Issues of Perceptual Anchoring in Ubiquitous Robotic Systems

Kevin LeBlanc and Alessandro Saffiotti
AASS Mobile Robotics Lab
Örebro University, Örebro, Sweden
{klc,asaffio}@aass.oru.se

***Abstract*— In ubiquitous robotic systems, robots are immersed in an environment containing an abundance of sensors, actuators and smart objects. In such a system, a robot can acquire information about an object from many different sources, possibly including the object itself. While this richness of information opens a new landscape of opportunities, it also adds the fundamental challenge of how to coordinate and integrate all the different types of information which are available. In this paper, we define a general computational framework to address this problem. Our framework is based on an extension of the concept of single-robot perceptual anchoring to the multi-robot case, and it can be applied to any ubiquitous robotics system. To make the framework more tangible, we apply it to a specific type of ubiquitous robotic system, called Ecology of Physically Embedded Intelligent Systems, or PEIS-Ecology. We also describe a sample implementation based on fuzzy logic, and present some illustrative experiments.**

I. INTRODUCTION

Integrating robots and smart environments is an idea which is gaining popularity in the field of autonomous robotics. The proponents of this idea suggest a vision in which a large number of robotic devices embedded in the environment communicate and cooperate in order to perform tasks. Robotic devices may include traditional mobile robots, but also distributed sensors, simple actuators, and home appliances.

This vision has been given different names, including ubiquitous robotic systems [9], network robot systems [15], intelligent spaces [12], sensor-actuator networks [8], and PEIS-Ecology [18]. In this paper, we shall generally refer to this vision as “ubiquitous robotics”. One of the tenants of this vision is that, by exploiting the cooperation between many simple pervasive robotic devices, an ubiquitous robotic system can perform tasks that are beyond the capabilities of current stand-alone robots.

As an example, consider a robot trying to grasp a milk bottle. In an ubiquitous robotic system, this robot would not need to use its camera to determine the properties of the bottle (shape, weight, etc.) which would be needed to compute the grasping parameters — a task which has proven to be elusive during several decades of robotic research. Instead, the bottle itself, enriched with an IC-tag, can hold this information and communicate it to the robot.

Ubiquitous robotic systems add a new dimension to robot-environment interaction. In a traditional robotic system the robot’s interaction with objects in the environment is physically mediated; that is, the properties of the objects are estimated using sensors, and their state can be modified using actuators. In a ubiquitous robotic system, a robot has an additional possibility: it can interact with an object directly, through digital communication. The robot can ask an object for its properties, and it can even ask it to perform an action.

While the ability to interact with objects via direct communication opens a new landscape of opportunities, it also creates a new scientific challenge: how to coordinate and integrate physical and digital interaction with the same object. Consider for instance the situation shown in Figure 1. A robot has seen a green box, which it has internally labeled `box-22`, and it should decide if it can push it or not based on its weight. Unfortunately, the robot does not have any sensor that allows it to estimate the weight of the box. Fortunately, the box happens to be a smart object, which has ID = Peis4, and which holds information about its own (fixed) weight. The robot can therefore ask the box how much it weighs; but to do so it needs to know the object’s ID. How can the robot know that the box that it is seeing is the smart object with ID = Peis4?

The problem above must be addressed in any ubiquitous robotic system which integrates physical perception and digital communication. Although a few approaches have been proposed that address specific instances of this problem [14], [1], to the best of our knowledge no general solution has been proposed. There is, however, a related problem in the case of a stand-alone robot which has been the subject of intense study: the so-called perceptual anchoring problem. Simply put, perceptual anchoring is the problem of creating the right correspondence between the symbolic name of a physical object and the perceptual data relative to that object. Coradeschi and Saffiotti [6] have proposed a general computational framework to deal with this problem.

