



HAL
open science

Situation Assessment for Human-Robot Interactive Object Manipulation

Emrah Akin Sisbot, Raquel Ros, Rachid Alami

► **To cite this version:**

Emrah Akin Sisbot, Raquel Ros, Rachid Alami. Situation Assessment for Human-Robot Interactive Object Manipulation. IEEE RO-MAN, Jul 2001, Atlanta, United States. hal-01977510

HAL Id: hal-01977510

<https://laas.hal.science/hal-01977510>

Submitted on 10 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Situation Assessment for Human-Robot Interactive Object Manipulation

E. Akin Sisbot, Raquel Ros and Rachid Alami
LAAS-CNRS

7 avenue du Colonel Roche
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS
F-31077 Toulouse, France

{sisbot, rrosespi, alami}@laas.fr

Abstract—In daily human interactions spatial reasoning occupies an important place. With this ability we can build relations between objects and people, and we can predict the capabilities and the knowledge of the people around us. An interactive robot is also expected to have these abilities in order to establish an efficient and natural interaction.

In this paper we present a situation assessment reasoner, based on spatial reasoning and perspective taking, which generates on-line relations between objects and agents in the environment. Being fully integrated to a complete architecture, this reasoner sends the generated symbolic knowledge to a fact data base which is built on the basis on an ontology and which is accessible to the entire system. This work is also part of a broader effort to develop a complete decisional framework for human-robot interactive task achievement.

I. INTRODUCTION

With the recent advances in robotics, robots have begun to appear in our daily lives. From intelligent cars to robotics vacuum cleaners, they are now sharing the environment with humans. The world of a robot, which was once fixed and structured, is now very dynamic and populated with humans with whom to interact.

An important challenge for researchers is to adapt the robot's reasoning capabilities to this new world, which is by default shaped for humans. In general, the internal representation of a robot is quite different from the one used by humans. For instance, while a robot perceives a scene in terms of numbers and coordinates, a person observes the same scene through symbolic locations and relations between objects. As a consequence, the way humans communicate will be shaped by this abstraction. When a person describes something, she uses abstract descriptions, e.g. "the book on the table", and abstract capabilities, e.g. "the restaurant can be seen from the station". Thus, a robot who will interact with humans should be able to extract, compute or infer these relations and capabilities in order to communicate efficiently in a natural way.

In this paper we present a situation assessment reasoner that generates relevant symbolic information from the geometry of the environment with respect to relations between objects and human capabilities. Moreover, the notion of

Perspective Taking [1] is employed in the heart of the system to provide the robot with the ability to put itself at the human's place and to reason about the world from different perspectives.

This work is part of a broader effort to develop a decisional framework for human-robot interactive task achievement, embedded in a cognitive architecture, involving a knowledge base, a task planner, a motion planner and a supervisor. The approach introduced in this work not only facilitates the interaction between robots and humans, but also bridges the gap between high level tasks and low level movement actions.

Section II briefly describes related work. Section III presents the proposed reasoner describing a set of relations for HRI. Section IV presents the integration of the reasoner into a complete architecture. Finally, Section V illustrates examples of the approach and Section VI concludes the paper.

II. PERSPECTIVE TAKING AND SPATIAL REASONING

Perspective Taking is a human ability which allows one to put him/herself in another person's point of view. Studied deeply in psychology literature [1], [2], this ability is crucial when interacting with people by allowing one to reason on others' understanding of the world in terms of visual perception ("what is visible for me, might not be visible for you"), spatial descriptions ("this is on my left from my view, but it's on you right from your view"), abilities ("I can reach this, but you can't"), beliefs, etc.

Therefore, in the last years these notions have been gradually employed in Human-Robot Interaction. [3] present a learning algorithm that takes into account information about a teacher's visual perspective in order to learn a task. [4] apply visual perspective taking for action recognition between two robots. [5] use both visual and spatial perspective taking for finding out the referent indicated by a human partner.

