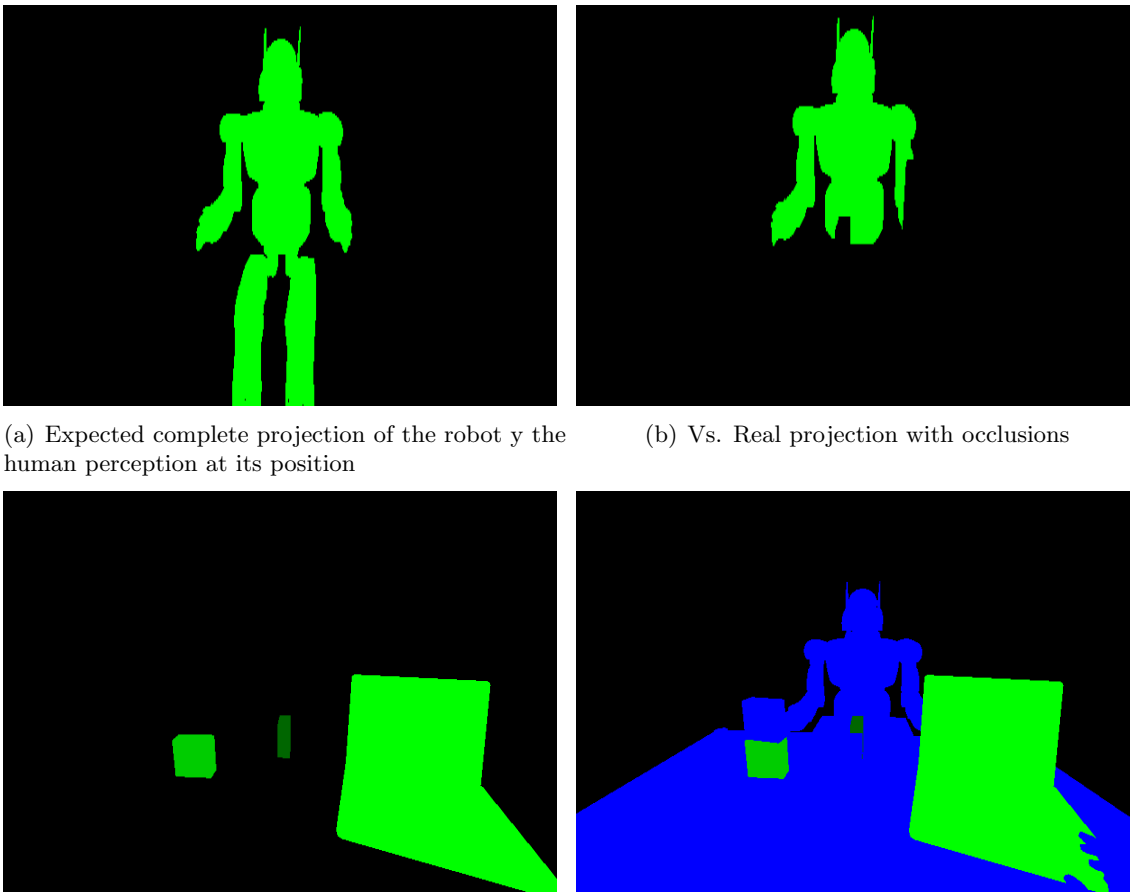
Visibility quality percentage of each element El_i on the human perception is defined by $WatchMulti$:

$$WatchMulti^n{}_i = \frac{Pr_{visible}(El_i)}{Pr_{desired}(El_i)} = \{Watch_1, Watch_2, ..., Watch_n\}$$

where n is the number of elements on the human field of view and Pr_x are the respective projections of each element i.

The objective of knowing the perception element or elements on the human perspective, is to get information of the objects that the human is paying attention, the gaze following can also give us the information of the direction of human gaze but we don't

Performing Gaze following can contribute to obtain information of the attention objects of the human, as it is used by many authors [Scassellati 99, Peters 08, Huang 08, Nagai 03, Sumioka 07]. However, it is difficult to acquire enough information to know the occluded objects inside the human perception, it is necessary to have a notion of the space, and especially 3D space. Moreover,



(a) Expected complete projection of the robot y the human perception at its position

(b) Vs. Real projection with occlusions

(c) Expected complete projection of an object

(d) Multiple objects perceived by the human,. The computer is almost totally perceived by the human (partially occluded by its own hand)

Figure 5.3: Relative projections: The robot is the target and differs from other elements on the environment. The laptop on the table is the target and differs from other elements on the environment. As the perspective taking system reasons the visibility by taking into account everything in the 3D environment, including the human himself, human's hand causes a small visual occlusion on the laptop. a),c) Desired relative projection b),d) Visible relative projection. The table and the objects are blocking the human's view.

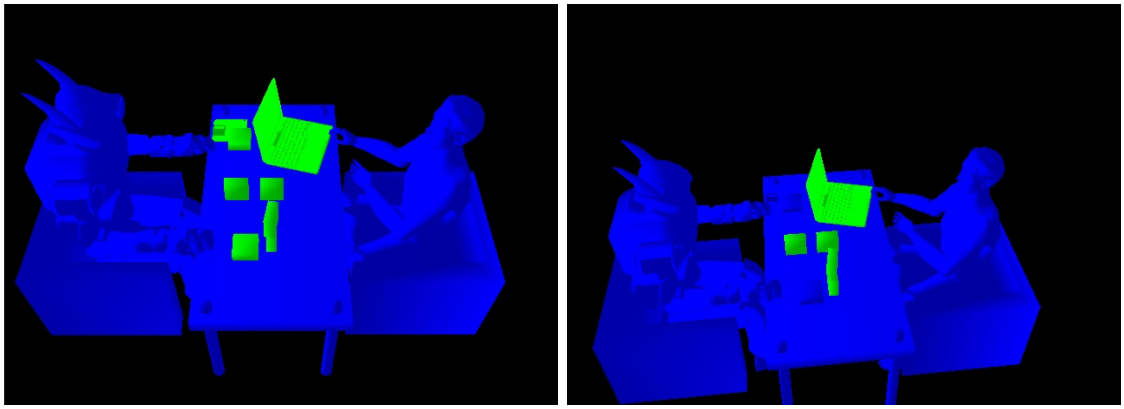
this notion can help the robot to know how the human perceives these objects, applying perspective taking level 2.

5.2.2 Mutual Seen Objects

Based on this process it is possible to specify the objects that are, or may be, on the mutual perception this by the intersection of the list obtained by dual mechanism of perspective taking on the robot and on the human. Expressed as:

$$E_{joint} = E_r \bigcap E_h \tag{5.1}$$

where the Elements inside the mutual perspective E_{joint} are obtained by the common elements perceived by the robot E_r and the by the human E_h. Figure 5.4 shows how the robot can select the objects that are inside the mutual perception.



(a) Robot perceived objects (b) Mutual perceived objects

Figure 5.4: Mutual perceived objects from both actors. The robot can perceive more objects from its position than the human.

5.2.3 Human Attention Objects

Some objects in the environment can be considered as fixed obstacles and can be excluded from the perspective taking. This functionality can allow the robot to react according to the context and human's activity. In the scenario illustrated by Figure 5.5, the table is considered as an obstacle and is not returned by perspective placement system. In the context of a person sitting next to a table, if the task is interacting with little movable objects, we can consider that the attention of the human will be mainly on the object on the table but not on the table itself.

The human is provided with a wide field of view. Nevertheless, when it is centering its attention to something the visual attention reduces its size to a particular cone form as it is explained in [Müller 05]. The objects outside the attentional field of view can also be ignored from the perspective taking process.

On the final elimination step, the robot analyzes the list of objects into the perspective process and like that we can obtain the objects perceived from the human's perspective.

Although all the process of elimination of the non-attentional objects (objects that are not of interest), it is possible to have several objects that can enter in the attentional field of view and that are not occluded by another object. Perspective taking of the human gaze is not enough to determine an attentional object, so that we have to consider temporal constraints of attention. A person has to spend a little amount of time on an object to make this an attentional object.

Finally, is possible that exist ambiguity on the object of attention, this is normal even some human are more capable than other to determine what is perceived by other persons. In the case of ambiguous attentional objects, we take the first closest object to the line in the center of the visual attention cone. In other words the closest to human's line of perception. For acquiring more reliable data, a process of data fusion with utterances and context should be applied to the process.

A snapshot of a scenario where a person is sitting on a table is illustrated in figure 5.5. In this example, the human is looking at the laptop. By using mental rotation and perspective taking systems, the robot determines that the object human is focusing is the laptop. Although the human looks also at bottle's and white box's direction, these to objects are evaluated as invisible by the system because of the occlusion of the laptop.

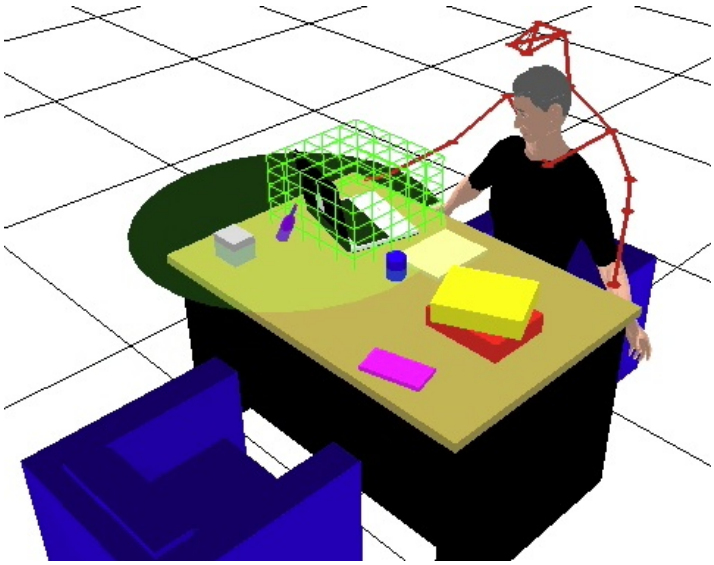


Figure 5.5: An instance of the situation assessment. Object marked with a green wire box (the laptop) is evaluated as visible. The violet bottle and the white box are not visible to the human because of the occluding laptop.