In this paper, we extend the work on perceptual anchoring to ubiquitous robotic systems. The result is a general frame-

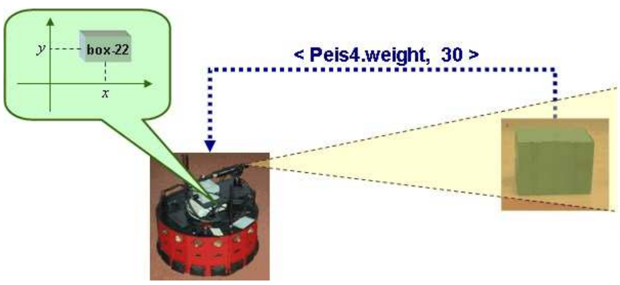

Fig. 1. Combining perceptual and digital information.

ICRA-07 Workshop on Omniscient Space
IEEE Int Conf on Robotics and Automation. Rome, Italy, April 2007

work which can account for situations like the one depicted in Figure 1, in which intrinsically different types of information must be combined. Our framework can be applied to any ubiquitous robotic system, as well as "standard" teams of cooperating robots. In order to make our framework concrete, however, we illustrate it in the context of one specific type of ubiquitous robotic system, called an *Ecology of Physically Embedded Intelligent Systems*, or PEIS-Ecology.

The rest of this paper is organized as follows. In the next two sections, we provide a reminder of the two main ingredients of this paper: PEIS-Ecologies, and perceptual anchoring. In section IV, we describe our generalized framework for perceptual anchoring in a PEIS-Ecology. In section V we describe a sample implementation of this framework. Finally, in section VI we describe two experiments, which illustrate the working of our framework. The first experiment involves information matching, and the second illustrates information fusion.

II. THE PEIS-ECOLOGY APPROACH

The concept of a PEIS-Ecology [18] combines insights from the fields of ambient intelligence and autonomous robotics, generating a radically new approach to the inclusion of robotic technology into everyday environments. In this approach, advanced robotic functionalities are not achieved through the development of extremely advanced robots, but rather through the cooperation of many simple robotic components. The concept of a PEIS-Ecology builds upon the following ingredients (see [2], [16] for more information).

First, any robot in the environment is abstracted by the *uniform notion* of a PEIS (Physically Embedded Intelligent System), which is any device incorporating some computational and communication resources, and possibly able to interact with the environment via sensors and/or actuators. A PEIS can be as simple as a toaster and as complex as a humanoid robot. In general, we define a PEIS to be a set of inter-connected software components residing in one physical entity. Each component may include links to sensors and actuators, as well as input and output ports that connect it to other components in the same or another PEIS.

Second, all PEIS are connected by a *uniform communication model*, which allows the exchange of information among PEIS, and can cope with them joining and leaving the ecology dynamically. This model has been implemented in a specific middleware, the PEIS-kernel, which implements a *distributed tuple space* over a P2P network. This tuple space provides the primary communication mechanism between PEIS, by way of the production and consumption of tuples.

Third, all PEIS can cooperate using a *uniform cooperation model*, based on the notion of linking functional components: each participating PEIS can use functionalities from other PEIS in the ecology in order to compensate or to complement its own. In our middleware this cooperation comes from *subscriptions* to tuples, which allow a consuming PEIS to automatically receive tuples created by a producing PEIS.

We define a PEIS-*Ecology* to be a collection of inter-connected PEIS, all embedded in the same physical environment. We call a set of connections between components within and across the PEIS in the ecology a *configuration* of the PEIS-Ecology. Note that the same ecology can be configured in many different ways depending on the current context. Relevant contextual aspects here include the current goals, situation, and resources. In our middleware a configuration corresponds to a *set of subscriptions* between components.

Examples of actual PEIS-Ecologies can be found at the PEIS-Ecology project home page [16].