Spatial reasoning [6], on the other hand, has been used for natural language processing for applications such as direction recognition [7], [8] or language grounding [9]. [10] presented a spatial reasoner integrated in a robot which computes symbolic positions of objects by using sonars. [11] present a formal computation of the *on* relation and validate the system through stereo vision based object acquisition.

This research was supported by a Marie Curie Intra European Fellowship and the European Communitys Information and Communication Technologies within the 7th European Community Framework Programme under grant agreements no. [220368], ARBI; no. [215805] CHRIS and with ANR Psirob AMORCES project.

[12] propose the use of a dynamic field approach to reason on the spatial relations occurred within a dialog.

III. SITUATION ASSESSMENT

In order to acquire a correct symbolic knowledge of the world, the robot should extract useful knowledge from the environment. In this work we are interested in extracting meaningful relations between entities which are involved in common and simple scenarios where a human interacts with a robot. More precisely, we focus on fundamental capabilities of agents, and spatial relations between objects and agents.

A. Capabilities

There are a number of common properties for a robot and a human related to their capabilities in a given situation: they can both reach, grasp, look, point, etc. In our context, we group robots and humans into a single category. Thus, we define agents as entities that can act in the environment and manipulate it. In this work we focus on the following capabilities from each agents' perspective¹.

1) *Sees*: An important ability to know about an agent is to predict “what it can see”, i.e. what is within its field of view (FOV). A robot being able to compute this information can then act accordingly. A typical example would be a clarification scenario where the human is searching for an object and the robot is able to infer that she is looking for the one that is not visible (otherwise the user would not be searching for it).

The *sees* relation is computed through a model-based approach within the 3D world representation. We place a virtual camera on the “eyes” of the agent parametrized with the agent’s FOV². To determine if an agent A sees an object O , the object alone is projected into the image obtained from the virtual camera, $P_{fov}(O)$. A second projection is then obtained by projecting the object within the environment (with occluding objects, if any), $P_{fov}(E)$. The ratio of visible to invisible pixels between the intersection of these two projections and the object projection alone determines if A sees O . Formally:

$$sees_A(O, \delta) = \begin{cases} true & \text{if } \frac{pixels(P_{fov}(E) \cap P_{fov}(O))}{pixels(P_{fov}(O))} > \delta \\ false & \text{otherwise} \end{cases} \quad (1)$$

where the function *pixels* computes the number of pixels within an image and δ corresponds to an arbitrary threshold.

In figure 1 the field of view of a person is illustrated with a grey cone (broader one). While he is able to see the two small boxes on the table in front of the him, the big box on his right is out of this FOV, and therefore, he is not able to see it.

¹Note that each of the capabilities described are computed from each agent point of view, and therefore, also stored in different models for further use at the decisional level.

²In the case of a human agent, the FOV corresponds to 180 degrees horizontal and 135 degrees vertical (although the FOV of the human eye is narrower, we model it as 180 degrees to compensate eye movements). For the robot, the FOV corresponds to the specifications of its sensors.

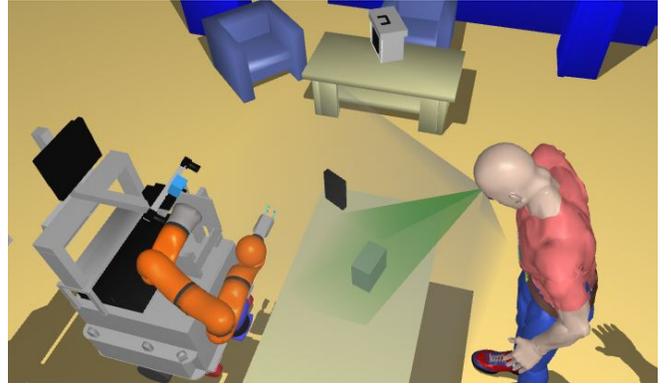


Fig. 1: Representation of the field of view (FOV) and the field of attention (FOA) of the human.