5.2.4 Human pointing gesture

Usually, persons in their communication use gesture movements to indicate places or objects (specially with hands and fingers), to point over objects which they may reference. In a Human-Robot interaction the utilization this kind of gestures can be useful for a fluent communication. To achieve this the robot has to take into account the place where the human (the one with whom it is interacting) is pointing and which objects can this person perceive from its place.

In order to establish the pointing region to select the candidate pointed objects, we define a cone of which origin is in the center of the human body part used to pointing, and the base is in the extremity of the line segment the same direction. The angle of the cone will represent natural uncertainty of pointing from humans, around 72 degrees [Pfeiffer 08], in addition to the uncertainty of human detection performed by the robot due to its sensors. We consider that candidates to be selected have not only to intersect this cone, but also its gravity center has to be inside the cone in order to be taken into account as a possible pointed object.

The fact that several objects could be inside the pointed region, extends the problem to the uncertainty of knowing specifically the object that is been referenced. For this reason we have to define a preference function f from each object Obj_i is a weight value where the robot is more attracted to orientate its camera and is obtained like:

$$f(Obj) = w_1 Dist_{coneline}(Obj) + w_2 Dist_{coneorigin}(Obj) \tag{5.2}$$

where w_1 is the weight of the linear distance $Dist_{coneline}$ from object center point to the cone center line, and w_2 is the weight of $Dist_{coneorigin}$ linear distance between object center point and human pointing extremity[1]. Figure 5.6 illustrates the distances that are considered in the preference function f.

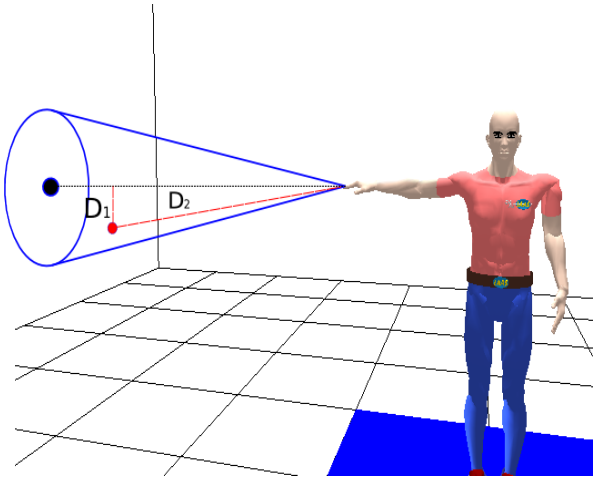


Figure 5.6: A human pointing with its right hand. The cone extends from the hand to the pointing direction. The point inside the cone are measured taking as parameters $Dist_{coneline}$ (D1) and $Dist_{coneorigin}$ (D2)

Sometimes, the information acquired about pointing indication is not sufficient to know which is the object that human is making reference. That is why robot has to take into account person's visual perspective and obtain a list of objects that is possible to be perceived by him.

Perceived objects are more probably referenced by human than hidden ones, then in order to assign an ordered sequence of candidate pointed objects, a value of pointing preference is calculated for each object. This value is obtained based on its proximity to the pointing cone obtained by f function, in addition to object's visibility for human, in other words, we take into account the human perception into the pointing direction, even when human is not looking at this direction, by computing the projection on human's perspective with his head oriented to the pointing direction. This is established by it's the $Preference$ function and is represented like:

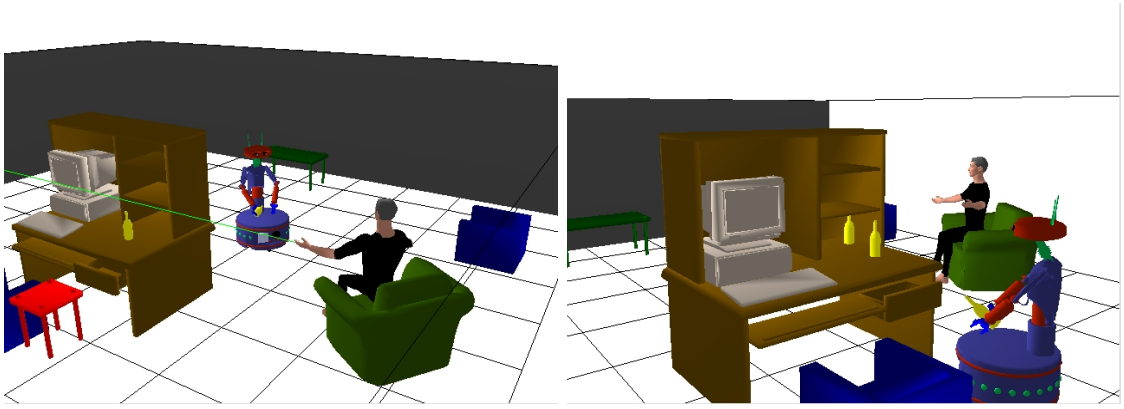
$$Preference(Obj) = \sigma_1 f(Obj) + \sigma_2 Watch(Obj) \tag{5.3}$$

Where Obj is the object, $f(Obj)$ is the preference pointing function explained before and $Watch(Obj)$ is the visibility function, and σ's are weights given to each function.

The object with greater value of preference will be considered as referenced by human, and after that the robot will proceed for the next interaction task.

Figure 5.7 shows an example of two objects one in the human's field of view and in its relative projection. Here there is are two bottles in the pointed direction but only one can be seen by the person who is pointing, the robot can see both of the bottles and it has to take a decision about which one is the more probably referenced object.

[1] Adapted values for w's is 0.8 and 0.2 respectively



(a) The human points for a bottle on the desk direction

(b) in other perspective there are two bottles on the pointend region

(c) Robot's perspective

(d) Human computed perspective, with head towards the pointed direction

Figure 5.7: There are two objects on a pointed region. The robot has to deal with the uncertainity by computing the possible perception of the human if he were looking at the pointed direction. Objects in the shared visual space are more possible to be referenced than hidden objects[Trafton 05a]

5.3 Integration and Results

5.3.1 Scenario

The experimental environment implements a Face-to-Face interaction scenario of human and our Humanoid Robot HRP2. The table serves as a common work platform and the objects on the table are a toolbox, a small box and a cup. Figure 5.8 shows the real scenario and figure 5.10 shows its 3D representation in the interface of our Move3D [Siméon 01] software platform, where the geometric tools are implemented.

Note that apart from using the static model of the environment, our system puts objects dynamically in this 3D model, which has been described in next section.

Figure 5.8: Face to face scenario. The table is the common work platform for the interaction objects.

5.3.2 Implementation and Results

The entire system has been carried to our HRP2 robotic platform. HRP2 is a humanoid robot developed by Kawada Industries, Inc. It has 30 degrees of freedom. In LAAS-CNRS, it has a vision system composed of four cameras on its head. The robot's height is 1570 mm and its width is 613 mm. Its mass is 58kg, including batteries.

The system architecture is consisting of various task-specific dedicated OpenGenom modules [Fleury 97]. The environment Move3D is managed by one of these modules, called GEO, in order to interface it with the other modules. A scheme of the system architecture is illustrated on the figure 5.11, showing all the data flow with the GEO module.

Two modules mainly do the acquisition of the dynamic changes in the environment: a vision

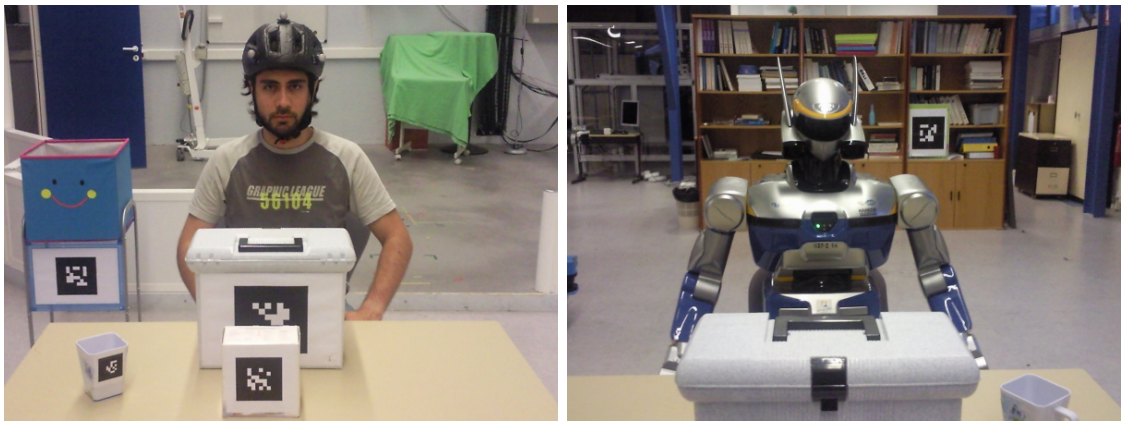


Figure 5.9: The scenario from robot's and human's eyes.