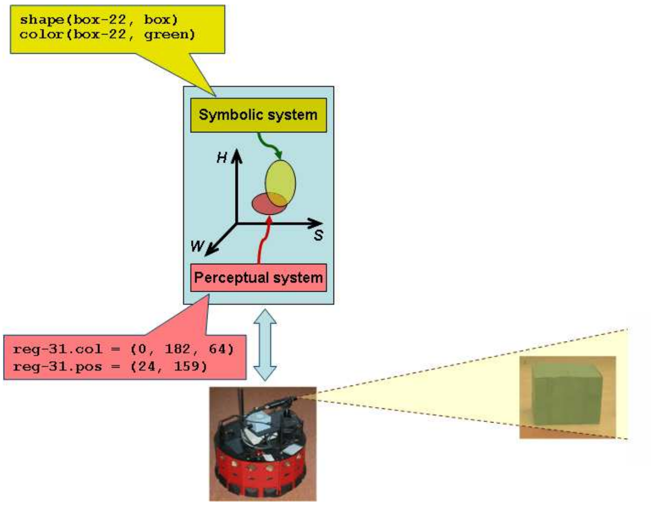

Fig. 2. An example of perceptual anchoring for a single robot.

III. PERCEPTUAL ANCHORING IN A SINGLE ROBOT

Before discussing the problem of anchoring in a PEIS-Ecology, we recall the basic ingredients for perceptual anchoring for the case of a single, non-ubiquitous robot.

The problem of perceptual anchoring, or simply anchoring, has been defined by Coradeschi and Saffiotti [5], [6] as the problem of creating and maintaining the connections, inside an intelligent robot system, between symbols and sensor data that refer to the same physical objects. The variant of the framework used in this paper is inspired by a later work that uses Gardenförs' conceptual spaces [4].

A. A simple example

Figure 2 provides an illustration of this problem. Consider an autonomous robot called Pippi, endowed with a symbolic system (e.g., a task planner) and with a perceptual system (e.g., a vision system). Suppose that the planner generates a plan that includes the action `PUSH(box-22)`, where `box-22` is a symbol used by the planner which denotes a specific green box in the physical world. Suppose then that the vision system perceives an object in front of Pippi. In order to perform the action `PUSH(box-22)`, Pippi first needs to determine if the object seen by the camera is the object which is called `box-22` by the planner. That is, it needs to *anchor* the symbol `box-22` to the sensor data coming from the camera. Once this is done, Pippi can use the data from the camera to control the `PUSH` behavior, being sure that in so doing it is actually pushing the right object.

Anchoring can be performed as illustrated in Figure 2. The robot uses an internal space, called the *anchoring space,* to put together information coming from the symbolic system and from the perceptual system. This is a multi-dimensional space, which includes one dimension for each quality of interest. In our example, the anchoring space includes two dimensions for color information (Hue and Saturation) plus one dimension for Weight. Other dimensions could also be included, e.g., Shape. In order to anchor the symbol `box-22` to the data from the camera we proceed as follows. First, the symbolic information about the properties of `box-22` are mapped into the anchoring space. In our example, the predicate `green` is mapped to an area in the Hue-Saturation plane (the large ellipsoid) that corresponds to the (Hue, Saturation) pairs that can be considered as "green". This mapping can be hand coded, or it can obtained in some other way (e.g., learned). Second, the perceptual data about the observed object in front of the robot is also mapped into the anchoring space. In our example, the (average) color of the corresponding region in the camera image is mapped to the smaller ellipsoid in the Hue-Saturation plane, which represents the mean and standard deviation of the (Hue, Saturation) values of the pixels in that region. Finally, a comparison between the two ellipsoids is made in order to determine whether they are compatible. If they are, then we conclude that the region in the camera image and the symbol `box-22` refer to the same physical object, and we *anchor* them. As a consequence, we can use other information computed from the camera image, e.g. the position of the box, to control the `PUSH(box-22)` behavior.

B. A computational framework

In more general terms, Coradeschi and Saffiotti's framework for perceptual anchoring is defined by the following ingredients (see the above references for a full account).