2) *Looks At*: The *looksAt* relation corresponds to what the agent is focused on, i.e., where its focus of attention is directed to. To compute it, we use the same model as the one used for the *sees* relation, but based on a narrower field of view: the field of attention (FOA). For a robot, its FOA is modeled as the overlapping region in the images captured from both cameras. In the case of the human, it corresponds to 30 degrees. The relation *looksAt* is formulated with the following equation:

$$looksAt_A(O, \delta) = \begin{cases} true & \text{if } \frac{pixels(P_{foa}(E) \cap P_{foa}(O))}{pixels(P_{foa}(O))} > \delta \\ false & \text{otherwise} \end{cases} \quad (2)$$

Figure 1 shows the field of attention of a person with a green cone (narrower one). In this example only the grey box satisfies the *looksAt* relation.

3) *Points At*: The relation *pointsAt* verifies if an object is pointed at by an agent. This relation is particularly useful during interaction when one of the agents is referring to an object saying “this” or “that” while pointing at it.

We define the Pointing Field (PF) as the region where the finger is most likely to point at modeled as a cone. Thus, the computing method for this relation is similar to the previous ones except that the virtual camera is placed on the agent’s pointing finger. The angular opening for the camera is arbitrarily set to 30 degrees. The relation *pointsAt* is formulated with the following equation:

$$pointsAt_A(O, \delta) = \begin{cases} true & \text{if } \frac{pixels(P_{pf}(E) \cap P_{pf}(O))}{pixels(P_{pf}(O))} > \delta \\ false & \text{otherwise} \end{cases} \quad (3)$$

Figure 2 illustrates two agents, a human and a robot, pointing at the same object. If a big object occludes a smaller one, and an agent is pointing at them, the outcome of the evaluation will result only in one relation, i.e., Agent *pointsAt* BigObject since the small one is not visible to the agent. On the contrary, if the small object is in front of the big one, then both objects will be pointed at by the human, generating an ambiguity which should be solved through higher level reasoning (e.g. context analysis).

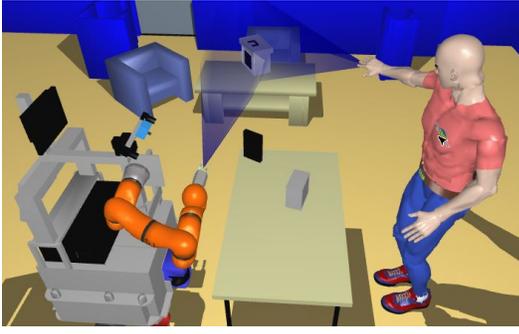


Fig. 2: Both agents, the human and the robot, point at the same object.

4) *Reachable*: Reachability is an important property to compute since it allows the robot to estimate the agents capability to reach an object, which is fundamental for task planning. E.g., if the user asks the robot to give her an object, the robot must compute a transfer point where the user is able to get the object.

Instead of relying on a simplistic agent-object distance function, or computing costly full grasp paths, we propose an intermediate method to compute reachability. We employ Generalized Inverse Kinematics with pseudo inverse method [13], [14] to find a collision free posture for the agent where its end-effector is at the center of the object within a given tolerance. The evaluation function returns success if at least one end-effector (in the case of humans, we usually have two hands) reaches the object with a collision free posture. If the algorithm finds a posture with collision, the function returns a “false+”, meaning that the object is reachable, but producing a collision. Since the approach corresponds to an estimation based on a direct path to the object, it could be the case that in fact, the agent may reach the object avoiding collisions while adapting its posture (specially human agents, who have a very flexible body). Finally, if the object is too far to reach, then a “false-” is returned. Thus, the reachability relation between an agent A and an object O with tolerance τ , is modeled as:

$$\text{reachable}(O, A, \tau) = \begin{cases} \text{true} & \text{if } (pos_{eeef} - pos_o < \tau) \wedge (IK(pos_{eeef}) \in \mathcal{C}_{free}) \\ \text{false+} & \text{if } (pos_{eeef} - pos_o < \tau) \wedge (IK(pos_{eeef}) \notin \mathcal{C}_{free}) \\ \text{false-} & \text{otherwise} \end{cases} \quad (4)$$

where pos_{eeef} , pos_o correspond to the agent’s end-effector and object position, \mathcal{C}_{free} to a set of collision free postures and $IK()$ to the inverse kinematics solver.