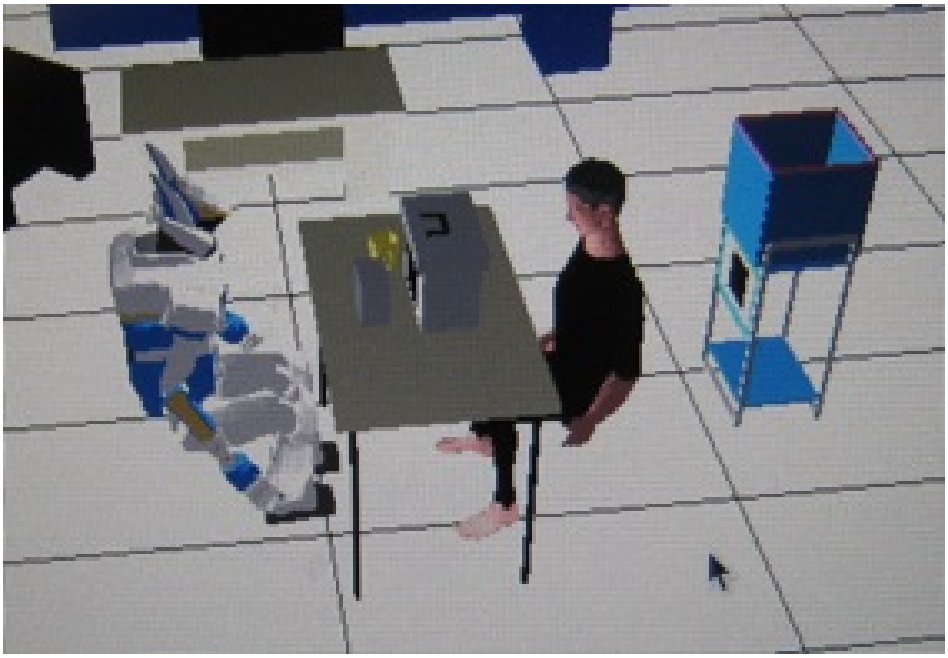


Figure 5.10: The 3D representation of the same environment including the robot, the human and the objects.

based module *ViMan*[2] and head markers detection based on motion capture system. ViMan Module uses tag on the object to identify and calculate its position in the 3D space. The GEO module continuously obtains this 3D positions and orientations, and then updates the environment placing models of the objects on the table (or elsewhere) dynamically, at the moment of its detection.

As a temporal platform for obtaining a precise motion of the human head as also the gaze orientation, a Motion Capture system was installed in the experimental environment. It consists of 10 cameras at different positions, which covers a volume in the environment.

[2]ViMan module is still in development, it is mainly conceived and implemented by Xavier Broquer and Jean-Phillipe Saut

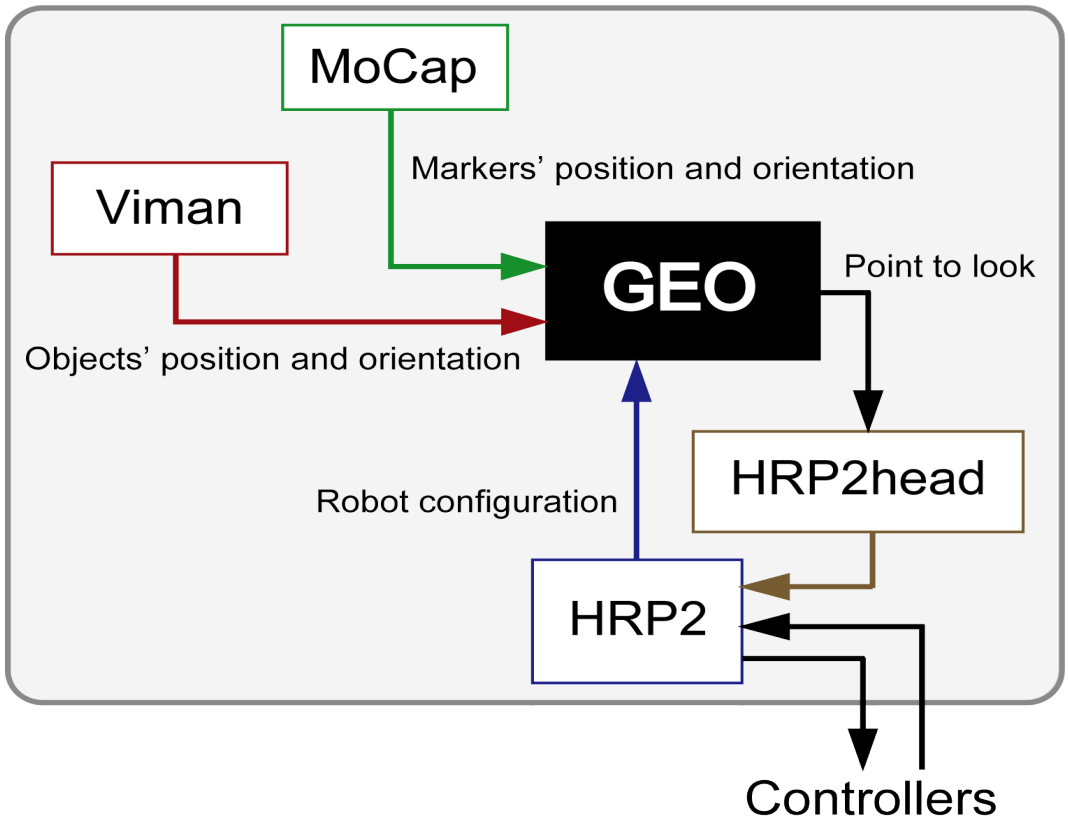


Figure 5.11: The system architecture. The GEO module receives markers positions from the MoCap Client and sets the human position and head orientation. From Viman, object positions are obtained and updated. Once the reasoning about human perspective is done, robot head configuration is passed to the controller modules.

The person, with whom the robot is intended to interact, is equipped with a special cap consisting of a set of markers. The server of the motion capture system broadcasts the position of the markers, a dedicated client running communicate to the server and update the position of the markers. Our GEO module acquires data from this client, interprets markers position and geometrically calculates the orientation of the human head in real time.

Once the perspective taking process is done and the attentional object has been defined, the point of the center of gravity on this object is sent to the *HRP2head* module that will take in charge the head motions to look at the attentional object.

Figure 5.12 shows the simplest face to face scenario. It illustrates the results on an image sequence from different videos; here the robot turns the head each time it detects that the human is changing of attentional object. The attentional object can be appreciated because it is on the center of the image of the robot's camera.

Figure 5.13 illustrates the results when two objects are in the same field of view but one of them is blocking the other. The attentional object is the one that the human can see, and not the hidden one.

Figure 5.12: Scenario 1: the robot looks to the object that the human is looking. In the images: The scenario (up), the GEO-Move3D interface showing the process and the attentional object marked with a green grid box around the object (down-left) and robot's camera (down-right)

Figure 5.13: Scenario 2: the robot is capable of detecting and looking at the object of attention, detecting visual occlusions between objects. The toolbox is occluding the small cup to the human.

5.4 Discussion

In this chapter it has been presented a first step of the development of a set of useful geometric tools that helps to the development of a shared attention in human robot interaction.

I have also shown the importance of the implementation of algorithms based on perspective taking and mental rotation concepts to obtain visual attentional objects.

Furthermore, I have not only developed the geometric tools on 3D simulation environments but also I have shown our implementation on a humanoid robotic platform, and obtaining promising results. Nevertheless, there is still some work to do on the evolution and improvement of the system.

At this step of the implementation, we have activated two joints on its neck. This means that, at this step of the integration of the geometric tools, it only work on a static behavior in a face to face interaction, moving only the robot's head to look at the same object that the human is looking at.

The system is intended to perform more complex shared attention and interaction actions. In the very near future, this system will be able to plan sensor based configurations and motions using its arms, hand, and waist joints, based on Inverse kinematics and collision detection. All this, integrating with other systems of human aware motion planning that also use perspective taking and mental rotation concepts, as seen on the previous chapter.

Also, we are currently working on process of influencing the human visual attention, reasoning on human perspective and tracking the human gaze on objects that the robot is currently manipulating. This will allow the robot to perform human understandable actions to achieve joint activities of interaction.

6

Conclusions and Perspectives

Psychological studies on human interactions are important to know how they use spatial reasoning abilities to achieve collaboration between them, and also an ability to reason about the current beliefs of the robot, the human and mutual beliefs. Perspective Taking and Mental rotation are two of these large number of abilities that helps to ease the interaction and understanding on humans, and that can help to the development of other abilities as the shared attention.

On this work we have presented how these human capacities can be introduced on computational mechanisms. Once integrated these abilities, they can help on the interaction between machine and humans. The integrations was shown in one hand for motion planning, in other hand for understanding human attention.

In the first part of the dissertation we have introduced to the concepts of perspective taking, mental rotation and joint attention through a short survey from different areas. We have started from the definition of this notions from the point of view of the psychological studies made on humans and animals, then we have presented different approaches to tackle the problem of integration of these abilities either implicitly or explicitly, on computer systems and robots.

On the second part, we explain the adaptation of the concept of perspective taking to a robot motion planning environment. The presented configuration planner called PSP, use techniques used by humans of rotating scenes or objects. All this is done by computing the perspective from different positions on the environment.

With the PSP system the robot is able to:

- Have a destination position for the navigation planner
- Have a configuration where the manipulation planner can start to plan its "giving" or "picking" task.
- Have a well defined interface between the higher level commands, coming from the Supervisor.
- Have a method of evaluation for the task goal achievement.

All this focussed on the human presence in all the steps of design the system by being closely integrated in the Human Aware Motion Planner and inside an architecture conceived explicitly for human robot interaction.

In the last part of this work, we have shown how our geometric spatial reasoning tools serve on understanding human attention or references, and also how to be understood by the person with whom the robot is interacting. The presented experiments with a humanoid robot, took us to notice that most of the authors that have tackled the shared attention on human robot interaction, perform gaze following to obtain the human attention. This tells us two things about gaze following:

- Is not possible to achieve *perspective-taking* level 2 if we only follow the gaze orientation.
- The robot can't perform *joint attention* stage 2 to 3 with only gaze following
- Is difficult to detect occlusions on the visual attention with only 2D information of the human gaze.

With the presented framework, it is possible to perform these abilities, by computing the human perspective, in an autonomous way.

Nevertheless, at this state, this method presents some drawbacks: it is very dependent of the graphic capabilities of the computer. It is also dependent on the human detection and on the object modeling methods. But those two problems are common on all the actual systems.

The perspectives presented previously are only a part of the list of future work; on the chapter 3:

- Learning method for the weight cost modifications, the robot can adapt its behaviors measuring its rewards based on the utility.
- More taks goals considerations, GoTo-Check or GoTo-Wait are examples of task that can be added to the utility model.
- Further search point methods, for having the best utility on the less time.
- Considering stable configurations for humanoid robots on the position selection.
- Linking task planner with motion planner, the plans of tasks must be validated with the motion actions and vice versa.