- A *symbol system*, including a set $\{x_1, x_2, \ldots\}$ of individual symbols (variables and constants) used to denote objects in the world.
- A *perceptual system*, which generates a flow of percepts $\{\pi_1, \pi_2, \ldots\}$ with their corresponding attributes; a percept is a structured collection of measurements assumed to originate from the same physical object.
- The *anchoring space* S, which is a metric space whose dimensions are the qualities of interest in the application domain.
- A pair of *grounding functions*: g^{sym}, which maps predicate symbols into areas of S, and g^{per}, which maps percepts into areas of S. These functions encode the correspondence between symbolic properties and perceived attributes, mapping both into the common anchoring space S.

In the example depicted in Figure 2, the symbol system uses the individual constant `box-22` to denote the box, and it associates it with the predicate `green`. The g^{sym} function maps this predicate to a set of (Hue, Saturation) values in the space S. The perceptual system generates, at any given point in time, a region (percept) in the segmented camera image, together with its position and its average RGB values. This region corresponds to the box observed by Pippi. The g^{per} function maps the average RGB values to a set of (Hue, Saturation) values in the space S.

Given the above formal ingredients, the task of the anchoring process is to create, and maintain in time, the correspondence between individual symbols and percepts which refer to the same physical objects. The anchoring process uses an internal data structure, called an *anchor*, to represent this correspondence for each object. The anchoring process creates and maintains these anchors in time. This is done through three general functionalities, which can be roughly described as follows.[1]

- **Find** Create a new anchor for a symbol x, and find a matching percept from the perceptual system. In our example, Find is used to establish the first association between the symbol `box-22` and image data.
- **Track** Update an existing anchor by incorporating new perceptual information.
- **Reacquire** Re-establish the symbol-percept association for an object which has not been observed for some time.

The anchoring functionalities above rely on three capabilities: *matching* information from different sources in the anchoring space; *fusing* information from these sources into a combined estimate; and *predicting* how the information will evolve with time. The way these capabilities are realized depends on the specific implementation, and on the specific application domain. In the example illustrated in Figure 2, matching is established by simply verifying that the two ellipsoids overlap. In the implementation described in section V below, we use fuzzy logic to perform both matching and fusion. (Prediction is not used in the experiments reported here.)

IV. PERCEPTUAL ANCHORING IN A PEIS-ECOLOGY

When we move from the case of a stand-alone robot to the case of a PEIS-Ecology, the anchoring problem changes in two important ways.

First, the same object can be observed by multiple PEIS. For instance, the box in the example above can be observed both by Pippi and by a static camera on the ceiling. In order for Pippi and the camera to exchange or combine information about the box, we need to make sure that their individual anchoring processes are aligned, e.g., that they agree about the name which should be used to refer to the box. That is, we need to perform *cooperative anchoring*. Multi-robot and multi-sensor observation is a well studied problem (e.g., [7], [19], [3]), but very little attention has been given to the inclusion of symbolic information into this problem (e.g., [20], [11]).

The second and more fundamental change is that in a PEIS-Ecology the observed object is not necessarily a passive object, but it can be a PEIS itself. As such, this object could hold its own *a-priori* information, acquire its own perceptual

[1] See [5] for a more formal definition, and [13] for some extensions.

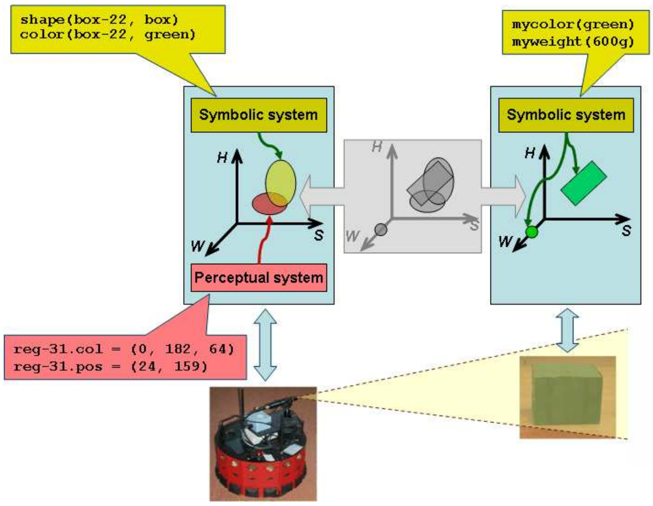

Fig. 3. Perceptual anchoring when the observed object is a PEIS.