Figure 3 shows different reachability postures for each object on the table. In the example, the bottle and the box are both reachable by the human, but the teddy bear is too far. Instead, from the robot’s perspective, the teddy bear is reachable, while the bottle is not.

B. Locations

As humans, one way of referring to the object’s positions is based on their symbolic descriptors instead of using their

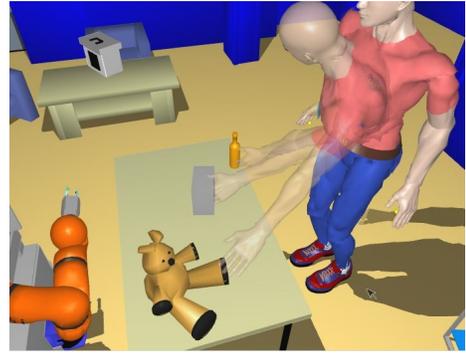


Fig. 3: Different reaching postures for the human: the bottle and the box are reachable, while the teddy bear is not.

precise position. In fact, in many cases, this information is the most precise one that we can give since we do not store the numeric coordinates of objects. The following relations are computed with respect to the position of the agents and the objects.

1) *Location according to an agent*: The relation `isLocatedAt` represents spatial locations between agents and objects. For example we say “it is on my right, on your left,...” We can compute these spatial locations by dividing the space around the referent (an agent) into n regions based on arbitrary angle values relative to the referent orientation. For example, for $n = 4$ we would have the space divided into *front*, *left*, *right* and *back*. Additionally, two proximity values, *near* and *far*, may also be considered. The number of regions and proximity values can be chosen depending on the context where the interaction takes place.

2) *Location according to an object*: We can also refer to object locations with respect to other objects in the environment, such as, above, next to, in, etc. These types of relations are widely studied in language grounding (e.g. [15] presented different models to define the *above* relation). In this work we use similar models based on the bounding boxes and center of mass of the objects to define three main relations (Figure 4):

- *isOn*: computes if an object O_1 is on another object O_2 by evaluating the center of mass of O_1 according to the bounding box of O_2 .

$$\text{isOn}(O_1, O_2) = \begin{cases} \text{true} & \text{if } m_{O_1}^x \in BB_{O_2}^x \wedge \\ & m_{O_1}^y \in BB_{O_2}^y \wedge \\ & m_{O_1}^z > \text{argmax}_z(BB_{O_2}^z) \\ \text{false} & \text{otherwise} \end{cases} \quad (5)$$

where $m_{O_i}^x$ corresponds to the x coordinate of the center of mass of O_i , and $BB_{O_i}^x$ the occupancy of O_i ’s bounding box in the x axis (the same for the rest of axis).

- *isIn*: evaluates if an object O_1 is inside another object O_2 based on their bounding boxes BB_{O_1} and BB_{O_2} :

$$\text{isIn}(O_1, O_2) = \begin{cases} \text{true} & \text{if } BB_{O_1} \subset BB_{O_2} \\ \text{false} & \text{otherwise} \end{cases} \quad (6)$$

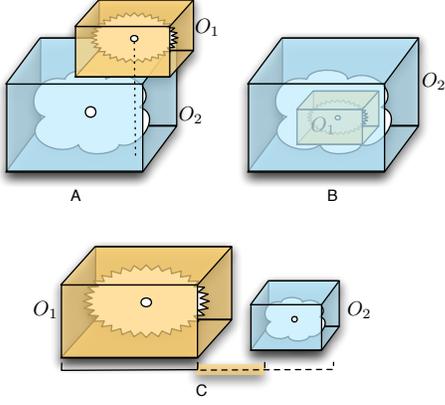


Fig. 4: Spatial relations between two objects: A) isOn relation, B) isIn relation, and C) isNextTo relation.

- *isNextTo*: indicates whether an object O_1 is next to another object O_2 . We cannot use a simple distance threshold to determine if two objects are next to each other since the relation is highly dependent on the dimensions of the objects. For instance, the maximum distance between large objects (e.g. two houses) to consider them as being next to each other is much larger than the maximum distance we would consider for two small objects (e.g. two bottles).