On the chapter 4:

- Influence on human attention, the robot has to adopt motions as for example "pointing gestures" in order to act on the human attention.
- "Attention seeking" and acknowledgment of new status, the robot cannot be able to know at every moment where the human is paying attention, a system that continuously veriÔ¨Åes what the human is attending is necessary for achieving a more ‚Äúnatural‚Äù interaction.
- Human Gesture recognition, we believe that the recognition human 3d configuration, must be completely integrated on the robotic platform and independent from other sources as the motion caption system.
- integration in a whole system of close interaction with human, manipulation planning, vocal utterance generation, voice recognition, as other high level systems must be working on a complete harmony in order to achieve a "natural" interaction between the robot and the human.

At the end, we can say that this is a contribution work on the integration of multidisciplinary systems, for the accomplishment of the human-robot interaction.

7 Résumé

7.1 Introduction

Un robot personnel qui puisse emporter quelque chose à boire ou à manger quand on arrive à la maison ou une machine qui se rappelle toujours de vous donner vos médicaments quand vous êtes malade; C'est mon propre rêve et c'est aussi une chose qui peut aider non seulement les gens paresseux comme moi, mais également les personnes avec des capacités de mouvement réduites.

L'insertion d'un robot dans la vie quotidienne de l'homme, génère de nouvelles questions, auxquelles la recherche doit répondre. Le partage de l'espace humain oblige le robot à raisonner sur comment se placer pour interagir avec l'homme, et ne pas le perturber. Il faut rajouter à cette interaction, une compréhension des intentions humaines par le robot pour accomplir une meilleure interaction. Le robot devra donc, adopter différents mécanismes de raisonnement humain, comme la "prise de perspective" et la "rotation mentale".

Ces deux concepts proviennent des études psychologiques sur l'interaction entre personnes. La prise de perspective consiste à se mettre à la place d'une autre personne et percevoir l'environnement de son point de vue. Il est possible de prendre sa propre perspective en appliquant une "prise de perspective egocentrique". Dit d'une autre façon, faire pivoter une image dans l'esprit pour se faire une idée la perception de l'environnement à partir de différents endroits.

La prise de perspective peut être utilisée par le robot afin de s'approcher de l'homme, ou pour calculer simultanément où le robot peut se placer et effectuer certaines tâches d'une manière compréhensible par le partenaire humain.

Atteindre une compréhension bidirectionnelle dans une interaction, implique non seulement de raisonner sur ses propres actions, mais aussi de raisonner sur les intentions du partenaire. Pour arriver à faire ceci, il est indispensable que le robot puisse interpréter le centre d'attention de l'homme et lui faire comprendre qu'il a bien compris sa référence en partageant l'attention sur un même objet.

Dans ce manuscrit, il est présenté différents outils de raisonnement spatial où la prise de perspective est utilisée pour aider des robots à réaliser différentes tâches d'interaction.

7.1.1 Contributions

Dans ce travail nous introduisons un ensemble d'algorithmes qui apportent des "capacités sociales" au raisonnement géométrique du robot. Notre plus grande contribution sur la planification

de mouvements, est l'adaptation de critères sociaux et l'adaptation d'un modèle de la vision humaine dans la recherche d'une configuration et d'un emplacement du robot pour l'interaction. Cette adaptation remplit les objectifs des tâches, comme la perception de l'objectif, ainsi que la minimisation de la distance de parcours du robot.

L'approche présentée propose un cadre de travail qui sert comme :

- Un lien entre un planificateur de navigation et un planificateur de manipulation pour accomplir des tâches.
- Un lien qui interprète des instructions de haut niveau et qui les transforme en buts spécifiques pour le planificateur de mouvement.
- Une façon d'évaluer l'accomplissement de la tâche
- Un schéma de raisonnement sur l'espace à travers les "yeux de l'humain" pour aider le robot à comprendre et être compris par la communauté humaine.

7.2 État de l'art

Il y a trois concepts qui viennent de la psychologie et qui seront abordés au fur et à mesure dans ce texte. Ces concepts sont, selon certains auteurs, indispensables pour l'interaction Homme-Robot afin d'établir une fluidité lors de la communication [Baron-Cohen 95b, Frith 01]:

- Rotation Mentale : La capacité de tourner des représentations mentales en deux ou trois dimensions.
- Prise de perspective : Le concept principal de ce travail et qui fait référence à la capacité de comprendre et de percevoir ce que les autres personnes disent, font ou regardent, en fonction de son point de vue. On se focalisera sur la " prise de perspective visuelle" (ce que les autres voient à partir de l'endroit où ils se trouvent).
- Attention Partagée/Jointe : Le fait que deux individus ou plus, font attention au même objet ou endroit pour une intention commune de communication.

Selon Flavell [Flavell 92], la prise de perspective est acquise chez l'homme en deux niveaux. Dans le premier niveau l'humain est capable de distinguer les objets que son partenaire est capable de voir. Dans le deuxième niveau une personne est capable de se faire une image mentale de la perception de son partenaire, autrement dit, elle est capable de se mettre dans la tête de son partenaire.

Tversky, Taylor et al. [Taylor 96, Tversky 99, Lee 01] ont fait différentes études sur la manière dont les humains changent leur perspectives. Ils concluent que pour qu'une personne en comprenne une autre, le développement de la prise de perspective est un outil nécessaire.

La prise de perspective est utilisée comme base pour développer des algorithmes et systèmes qui permettent une interaction plus fiable entre machines et personnes. C'est le cas pour la réalisation de systèmes d'images de synthèse en 3D. Où la perspective de l'homme doit être simulée pour permettre une immersion dans l'ambiance virtuelle [Zhang 97, Chhugani 05, de Sá 98] [Menchaca-Brandan 07].

Dans la Robotique, cette capacité humaine est aussi représentée sous différentes formes, par exemple pour le développement de simulateurs de robots [Faust 06] ou pour la modélisation

d'objets [Pito 99, Wang 06, Foissotte 09] avec differents capteurs de robots, comme par exemple capteurs laser ou cameras de video.

L'interaction Homme-Robot me l'accent sur le sujet en commençant par prendre en compte le champ de vision de l'homme [Richarz 06], ou encore plus explicitement prendre en compte la perspective de la personne en considérant les objets qui peuvent être visibles ou pas pour l'apprentissage du robot [Trafton 05a, Trafton 05b, Breazeal 06, Berlin 06] guidé par l'homme.

D'autre part nous avons la capacité humaine d'avoir une attention partagée dans une interaction face-à-face. Largement étudié par les psychologues, ce concept se réfère au fait d'être attentif au même objet. L'attention partagée peut être présentée en 3 types dans les premières années de vie humaine [Tomasello 99]:

- Vérification de l'attention : Lorsque l'on vérifie que la personne fait attention a soi-même.
- Suivi d'attention : Quand la personne est capable de suivre la direction du regard du partenaire.
- Diriger l'attention : Quand la personne est capable de manipuler l'attention de l'autre personne.

Cette capacité a été aussi représentée dans diverses études sur l'interaction homme - machine. Quelques chercheurs se focalisent sur la mesure de l'engagement en l'interaction [Peters 08], d'autres sur l'identification de l'objet d'intérêt [Huang 08, Nagai 03, Sumioka 07]. Les deux types d'études se focalisent aux suivi de la direction du regarde.

Kaplan [Kaplan 06] revient à la classification de Tomasselo et mentionne que pour arriver à une meilleure attention partagée entre l'homme et le robot, il doit exister au moins quatre préconditions:

- Détection de l'attention : pour suivre l'attention de la personne.
- Manipulation de l'attention : pour diriger l'attention de la personne.
- Coordination sociale : pour arriver à réaliser des actions coordonnées.
- Compréhension Intentionnelle : pour se rendre compte si les deux individus sont en train de partager l'attention et atteindre le même but.

Scassellati [Scassellati 99] divise l'accomplissement de l'attention partagée en quatre tâches : regard mutuel, suivi du regarde, pointage impératif d'un objet et un pointage déclaratif de un objet distant. La plupart des travaux dans la littérature proposent des approches par suivi du regard, c'est à dire l'étape deux.

De façon complémentaire, il existe des travaux qui mesurent comment les personnes dirigent l'attention d'un robot [Imai 03, Ogata 09]. La prise de perspective est aussi utilisée pour pouvoir obtenir les objets d'attention, ces objets étant définis soit manuellement [Brooks 04, Okamoto 05] soit automatiquement [Johnson 05].

7.3 La prise de perspective appliquée à la planification de mouvements.

Dans la robotique il existe différentes méthodes de planification afin qu'un robot puisse se déplacer de façon autonome. Dans ces méthodes, on peut trouver celles qui s'intéressent à la planification de mouvements qui, comme son nom l'indique, calculent une série de configurations du robot pour aller d'un endroit à un autre tout en évitant des collisions avec les objets de l'environnement.

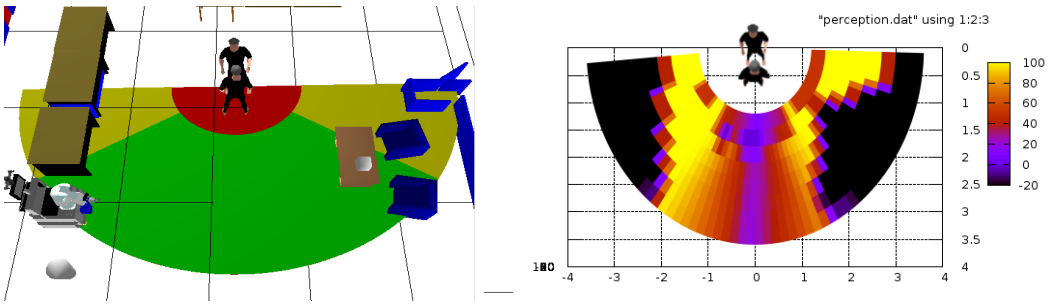
Les planificateurs de navigation, d'une part, ont besoin de connaître la position finale pour trouver la trajectoire depuis l'emplacement courant. De plus, pour l'interaction avec l'homme, il est nécessaire de savoir si la position finale est assez "bonne" pour interagir en respectant les espaces d'intimité ainsi que les préférences de l'homme. D'autre part, les planificateurs de manipulation ont besoin de connaître un emplacement depuis lequel le robot est capable d'atteindre la position désirée.