information, and communicate this to the observing PEIS. This information could be about itself, or about other entities. In the box example, a PEIS-box might hold information about its own size, shape and friction coefficients; it might also be equipped with sensors which allow it to measure its own weight and temperature; and it could communicate all these data to any other PEIS through the PEIS-middleware.

The fundamentally new aspect brought about by the PEIS-Ecology approach is that the traditional distinction between the robot and the environment becomes blurred. In fact, in a PEIS-Ecology the robot-object relation should be seen, for those objects which are PEIS, as a robot-robot relation, and it should be addressed using the tools of cooperative robotics.

A. A simple example

Figure 3 illustrates the anchoring problem for a case in which an observed box is itself a PEIS. The PEIS-box has its own symbol system, which contains some simple *a-priori* symbolic information about its color and weight. In this case, the box does not have any sensors, hence it does not have its own perceptual system. An example of this situation could be a package of soap, with an RFID tag.

The PEIS-box has its own internal anchoring space. In this example, for simplicity, this space has the same dimensions (Hue, Saturation, and Weight) as Pippi's anchoring space. The anchoring process in this PEIS-Ecology scenario can be informally described as follows. First, we map all the information represented in each individual anchoring space to a common space. In the figure, this space is represented by the gray area between the two PEIS. Notice that this step is trivial in our example, since the anchoring spaces of the two PEIS involved are the same; in the general case, the establishment of a common space and the mapping to this space might be more complex. Second, we verify that all the (symbolic and perceptual) information in the common anchoring space is compatible. In our case, there is a non-empty intersection between the areas in the (Hue,Saturation) plane corresponding to the different items of information, and therefore we conclude that all these items refer to the same object. In particular, this implies that the PEIS-box is the same physical object as the object denoted by the symbol `box-22` inside Pippi. Third, we combine all the available information to get a more accurate and complete estimate of the properties of the box, and we make the result available to all PEIS. As a result of this, Pippi will know the weight of the box, and will thus be able to decide whether it can push it. Finally, the properties of the object should be tracked over time. Tracking is an important part of the anchoring framework, but it is out of the scope of our current discussion.

B. A computational framework

We now present our computational framework for anchoring in a PEIS-Ecology. This framework generalizes the one for stand-alone anchoring, and it can be summarized as follows.

We consider a set $P = \{p_1, \ldots, p_n\}$ of PEIS. For each PEIS p_i in P, we define:

- An *anchoring space* S_i. This is usually a multi-dimensional space.
- A set $X_i = \{x_i^1, \ldots, x_i^k\}$ of *information sources* for p_i. For a typical PEIS, its local symbol system and each sensing modality available to the perceptual system act as information sources.
- For each source x_i^j, we define a *grounding function* g_i^j that maps its information to the anchoring space S_i. If the only two information sources in the PEIS are the symbol system and a single-modality perceptual system, then the only g functions for it are the g_i^{sym} and the g_i^{per} already discussed in the single-robot case.

Given the set P of PEIS, we call *shared anchoring space* the space in which we perform cooperative anchoring among these PEIS. We assume to have, for each PEIS p_i, an invertible function

$$f_i : S_i \to S$$

that maps information for its local space to the shared space. The function must be invertible because we want to map the result of the combination of information, which is in S, back to each individual space S_i in order to be used by each individual PEIS p_i.

The definition of the shared anchoring space S might not be obvious in some case. This space should be at least as rich as any one of the individual S_i's so that the function f_i can be invertible — said differently, we do not lose information in the translation from S_i to S. The functions f_i may involve cylindrical extensions and non-linear transformations.