An object O_1 is evaluated as next to object O_2 if part of the bounding box of O_2 , BB_{O_2} , is closer than half of the largest dimension of the bounding box of O_1 , BB_{O_1} . The relationship is formulated as follows:

$$\text{isNextTo}(O_1, O_2) = \begin{cases} \text{true} & \text{if } BB_{O_2} \cap (BB_{O_1} + \max_i(BB_{O_1}^i)) \neq \emptyset \\ & \vee BB_{O_1} \cap (BB_{O_2} + \max_i(BB_{O_2}^i)) \neq \emptyset \\ \text{false} & \text{otherwise} \end{cases} \quad (7)$$

IV. INTEGRATION

The situation assessment system is integrated into Move3D platform [16], which provides with a 3D world representation that we can reason on. The reasoner computes the relations with the 3D object and agent models which are placed in Move3D (figures 2 and 3 illustrate the 3D environment of Move3D).

The system is integrated in Jido and HRP-2 robots. The module is linked to a number of perception modules in order to maintain a consistent representation of the real world and the virtual 3D world of the robot.

The system supports multiple agents (multiple robots and humans) and is sufficiently generic, which allows to transport it to other type of robots.

Three types of input are crucial for the system:

- **Human information.** In order to build a correct model of the human we used different sensors: motion capture (providing the position and orientation of human head and hands), or stereo camera and laser (providing leg, head and hands positions). The different incoming data

is used to build the complete human model by using Generalized Inverse Kinematics.

- **Object information:** objects in the environment are localized and identified with 2D visual tags base on the ARToolkit [17]. The situation assessment module constantly updates the objects' positions' according to the vision data.
- **Robot information:** the actuators provide the configuration of the robot.

The situation assessment module, having a synchronized environment model with the help of the perception data, computes on-line the spatial and perspective relationships. The computation is triggered by a motion of the objects, the human or the robot. All relations except the reachability are computed in 200ms. The reachability relation is computed on demand due to its computationally expensive inverse kinematics operation.

The resulting relations are sent to the symbolic knowledge of our architecture, the OpenRobots Ontology³ [18], for storage, maintenance and further inferences. For example, based on the *isIn* relation the system can infer its inverse relation *contains*. Moreover, we can also define rules such as: if an agent looks at an object and also points at it, then the agent is focused on the object. Thus, the *focusesOn* relation can be automatically inferred at a symbolic level in the ontology.

The ontology server maintains the knowledge produced by the situation assessment module and makes it available to the whole system. Furthermore the relations for each agent are stored separately in order to build a separate cognitive model for each agent. The supervision system as well as the task planner (which are both out of the scope of this paper) use the information generated by the situation assessment reasoner through the ontology.

V. EXAMPLES

Figure 5 illustrates a complete example of the situation assessment reasoner in a real scenario. A person, Achile, and a robot, Jido, are face to face around a table. Objects on the table are marked with tags, and the robot uses its stereo cameras to recognize and localize them. Achile, on the other hand, carries a helmet and a bracelet equipped with markers to be tracked by the motion capture system.

The symbolic information generated by the situation assessment reasoner is described next. At each step new relations are added, and previous ones are updated or removed, or maintained if no changes take place.

A. Figure 5.1

There is a pink box on the table. The box and Achile are detected and placed in the 3D model of the environment⁴. The following relations are computed:

```
PINK_BOX isOn TABLE
```

³The situation assessment system and the ontology are open source and can be accessed through <http://www.openrobots.org>.

⁴In the robot's virtual world, one side of the pink box has been removed to allow the reader to see inside the box.