Il est donc nécessaire qu'un autre type de planificateur puisse trouver une configuration qui permet au robot d'accomplir sa tâche. C'est pour cette raison que nous avons développé Perspective Placement (PSP).

La technique utilisée par PSP peut s'appliquer non seulement pour l'interaction avec des gens mais aussi pour atteindre des objets. Elle consiste tout d'abord à délimiter l'espace de configuration dans une zone proche de l'objet d'interaction (que ce soit un humain ou un objet). Une fois l'espace défini, l'étape suivante est de discrétiser la zone en points sur le sol dans l'espace polaire, et finalement d'évaluer chaque point obtenu lors de la discrétisation pour choisir le meilleur.

L'évaluation est faite en terme d'utilité, basé sur l'assignation des qualités et des coûts. Ces deux propriétés dépendront de la tâche et il s'agit de maximiser la qualité et de minimiser les coûts.

Notre tâche de base est celle de la perception. Dans cette tâche, la qualité est basée sur l'implémentation de la prise de perspective égocentrique sur le robot et la quantité que l'objet desiré est perụ par le robot à partir de chaque position. Le robot doit éviter au maximum les obstacles visuels qui peuvent géner sa perception et nuire au bon déroulement de la tâche. La figure 7.1 montre les valeurs obtenus pour la qualité de la perception.



(a) Deux hommes en train de discuter. L'objectif du robot est de s'approcher de l'humain qui se trouve dans la partie haute de l'image.

(b) Les valeurs de perception changent à mesure que le robot s'eloigne de l'obstacle visuel, dans cet exemple, la personne en face de l'objectif.

Figure 7.1: Scenario avec occultations visuelles et aires de collision. Differentes valeurs de qualité de perception sont assignées à chaque point dans l'aire d'interaction. Les zones noires en b) indiquent les points qui ne sont pas accessibles à cause des obstacles.

D'un autre coté, les coûts sont les éléments qui interviennent de façon indirecte dans les activités du robot. Les coûts ont différents caractéristiques à prendre en compte :

- Distance: Entre la position courante du robot et celle du point à évaluer.
- Sécurité et confort de l'humain : Les personnes ont différentes préférences par rapport à la proximité du robot.
- Coût frontal : Le position la plus favorable est celle où le robot est face-à-face avec l'homme.
- Attention de l'homme : le robot doit essayer d'entrer dans le centre d'attention visuel de l'homme.

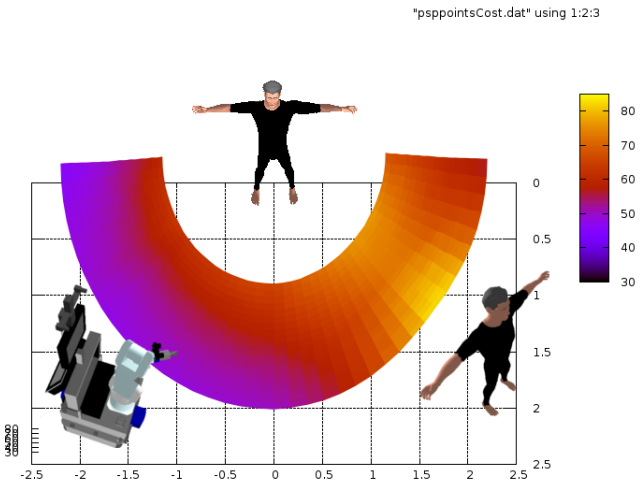
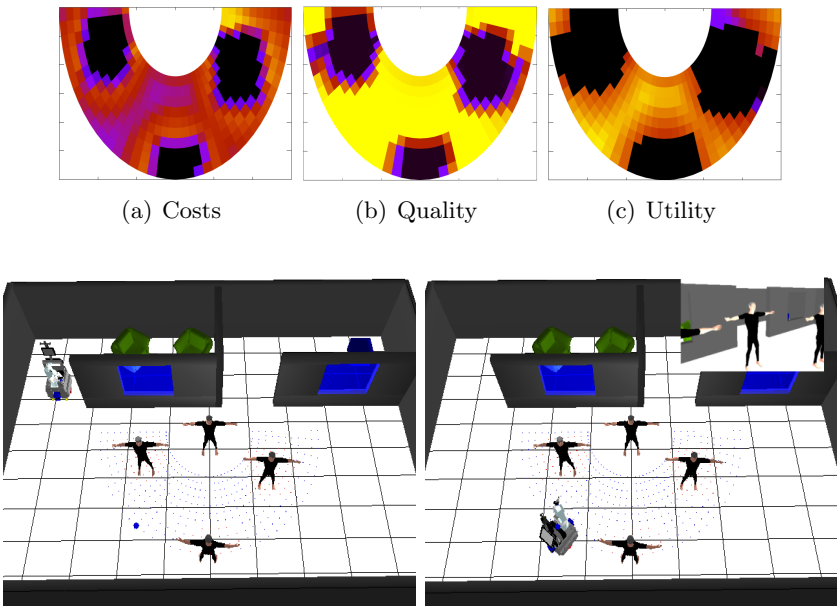


Figure 7.2: Coûts calculés en chaque point. a) les points tout au tour de l'aire de champ de vision. b) points de l'aire d'interaction, les coûts les plus bas sont affectés par la proximité du robot ainsi que la proximité de la partie frontal de l'humain.

On peut observer quelques résultats dans la recherche, d'un meilleur point de vue, dans les images suivantes.

Le planificateur fait partie de notre architecture implémenté sur notre robot Jido. L'approche a également été testée avec des utilisateurs naïfs.



(a) Costs (b) Quality (c) Utility

(d) Initial configuration on the robot position. Final position on the blue sphere

(e) Final configuration and its perspective

Figure 7.3: Le robot doit parler à la personne qui est au milieu supérieur de l'image. Chaque personne qui est présentée dans l'environnement modifie les coûts et les qualités des zones autour d'eux. Le robot trouve une position qui respecte les contraintes imposées par les humains.

Figure 7.4: Un scénario complet d'une tâche "prends et donne". La séquence montre que le robot qui doit emporter une bouteille à l'homme en deux cas; dans le premier cas l'humain est debout et dans le deuxième il est assise, le robot s'adapte à chaque état de l'humain.

7.4 La prise de perspective dans l'attention partagée entre l'homme et le robot.

Pour qu'un robot puisse réussir un raisonnement spatial complet pour une interaction avec l'homme, il doit non seulement prendre en considération sa propre perspective mais aussi le point de vue de l'homme.

L'attention partagée nécessite certaines capacités psychologiques tel que la prise de perspective et la rotation mentale. C'est pour cette raison que le système de prise de perspective a été adapté pour pouvoir trouver les objets observables par une personne dont on connaît la position et la direction de son regard.

Notre système consiste à l'adaptation d'une camera virtuelle dans les yeux de l'humain, puis sur l'identification des objets dans son champ de vision d'attention (une zone réduite par rapport à tout le champ de vision humain, et représenté par un cône). Les objets visibles par l'homme seront alors, pris en compte comme candidats pour être l'objet d'attention. Pour choisir un objet parmi les candidats, on mesure la proximité de chaque objet au centre du cône du champ de vision.

Pour arriver à un partage d'attention, le robot doit considérer sa propre perception et les objets visibles pour pouvoir trouver une configuration où il puisse voir l'objet que l'homme est en train de regarder, afin de pouvoir obtenir un canal de communication commun entre les deux (l'homme et le robot).

Cette communication implicite peut être renforcé par un geste explicit qui indique l'objet de référence tel qu'une signalisation par un pointage avec les mains. Notre système d'attention partagée est aussi intégré sur la plateforme HRP2 avec l'architecture montré dans l'image 7.5.

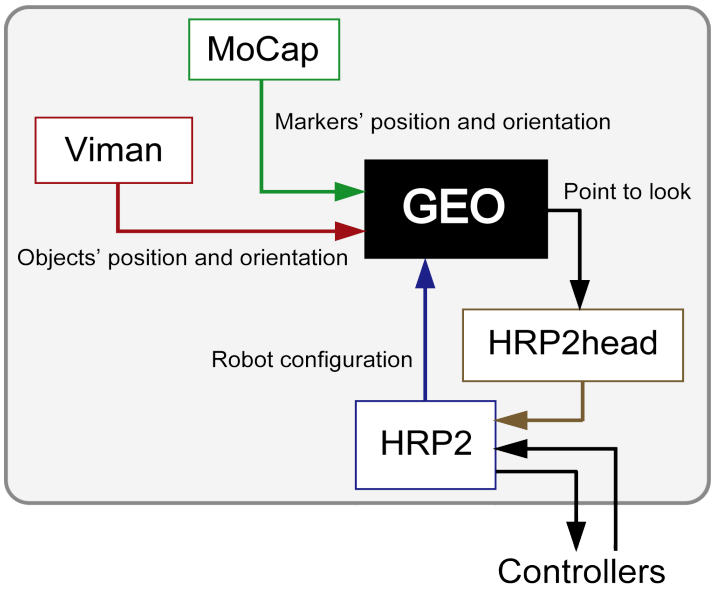


Figure 7.5: L'achitecture du système GEO, lequel reçoit l'information sur la capture du mouvement de l'homme ainsi que sur la position des objets dans l'environnement en utilisant les caméras embarquées .

Les résultats de l'intégration sur le robot sont visibles dans l'image 7.6.

Figure 7.6: Scenario 1: Séquence d'images qui montre comment le robot raisonne sur le point de vue de l'homme et après, il regarde au même objet que la personne en face de lui est en train de regarder.

7.5 Conclusion

Dans ce travail nous avons présenté une façon d'adapter des capacités humaines dans des mécanismes computationnelles. Une fois ces capacités intégrés elles aideront à l'interaction entre l'homme et le robot. On a démontré ces approches, d'un coté, comme support pour la planification de mouvements et d'un autre comme un système pour la compréhension du centre d'attention de l'homme.

Le planificateur de configurations PSP présenté ici, utilise des techniques utilisées par l'humain pour pivoter les objets ou les environnements. Ceci est fait par le calcul de la perspective du robot depuis différentes positions dans l'environnement.