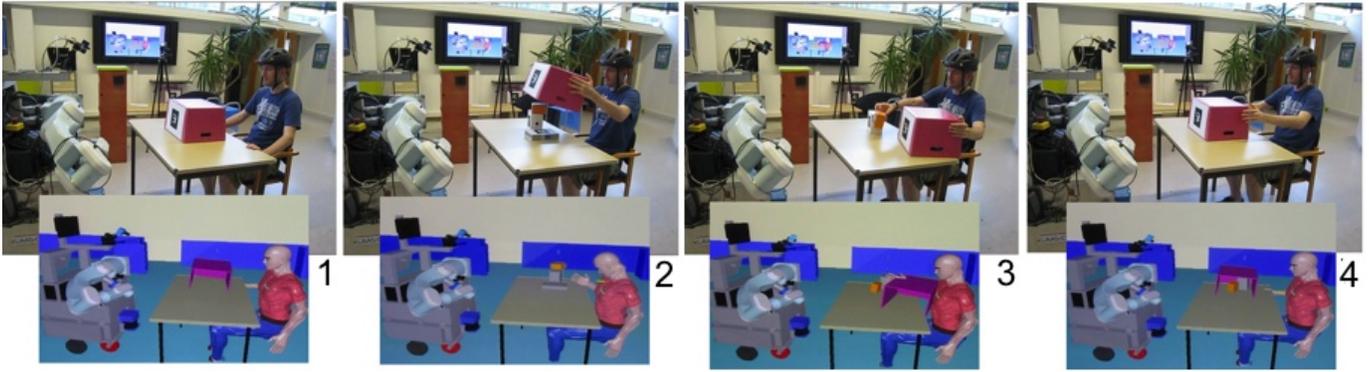


Fig. 5: An example human-robot face to face interaction scenario. Each subfigure shows the real situation on top and the robot’s 3D virtual world at the bottom. The complete video can be found at <http://www.youtube.com/watch?v=7oxjAOZfsBo>.

```
PINK_BOX isLocated in FRONT of ACHILE and JIDO5
PINK_BOX isVisible6 true to ACHILE and JIDO
PINK_BOX isReachable true to ACHILE and JIDO
JIDO looksAt PINK_BOX
```

B. Figure 5.2

Achile lifts the pink box revealing three objects previously hidden. The appearance of these new objects generates:

```
GREY_BOX isOn TABLE
GREY_BOX isLocated in FRONT of ACHILE and JIDO
GREY_BOX isVisible true to ACHILE and JIDO
GREY_BOX isReachable true to ACHILE and JIDO
GREY_BOX isNextTo GREY_TAPE
```

```
GREY_TAPE isOn TABLE
GREY_TAPE isLocated in FRONT of ACHILE and JIDO
GREY_TAPE isVisible true to ACHILE and JIDO
GREY_TAPE isReachable true to ACHILE and JIDO
GREY_TAPE isNextTo GREY_BOX
```

```
ORANGE_BOX isOn GREY_BOX
ORANGE_BOX isLocated in FRONT of ACHILE and JIDO
ORANGE_BOX isVisible true to ACHILE and JIDO
ORANGE_BOX isReachable true to ACHILE and JIDO
ORANGE_BOX isNextTo GREY_BOX
```

```
JIDO looksAt GREY_BOX & GREY_TAPE & ORANGE_BOX
```

```
remove: all relations of the PINK_BOX
```

C. Figure 5.3

Achile moves the orange box and places it next to the grey box. The `isOn` relationship between the orange box and the grey tape is now transformed into a `isNextTo` relation. Since the `PINK_BOX` is detected again by the robot, the relations of fig 5.1 are regenerated in addition to the following facts:

```
remove: ORANGE_BOX isOn GREY_BOX
ORANGE_BOX isNextTo GREY_TAPE
ORANGE_BOX isOn TABLE
```

⁵Identical relations for different agents are grouped to save space.

⁶If agent an *A* sees an object *O* then *O* is visible to *A*. We adopt this predicate to be coherent with the information stored in the knowledge module.