Avec le système PSP le robot est capable de calculer la prise de perspective égocentrique en lui permettant de :

- Avoir une destination pour un planificateur de navigation.
- Avoir une configuration ou un planificateur de manipulation qui peut commencer à planifier différentes tâches (prendre, donner, etc.)
- Avoir une interface bien définie entre les instructions de haut niveau et le planificateur de mouvements.
- Avoir une méthode d'évaluation pour accomplir plusieurs types de tâches.

Avec le système GEO montré dans la dernière partie de ce travail, le robot est capable d'obtenir la perspective de l'homme et de raisonner sur les perspectives de tous les deux, permettant au robot de produire un canal de communication entre lui et l'homme pour arriver à une première étape d'attention partagée entre eux. Ce système nous a permit de connaîttre principalement que:

- Il n'est pas possible d'obtenir une prise de perspective de niveau deux si on ne réalise que le suivi du regard, comme il est fait par la plus part des personnes qui essayent d'achever l'attention partagée entre l'homme et la machine.
- De même pour les étapes 2 et 3 de l'attention partagée, le suivi du regard ne proportionne pas assez d'information pour pouvoir y arriver.
- Les occlussions provoqués par d'autres objets sont très difficiles à les détecter si on raisonne seulement sur l'information d'un espace 2D.

Bibliography

[Abdel-Malek 05] K. Abdel-Malek, Z. Mi, J. Yang & K. Nebel. *Optimization-based layout design*. In ABBI 2005, volume 2, pages 187–196, 2005.

[Ackerman 96] Edith K. Ackerman. *Perspective-Taking and Object Construction: Two Keys to Learning*. Constructionism in Practice: Designing, Thinking, and Learning in a Digital World, 1996. Mahwah, New Yersey.

[Alami 98] R. Alami, R. Chatila, S. Fleury, M. Ghallab & F. Ingrand. *An architecture for autonomy*. International Journal of Robotic Research, vol. 17, pages 315–337, 1998.

[Alami 06] Rachid Alami, Raja Chatila, Aurelie Clodic, Sara Fleury, Matthieu Herrb, Vincent Montreuil & Emrah Akin Sisbot. *Towards Human-Aware Cognitive Robots*. The Fifth International Cognitive Robotics Workshop (The AAAI-06 Workshop on Cognitive Robotics), 2006.

[Ali 09] Saif Ali, Jieping Ye, Anshuman Razdan & Peter Wonka. *Compressed Facade Displacement Mapping*. IEEE Transactions on Visualization and Computer Graphics, no. 2, 2009.

[Baba 06] Abedallatif Baba & Raja Chatila. *Simultaneous environment mapping and mobile target tracking*. International Conference on Intelligent Autonomous Systems, 2006.

[Baerlocher 04] P. Baerlocher & R. Boulic. *An inverse kinematics architecture enforcing an arbitrary number of strict priority levels*. The Visual Computer, vol. 20, pages 402–417, 2004.

[Banta 00] Joseph E. Banta, Laurana M. Wong, Cristophe Dumont & Mongi A. Abidi. *The Next-Best-View System for Autonomous 3-D Object Reconstruction*. IEEE Transactions on Systems, Man and Cybernetics, vol. 30, pages 589–598, 2000.

[Baron-Cohen 95a] Simon Baron-Cohen. *The Eye Direction Detector (EDD) and The Shared Attention Mechanisms(SAM): Two cases for evolutionary psychology*. In P. Moore & P. Dunham, editeurs, Joint Attention: Its origins and role in development. Lawrence Erlbaum Associates, 1995.

[Baron-Cohen 95b] Simon Baron-Cohen. Mindblindness: An essay on autism and theory of mind. MIT Press., 1995.

[Berlin 06] Matt Berlin, Jesse Gray, Andrea L. Thomaz & Cynthia Breazeal. *Perspective Taking: An Organizing Principle for Learning in Human-Robot Interaction.* In International Conf. on Artificial Intelligence, AAAI, volume AAAI, 2006. Boston, Mt.

[Bottino 06] Andrea Bottino & Aldo Laurentini. *What's NEXT? An Interactive Next Best View Approach*. Journal of Pattern Recognition, pages 126–132, 2006.

[Breazeal 06] Cynthia Breazeal, Matt Berlin, Andrew Brooks, Jesse Gray & Andrea L. Thomaz. *Using Perspective Taking to Learn from Ambiguous Demonstrations*. Robotics and Autonomous Systems, pages 385–393, 2006.

[Brèthes 05] L. Brèthes, F. Lerasle & P. Danès. *Data fusion for visual tracking dedicated to human-robot interaction*. International Conference on Robotics and Automation, pages 2087–2092, 2005.

[Brooks 04] Andrew G. Brooks, Jesse Gray, Guy Hoffman, Andrea Lockerd, Hans Lee & Cynthia Breazeal. *Robot's play: interactive games with sociable machines*. Computational Entertainment, vol. 2, no. 3, pages 10–10, 2004.

[Broquère 08] Xavier Broquère, Daniel Sidobre & Ignacio Herrera-Aguilar. *Soft motion trajectory planner for service manipulator robot*. In In IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, Nice, France, September 2008.

[Butterworth 95] G. Butterworth. *Origins of mind in perception and Action*. In P. Moore & P. Dunham, editeurs, Joint Attention: Its origins and role in development. Lawrence Erlbaum Associates, 1995.

[Chhugani 05] Jatin Chhugani, Burdirijanto Purnomo, Shankar Krishnan, Jonathan Cohen, Suresh Venkatasubramanian, David Johnson & Kumar Subodh. *vLOD : High-Fidelity Walkthrough of large Virtual environments*. IEEE Transactions on Visualization and Computer Graphics, no. 1, 2005.

[Cogniron 08] Cogniron. *COGNIRON Project RA3 Final Delivrable.* Rapport technique, Rapport Technique LAAS-CNRS, 2008.

[Costella 95] John P. Costella. *A Beginner's Guide to the Human Field of View*. Rapport technique, School of Physics, The University of Melbourne, November 1995.

[Dautenhahn 06] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. Nehaniv, E. Sisbot, R. Alami & T. Simeon. *How may I serve you?, A Robot Companion Approaching a Seated Person in a Helping Context*. Conference on Human-Robot Interaction, 2006.

[Davidson 84] Donald Davidson. Inquiries into truth and interpretation. New York:Clarendon Press, 1984.

[de Sá 98] Antonio Gomes de Sá & Joachim Rix. *Virtual Prototyping - The Integration of Design and Virtual Reality*. Rapport technique, Fraunhofer Institute for Computer Graphics, Darmstadt, Germany, 1998.

[Faust 06] Josh Faust, Cheryl Simon & William D. Smart. *A Video Game-based Mobile Robot Simulation Environment*. In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Beijing, China., 2006.

[Fedrizzi 09] Andreas Fedrizzi, Lorenz Moesenlechner, Freek Stulp & Michael Beetz. *Transformational Planning for Mobile Manipulation based on Action-related Places*. In Proceedings of the International Conference on Advanced Robotics (ICAR)., 2009.

[Feil-Seifer 05] David J. Feil-Seifer & Maja J. Matarić. *A Multi-Modal Approach to Selective Interaction in Assistive Domains*. In IEEE Proceedings of the International Workshop on Robot and Human Interactive Communication, pages 416–421, Nashville, TN, Aug 2005.

[FET 09] FET. *The European Future Technologies Conference: Science Beyond Fiction*, April 2009.

[Flavell 92] John H. Flavell. *Perspectives On Perspective-Taking*. In H. Beilin P.B. Pufall Eds. Piaget's Theory: Prospects and possibilities. The Jean Piaget Symposium series, pages 107–139, Hillsdale, NJ Erlbaum, 1992.

[Fleury 97] S. Fleury, M. Herrb & R. Chatila. *Genom: a Tool for the Specification and the Implementation of Operating Modules in a Distributed Robot Architecture*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Grenoble, FR., 1997.

[Foissotte 09] Torea Foissotte, Olivier Stasse, Adrien Escande, Pierre-Brice Wieber & Abderrahmane Kheddar. *A Two-Steps Next Best View Algorithm for Autonomous 3D Object Modeling by a Humanoid Robot*. In IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.

[Fontmarty 07] Mathias Fontmarty, Thierry Germa, Brice Burger, Luis Felipe Marin & Steffen Knoop. *Implementation of human perception algorithms on a mobile robot*. In Le 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV 2007), Toulouse,France, 2007.

[Frith 01] Uta Frith. *Mind Blindness and the Brain in Autism*. Neuron, Cell Press., vol. 32 Issue 6, pages 969,979, 2001.

[Green 07a] Anders Green. *Characterising Dimensions of Use for Designing Adaptive Dialogues for Human-Robot Communication*. In IEEE RO-MAN, 16th IEEE International Symposium on Robot & Human Interactive Communication, pages 1078–1083, Jeju Island, Korea, August 26-29 2007.

[Green 07b] Anders Green. *The Need for a Model of Contact and Perception to Support Natural Interactivity in Human-Robot Communication*. In IEEE RO-MAN, 16th IEEE International Symposium on Robot & Human Interactive Communication, pages 552–557, Jeju Island, Korea, August 26-29 2007.

[Hall 66] E. T. Hall. The hidden dimension. Doubleday, Garden City, N.Y., 1966.

[Hicheur 05] H. Hicheur, S. Glasauer, S. Vieilledent & A. Berthoz. *Head direction control during active locomotion in humans*. in Wiener,SI, Taube,JS (Eds.), Head Direction Cells and the Neural Mechanisms of Spatial Orientation MIT Press, pages 383–408, 2005.

[Hoeller 07] Frank Hoeller, Dirk Schulz, Mark Moors & Frank E. Schneider. *Accompanying Persons with a mobile robot using motion prediction and probabilistic roadmaps*. In 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29 - November 2, 2007, Sheraton Hotel and Marina, San Diego, California, USA, pages 1260–1265, 2007.

[Hoffman 04] G. Hoffman & C. Breazeal. *Collaboration in Human-Robot Teams*. In 1st AIAA Intelligent Systems Conference, Chicago, IL, USA, September 2004.