D. Figure 5.4

This time Achile puts the big pink box on top of the objects on the table, thus hiding them all. When the `sees` relation is now tested, the outcome is false since they are now occluded from the agents’ views. However, the relations among the hidden objects are not deleted since the reasoner will maintain their last state until a new one is computed. In other words, “if you cannot see, you cannot tell” and we decided to adopt the assumption that the state remains until otherwise computed or indicated. Thus, the pink box covering the two objects generates the following relations:

```
ORANGE_BOX isIn PINK_BOX
ORANGE_BOX isVisible false to JIDO and ACHILE
ORANGE_BOX isReachable false to JIDO and ACHILE
```

```
GREY_BOX isIn PINK_BOX
GREY_BOX isVisible false to JIDO and ACHILE
GREY_BOX isReachable false to JIDO and ACHILE
```

E. Figure 6

Figure 6 illustrates different situations for the *reachable* relation in another example. In this case, Jido and Achile are placed face to face around a table with a different set of objects (Figure 6.1).

The robot first estimates if the small grey box is reachable to itself. After finding a collision free posture to reach the object (Figure 6.2), the relation *reachable* is satisfied and the following statement is generated:

```
SMALL_GREY_BOX isReachable true to JIDO
```

Next the robot switches to the human’s perspective to estimate if the same object is reachable to Achile as well. Since a collision free reaching posture (Figure 6.3) is found, the following statement is generated:

```
SMALL_GREY_BOX isReachable true to ACHILE
```

In the last scene, Achile moves towards his left, farther from the object (Figure 6.4). The situation is then reevaluated. In this occasion though, the reasoner cannot find a satisfactory posture for the human to reach the box because he is too far from the target. The following relation is updated:

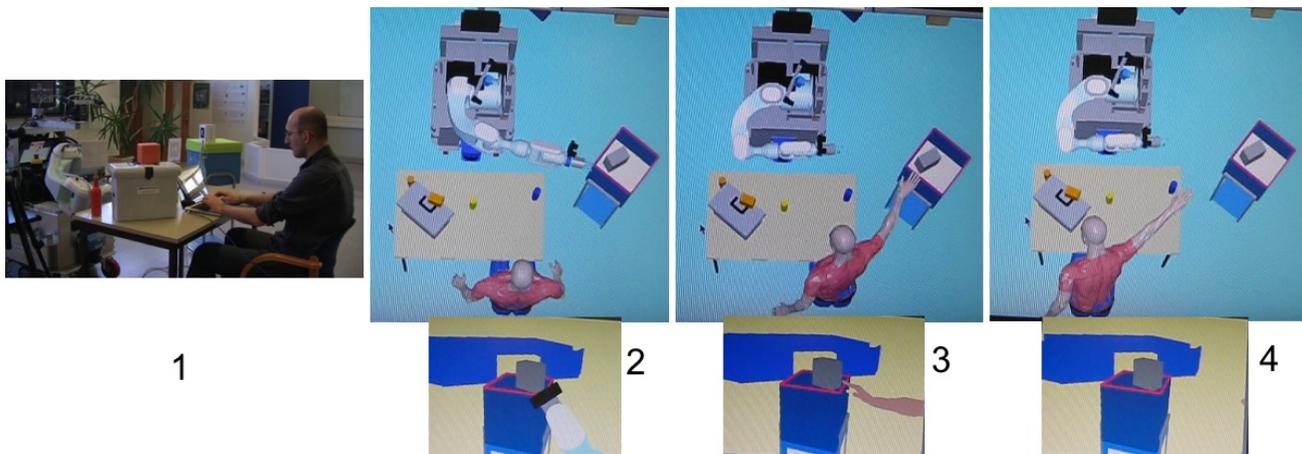


Fig. 6: An example illustrating the *reachable* relation. The relation is computed from the perspectives of both the robot and the agent. The computed posture at each step is illustrated with a global view of the scene (top), and from a closest view (bottom). The complete video can be found at <http://www.youtube.com/watch?v=8ZBoE8HILg>

SMALL_GREY_BOX isReachable false to ACHILE

VI. CONCLUSION

In this paper we have presented a geometric reasoner that generates symbolic relations between objects and agents by using perspective taking and spatial reasoning. The spatial relations have been computed in a 3D model of the environment, which is kept synchronized with the real world. They are then stored, and further inferred in an ontology, where the relations become available for the whole architecture. The situation assessment system is also integrated to our robot, Jido, and runs on several exemplary scenarios.