[Hsu 06] S. W. Hsu & T. Y. Li. *Third-Person Interactive Control of Humanoid with Real-Time Motion Planning Algorithm*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Beijing, China., 2006.

[Huang 08] Han-Pang Huang, Jia-Hong Chen & Hung-Jing Jian. *Development of the joint attention with a new face tracking method for multiple people*. In In IEEE International Workshop Advanced robotics and Its Social Impacts, Taipei, Taiwan, 2008.

[Huettenrauch 06] Helge Huettenrauch, Kerstin Sevirson Eklundth, Anders Green & Elin A. Topp. *Investigating Spatial Relationships in Human-robot Interaction*. In Proc in (IEEE/RSJ) International Conference on Intelligent Robots and Systems, Beijin, China, 2006.

[Ido 06] Junichi Ido, Yoshio Matsumoto, Tsukasa Ogasawara & Ryuichi Nisimura. *Humanoid with Interaction Ability Using Vision and Speech Information*. In Proc. in (IEEE/RSJ) International Conference on Intelligent Robots and Systems, pages 1316–1321. IEEE, 2006.

[Imai 03] M. Imai, T. Ono & H. Ishiguro. *Physical relation and expression: joint attention for human-robot interaction*. IEEE Transactions on Industrial Electronics, vol. 50, no. 4, Aug 2003.

[Isard 98] M. Isard & A. Blake. *CONDENSATION - Conditional Density Propagation For Visual Tracking*. Int. Journal on Computer Vision (*IJCV'98), vol. 29, no. 1, pages 5–28, 1998.*

[Johnson 05] *M. Johnson & Y. Demiris.* Perceptual Perspective Taking and Action Recognition. *International Journal of Advanced Robotic Systems, vol. 2, no. 4, pages 301–308, 2005.*

[Kanda 07] *T. Kanda, M. Kamasima, M. Imai, T. Ono, D. Sakamoto, H. Ishiguro & Y. Anzai.* A Humanoid Robot that Pretends to Listen to Route Guidance from a Human. *Autonomous Robots, vol. 22(1), pages 87–100, 2007.*

[Kaplan 06] *F. Kaplan & V.V. Hafner.* The challenges of Joint Attention. *The challenges of joint attention, Interaction Studies, vol. 7 Issue 2, pages 135 – 169, 2006.*

[Katayama 03] *M. Katayama & H. Hasuura.* Optimization principle determines human arm postures and "comfort". *In SICE 2003 Annual Conference, volume 1, pages 1000–1005, 2003.*

[Kendon 90] *A. Kendon. Conducting interaction - pattens of behavior in focused encounters. Cambridge University Press., 1990.*

[Kleinehagenbrock 02] *M. Kleinehagenbrock, S. Lang, J. Fritsch, F. Lomker, G. A. Fink & G. Sagerer.* Person Tracking with a Mobile Robot based onMulti-Modal Anchoring. *In Int. Workshop on Robot and Human Interactive Communication, Berlin, Germany, 2002.*

[Lambrey 08] *S. Lambrey, M.-A. Amorim, S. Samson, M. Noulhiane, D. Hasboun, S. Dupont, M. Baulac & A. Berthoz.* Distinct Visual Perspective-Taking Strategies Involve the Left and Right Medial Temporal Lobe differently. *Brain Journal on Neurobiology, Oxford University Press, vol. 131, pages 523–534, 2008.*

[Latombe 91] *Jean-Claude Latombe. Robot motion planning. Kluwer Academic Publishers, Norwell, MA, USA, 1991.*

[Laumond 97] *Jean-Paul Laumond. Robot motion planning and control. Telos, Springer Verlag, 1997.*

[Lavalle 06] *Steve Lavalle. Planning algorithms. Cambridge University Press, 2006.*

[Lee 01] *P. U. Lee & B. Tversky.* Costs of Switching Perspectives in Route and Survey Description. *In Proceedings of the twenty-third Annual Conference of the Cognitive Science Society, Edinburgh, Scotland., 2001.*

[Lee 02] *Joo-Ho Lee, Kazuyuki Morioka & Hideki Hashimoto.* Physical Distance based Human Robot Interaction in Intelligent Environments. *In IEEE 2002 28th Annual Conference of the Industrial Electronics Society, Sanya, China, November 2002.*

[Li 05] *Y. F. Li, B. He & Paul Bao.* Automatic View Planning with Self-termination in 3D Object Reconstructions. *Journ. on Sensors and Actuators, pages 335–344, 2005.*

[Low 03] *Setha M. Low & Denise Lawrence-Zuniga. The antropology of space and place: Locating culture. Blackwell Publishing, 2003.*

[Low 06] *Kok-Lim Low & Anselmo Lastra.* An Adaptative Hierarchical Next-Best-View Algorithm. *In 14th Pacific Conference on Computer Graphics and Applications, pages 589–598, Taipei, Taiwan, October 2006.*

[Marler 05] *R. T. Marler, J. Yang, J. S. Arora & K. Abdel-Malek.* Study of Bi-Criterion Upper Body Posture Prediction using ParetoOptimal Sets. *In IASTED International Conference on Modeling, Simulation and Optimization, Oranjestad, Aruba, Canada., 2005.*

[Menchaca-Brandan 07] *M. Alejandra Menchaca-Brandan, Andrew M. Liu, Charles M. Oman & Alan Natapoff.* Influence of Perspective-taking and Mental Rotation Abilities in Space Teleoperation. *In HRI '07:Proceeding of the ACM/IEEE international conference on Human-robot interaction, Arlington, Virginia, USA., 2007.*

[Menezes 05] *P. Menezes, F. Lerasle, J. Dias & R. Chatila.* A Single Camera Motion Capture System dedicated to Gestures Imitation. *In Int. Conf. on Humanoid Robots (HUMANOID'05), pages 430–435, Tsukuba, 2005.*

[Mitsunaga 06] *Noriaki Mitsunaga, Takahiro Miyashita, Hiroshi Ishiguro, Kiyoshi Kogure & Norihiro Hagita.* Robovie-IV: A Communication Robot Interacting with People Daily in an Office. *In Proc. in (IEEE/RSJ) International Conference on Intelligent Robots and Systems, pages 5066–5072. IEEE, 2006.*

[Mizuhara 99] *Hiroaki Mizuhara, Jing-Long Wu & Yoshikazu Nishikawa.* The degree of human visual attention in the visual search. *In Fourth International Symposium on Artificial Life and Robotics, Oita, Japan, January 19-22 1999.*

[Moll 06] *Henrike Moll & Michael Tomasello.* Level 1 perspective-taking at 24 months of age. *British Journal of Developmental Psychology, vol. 24, pages 603–613, 2006.*

[Mueller 06] *Pascal Mueller, Peter Wonka, Simon Haegler, Andreas Ulmer & Luc Van Gool.* Procedural Modeling of Buildings. *ACM Transactions on Graphics, no. 3, pages 614–623, 2006.*

[Müller 05] *Notger G. Müller, Maas Mollenhauer, Alexander Rösler & Andreas Kleinschmidt.* The attentional field has a Mexican hat distribution. *Vision Research, vol. 45, no. 9, pages 1129 – 1137, 2005.*

[Nagai 03] *Yukie Nagai, Koh Hosoda & Minoru Asada.* How does an infant acquire the ability of joint attention?: A Constructive Approach, *2003.*

[Nakauchi 02] *Yasushi Nakauchi & Reid Simmons.* A social robot that stands in line. *Autonomous Robots, vol. 12, no. 3, pages 323–324, 2002.*

[Nakayasu 07] *H. Nakayasu, Y. Seya, T. Miyoshi & N. Keren.* Measurement of Visual Attention and Useful Field of View during Driving Tasks Using a Driving Simulator. *In The 2007 Mid-Continent Transportation Research Symposium, Iowa, USA., August 2007.*

[Null 06] *Bradley D. Null & Eric D. Sinzinger.* Next Best View Algorithms for Interior and Exterior Model Acquisition. *LNCS, Advances in Visual Computing, vol. 4292/2006, pages 668–677, 2006.*

[Ogata 09] *Tetsuya Ogata, Ryonosuke Yokoya, Jun Tani, Kazunoris Komatani & Hiroshi G. Okuno.* Prediction and Imitation of other's motions by reusing own forward-inverse model in robots. *In IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.*

[Okamoto 05] *Masashi Okamoto, Yukiko I. Nakano, Kazunori Okamoto, Ken'ichi Matsumura & Toyoaki Nishida.* Producing Effective Shot Transitions in CG contents Based on a Cognitive Model of User Involvement. *in Special Section of Life-like Agent and its Communication, IEICE Transactions of Information and Systems, no. 11, pages 2523–2532, 2005.*

[Pacchierotti 05] *E. Pacchierotti, H. Christensen & P. Jensfelt.* Embodied social interaction for service robots in hallway environments. *Field and Service Robotics, pages 476–487, 2005.*

[Perez 08] *Julien Perez, Cécile Germain-Renaud, Bálizs Kégl & Charles Loomis.* Utility-Based Reinforcement Learning for Reactive Grids. *Autonomic Computing, International Conference on, pages 205–206, 2008.*

[Peters 08] *C. Peters, S. Asteriadis, K. Karpouzis & E. de Sevin.* Towards a Real-time Gaze-based Shared Attention for a Virtual Agent. *In Workshop in Affective Interaction in Natural Environments, AFFINE, Satellite Workshop of the ACM International Conference on Multimodal Interfaces (ICMI), Crete, Grece, October 2008.*

[Pfeiffer 08] *Thies Pfeiffer, Marc E. Latoschik & Ipke Wachsmuth.* Conversational Pointing Gestures for Virtual Reality Interaction: Implications from an Empirical Study. *In IEEE Virtual Reality 2008 Conference, Reno, Nevada, USA., 2008.*

[Piaget 52] *Jean Piaget. The origins of intelligence on children. Oxford International Universities Press., 1952.*

[Piaget 56] *J. Piaget & B. Inhelder. The child's conception of space. London Routledge, 1956.*

[Pito 99] *Richard Pito.* A Solution to the Next Best View Problem for Automated Surface Acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 10, pages 1017–1030, 1999.*

[Richarz 06] *Richarz, J. Martin, C. Scheidig & H. M. A. Gross.* There You Go! - Estimating Pointing Gestures In Monocular Images For Mobile Robot Instruction. *In International Symposium on Robot and Human Interactive Communication ROMAN, Univ. of Hertfordshire, Hatfield, UK, September 2006.*

[Rix 99] *Joachim Rix & André Stork.* Combining ergonomic and field-of-view analysis using virtual humans. *In SME Computer Technology Solutions Conference, Detroit, 1999.*

[Sanchiz 99] *J.M. Sanchiz & R.B.Fisher.* A Next-Best-View Algorithm for 3D Scene Recovery with 5 Degrees of Freedom. *In in Proc. 10th British Machine Vision Conference, University of Nottingham, 1999.*

[Satake 09] *Satoru Satake, Takayuki Kanda, Dylan F. Glas, Michita Imai, Hiroshi Ishiguro & 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, HRI 2009, La Jolla, California, USA, March 9-13, 2009, pages 109–116, 2009.*

[Scassellati 99] *Brian Scassellati.* Imitation and Mechanisms of Joint Attention: A Developmental Structure for Building Social Skills on a Humanoid Robot. *in C. Nehaniv, ed., Computation for Metaphors, Analogy and Agents. LNAI Springer-Verlag, vol. 1552, 1999.*

[Schmalstieg 97] *Dieter Schmalstieg.* A Survey of Advanced Interactive 3-D Graphics Techniques, *1997.*

[Shepard 71] *R. N. Shepard & J. Metzler.* Mental Rotation of three dimentional objects. *Science, vol. 171, pages 701–703, 1971.*

[Shepard 96] *R. N. Shepard & L. A. Cooper. Mental images and their transformations. MIT Press., 1996.*

[Shulz 01] *D. Shulz, W. Burgard, D. Fox & A.B. Cremers.* Tracking multimple moving objects with a mobile robot. *In Proc. of the IEEE Computer Society Conference on computer vision and pattern recognition (CVPR), Kauai,HW, 2001.*

[Sidner 04] *Candace L. Sidner, Cory D. Kidd, Christopher Lee & Neal Lesh.* Where to look: a study of human-robot engagement. *In Proceedings of the 2004 International Conference on Intelligent User Interfaces, pages 78–84, Funchal, Madeira, Portugal, January 13-16 2004.*

[Sidner 08] *Candace L. Sidner.* Humanoid Agents as Hosts, Advisors, Companions, and Jesters. *In Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference, pages 11–15, Coconut Grove, Florida, USA, May 15-17 2008.*

[Siméon 01] *T. Siméon, JP. Laumond & F. Lamiraux.* Move3D: a Generic Platform for Motion Planning. *In 4th International Symposium on Assembly and Task Planning, Japan, 2001.*

[Sisbot 08] *Emrah Akin Sisbot.* Towards Human Aware Robot Motions. *PhD thesis, Universite de Toulouse (Toulouse III), October 2008.*

[Staudte 09] *Maria Staudte & Matthew W. Crocker.* Visual attention in spoken human-robot interaction. *In HRI '09: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, pages 77–84, New York, NY, USA, 2009. ACM.*

[Sumioka 07] *H. Sumioka, M. Asada & Y. Yoshikawa.* Causality detected by transfer entropy leads acquisition of joint attention. *In IEEE 6th International Conference on Development and Learning, London, England, 2007.*

[Sutton 98] *Richard S. Sutton & Andrew G. Barto. Reinforcement learning: An introduction. A Bradford Book, MIT Press, Cambridge, MA, 1998.*

[Svenstrup 09] *Mikael Svenstrup, Soren Tranberg, Hans Jorgen Andersen & Thomas Bak.* Pose Estimation and Adaptative Robot Behaviour for Human-Robot interaction. *In IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009.*

[Takemura 07] *Hiroshi Takemura, Keita Ito & Hiroshi Mizoguchi.* Person Following Mobile Robot under Varying Illumination Based on Distance and Color Information. *In Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics, Sanya, China, December 2007.*

[Taylor 96] *Holly A. Taylor & Barbara Tversky.* Perspective in Spatial Descriptions. *Journal of Memory and Language, vol. 35, pages 371–391, 1996.*

[Thayananthan 03] *A. Thayananthan, B. Stenger, P.H.S. Torr & R. Cipolla.* Learning a Kinematic Prior for Tree-Based Filtering. *In British Machine Vision Conf. (BMVC'03), volume 2, pages 589–598, Norwick, 2003.*

[Tolani 00] *D. Tolani, A. Goswami & N. Badler.* Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models and Image Processing, vol. 62 Issue 5, pages 353–388, 2000.*

[Tomasello 95] *Michael Tomasello.* Joint attention as social cognition. *In P. Moore & P. Dunham, editeurs, Joint Attention: Its origins and role in development. Lawrence Erlbaum Associates, 1995.*

[Tomasello 99] *Michael Tomasello. The cultural origins of human cognition. Harvard University Press., 1999.*

[Tomasello 08] *Michael Tomasello. Origins of human communication. MIT Press., 2008.*

[Trafton 05a] *J. Gregory Trafton, Nicholas L. Cassimatis, Magdalena D. Bugajska, Derek P. Brock, Farilee Mintz & Alan C. Schultz.* Enabling Effective Human-robot Interaction Using Perspective-taking in Robots. *IEEE Transactions on Systems, Man, and Cybernetics, vol. Part A, pages 460–470, 2005.*

[Trafton 05b] *J.G. Trafton, A. C. Schultz, M. Bugajska & F. Mintz.* Perspective-taking with Robots: Experiments and Models. *In Robot and Human Interactive Communication ROMAN, pages 580 – 584., 2005.*

[Turk 91] *M.A. Turk & A.P. Pentland.* Face Recognition using Eigenfaces. *In Int. Conf. on Computer Vision and Pattern Recognition (CVPR'91), pages 586–591, 1991.*

[Tversky 99] *B. Tversky, P.Lee, & S. Main Waring.* Why Do Speakers Mix Perspectives. *Spatial Cogn. Computat., vol. 1, pages 312–399, 1999.*

[Viola 01] *P. Viola & M. Jones.* Rapid Object Detection using a Boosted Cascade of Simple Features. *In Int. Conf. on Computer Vision and Pattern Recognition (CVPR'01), 2001.*

[Walters 05] *M.L. Walters, K. Dautenhahn, R. te Boekhorst, K. L. Koay, C. Kaouri, S. Woods, C. Nehaniv, D. Lee & I. Werry.* The Influence of Subjects Personality Traits on Personal Spatial Zones in a Human-Robot Interaction Experiment. *IEEE International Symposium on Robot and Human Interactive Communication, 2005.*

[Wang 06] *Pengpeng Wang & Kamal Gupta.* A Configuration Space View of View Planning. *In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Beijin, China, 2006.*

[Wang 08] *Jianwei Wang, T. Korhonen & Yuping Zhao.* Weighted Network utility Maximization Aided by Combined Queueing Priority in OFDMA Systems. *In Communications, 2008. ICC '08. IEEE International Conference on, pages 3323–3327, May 2008.*

[Warreyn 05] *Petra Warreyn, Hebert Roeyers, Tine Oelbrandt & Isabel De Groote.* What are you looking at? Joint Attention and Visual Perspective Taking in Young children with Autism Spectrum Disorder. *Journal of Developmental and Physical Disabilities, vol. 17, no. 1, pages 55–73, 2005.*

[Wong 99] *L. M. Wong, C. Dumont & M. A. Abidi.* Next Best View System in a 3-D Object Modeling Task. *In In Proc. International Symposium on Computational Intelligence in Robotics and Automation CIRA, pages 306–311, 1999.*

[Xavier 05] *Joao Xavier, Marco Pacheco, Daniel Castro & Antonio Ruano.* Last line, arc/circle and leg detection from laser scan data in a Player driver. *In In IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005.*

[Yamane 03] *K. Yamane & Y. Nakamura.* Natural motion animation through constraining and deconstraining at will. *IEEE Transactions on Visualization and Computer Graphics, vol. 9 Issue 3, pages 352 – 360, 2003.*

[Yamaoka 08] *Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro & Norihiro Hagita.* How Close? Model of proximity control fo Information-presenting Robots. *In Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, Amsterdam, the Netherlands, 2008.*

[Yamaoka 09] *Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro & Norihiro Hagita.* Developing a model of robot behavior to identify and appropriately respond to implicit attention-shifting. *In Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, pages 133–140, La Jolla, California, USA., 2009.*

[Yin 09] *Xuetao Yin, Peter Wonka & Anshuman Razdan.* Generating 3D Building Models from Architectural Drawings: A Survery. *IEEE Transactions on Visualization and Computer Graphics, no. 2, 2009.*

[Yoda 97] *M. Yoda & Y. Shiota.* The Mobile Robot Which Passes a Man. *In Proceedings of The IEEE International Workshop on Robot and Human Comminication (ROMAN 97), 1997.*

[Yoshimi 06] *Takashi Yoshimi, Manabu Nishiyama, Takafumi Sonoura, Hideichi Nakamoto, Seiji Tokura, Hirokazu Sato, Fumio Ozaki, Nobuto Matsuhira & Hiroshi Mizoguchi.* Development of a Person Following Robot with Vision Based Target Detection. *In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006, October 9-15, 2006, Beijing, China, 2006.*

[Zacharias 07] *Franziska Zacharias, Christoph Borst & Gerd Hirzinger.* Capturing robot workspace structure: representing robot capabilities. *In IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29 - November 2, 2007, Sheraton Hotel and Marina, San Diego, California, USA, pages 3229–3236, 2007.*

[Zacks 01] *Jeffrey M. Zacks, Jon Mires, Barbara Tversky & Eliot Hazeltine.* Mental spatial transformations of objects and perspective. *Spatial Cognition and Computation, vol. 2, no. 4, pages 315–332, 2001.*

[Zhang 97] *Hansong Zhang, Dinesh Manocha, Tom Hudson & Kenny Hoff.* Visibility Culling Using Hierarchical Occlusion Map. *In Proceedings of SIGGRAPH, 1997.*