These relations not only contribute to generating an efficient symbolic representation of the world state, but also to creating a natural dialog and a bridge between symbolic and geometric planning, which we plan to do in the future. We also plan to integrate new relations into the reasoner, such as “graspability”, to compute if an agent can grasp an object, and additional spatial descriptors. Relations involving the field of attention of the human can be fed by a more efficient estimation (e.g. [19]) in order to represent Visual Focus of Attention. Moreover, we will also include dynamic relations, such as “object O is moved by agent A ” which will help to better understand continuous changes in the environment and will allow to predict future states of the world.

REFERENCES

- [1] J. H. Flavell, *Perspectives on Perspective Taking*. Lawrence Erlbaum Associates, 1992, pp. 107–139.
- [2] B. Tversky, P. Lee, and S. Mainwaring, “Why do speakers mix perspectives?” *Spatial Cognition and Computation*, vol. 1, no. 4, pp. 399–412, 1999.
- [3] C. Breazeal, M. Berlin, A. Brooks, J. Gray, and A. L. Thomaz, “Using perspective taking to learn from ambiguous demonstrations,” *Robotics and Autonomous Systems*, pp. 385–393, 2006.
- [4] M. Johnson and Y. Demiris, “Perceptual perspective taking and action recognition,” *Advanced Robotic Systems*, vol. 2, no. 4, pp. 301–308, 2005.
- [5] J. G. Trafton, N. L. Cassimatis, M. D. Bugajska, D. P. Brock, F. Mintz, and A. C. Schultz, “Enabling effective human-robot interaction using perspective-taking in robots,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. Part A, pp. 460–470, 2005.
- [6] J. O’Keefe, *The Spatial Prepositions*. MIT Press, 1999.
- [7] T. Kollar, S. Tellex, D. Roy, and N. Roy, “Toward understanding natural language directions,” in *HRI*, 2010, pp. 259–266.
- [8] C. Matuszek, D. Fox, and K. Koscher, “Following directions using statistical machine translation,” in *Proc. of Int’l Conf. on Human-Robot Interaction*. ACM Press, 2010.
- [9] S. Tellex, “Natural language and spatial reasoning,” Ph.D. dissertation, Massachusetts Institute of Technology, 2010.
- [10] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, “Spatial language for human-robot dialogs,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 2, pp. 154–167, 2004.
- [11] K. Sjo, A. Aydemir, T. Morwald, K. Zhou, and P. Jensfelt, “Mechanical support as a spatial abstraction for mobile robots,” in *Int. Conf. on Intelligent Robots and Systems*, 2010.
- [12] Y. Sandamirskaya, J. Lipinski, I. Iossifidis, and G. Schoner, “Natural human-robot interaction through spatial language: a dynamic neural field approach,” in *RO-MAN*, 2010.
- [13] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Longman Publishing Co., Inc., 1990.
- [14] P. Baerlocher and R. Boulic, “An inverse kinematics architecture enforcing an arbitrary number of strict priority levels,” *The Visual Computer: International Journal of Computer Graphics*, vol. 20, no. 6, pp. 402–417, 2004.
- [15] T. Regier and L. Carlson, “Grounding spatial language in perception: An empirical and computational investigation,” *Journal of Experimental Psychology*, 2001.
- [16] T. Siméon, J.-P. Laumond, and F. Lamiraux, “Move3d: A generic platform for path planning,” in *IEEE International Symposium on Assembly and Task Planning*, Fukuoka, Japan, May 2001, pp. 25–30.
- [17] H. Kato and M. Billinghurst, “Marker tracking and hmd calibration for a video-based augmented reality conferencing system,” in *IWAR ’99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*. Washington, DC, USA: IEEE Computer Society, 1999, p. 85.
- [18] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, and M. Beetz, “Oro, a knowledge management platform for cognitive architectures in robotics,” in *International Conference on Intelligent Robots and Systems*, 2010.
- [19] S. O. Ba and J.-M. Odobez, “Head pose tracking and focus of attention recognition algorithms in meeting rooms,” in *International evaluation conference on Classification of events, activities and relationships*, 2007, pp. 345–357.