In the example in Figure 3, Pippi and the PEIS-box have the same local anchoring space, so the shared anchoring space is the same, and their f transformations are both the identity function. Pippi has two information sources, its symbol system and its vision-based perceptual system, while the PEIS-box has its local symbol system as its only information source. The grounding functions are graphically represented in the figure by the arrows that create shapes in the anchoring space.

Once we have mapped all the available information into the shared space S, we can define the anchoring functionalities Find, Track and Reacquire on this space. These functionalities are defined as for the single-robot case, with the only difference that all the operations involved are performed on the shared space S. In particular, matching, fusion and prediction are performed on that space. As in the single-robot case, the details of these operations are not specified by the framework; they can be freely chosen for a given domain. The results of the anchoring functionalities can then be projected back into each individual PEIS.

In practice, a specific implementation of this framework might represent the shared space explicitly in a centralized anchoring service, and implement the anchoring algorithm in that service; alternatively, it might keep the representations and the algorithms distributed among the PEIS.

V. IMPLEMENTATION

We have implemented the framework described previously using fuzzy logic. It is important to note that this is just one possible way to implement the framework; however, the fuzzy logic approach has a number of advantages. One of the main motivations for the use of fuzzy logic is the representational power of fuzzy sets. Fuzzy sets allow various types of information, and in particular various types of uncertainty (e.g. imprecision, vagueness, ambiguity, unreliability, etc.) to be represented effectively. Moreover, there are straightforward and formal mechanisms for fusing and matching information contained in fuzzy sets. Fuzzy logic is also a well-founded theory, which can be implemented in a computationally efficient manner.

We use fuzzy sets to represent many types of uncertain information in our implementation. See [22], [23] for a look at the origins and details of fuzzy sets, and see [17] for an overview of how fuzzy sets are used in autonomous robotics. Simply put, a fuzzy set A over the n-dimensional space S is characterized by a membership function

$$\mu_A : S \to [0, 1]$$

which gives the degree of membership $\mu_A(s)$ of each point s in S to the fuzzy set A. This degree of membership is represented by a real number in the range $[0, 1]$.

A. Anchor Spaces

The individual anchor space S_i of each PEIS p_i is defined by its n dimensions, each of which corresponds to a quality of interest for objects in the domain. Information about a given object is represented in this space by an n-dimensional fuzzy set A, which is implemented as an n-dimensional array of floats. The value in each cell s corresponds to the value of $\mu_A(s)$.

For example, in the first experiment reported below, we represent color information using $n = 3$, to represent the three dimensions of the HSV color space (note that the hue component is wrapped). In the second experiment we describe, we represent 2D position information using a grid of (x, y) values (i.e. $n = 2$) in a given global reference frame. Note that the mechanisms reported here all extend to an arbitrary n.

The shared anchor space S is implemented in the same way as the individual anchor spaces. So each PEIS uses an individual anchor space which has the same dimensionality and units as the shared anchor space. This implies that for each PEIS p_i, the function f_i which maps information from the individual anchor space S_i to the shared anchor space S is the identity function. The trick behind this is that the burden of converting information to suit the shared space is placed on the individual PEIS directly, whenever information is incorporated into its individual anchor space. The alternative would be to have this conversion done in a separate step. This choice does not limit the generality of the framework; it merely simplifies the implementation, by affecting how and when certain computations are made.

B. Information Sources

As was mentioned before, each PEIS p_i may incorporate a number of sources of information. In particular, the local symbol system and the different sensing modalities available to the perceptual system can act as sources of information. For each source x_i^j, the grounding function g_i^j maps items of information provided by that source to fuzzy sets over the individual anchor space S_i.

For instance, consider the scenario illustrated in Figure 3. Let $\mathcal{C}_1$ denote the measured color of an object detected by a camera on PEIS p_1 (Pippi), and let $\mathcal{P}_1$ denote the predicates in Pippi's symbol system which refer to colors. Then, we have two g-functions:

- $g_1^{\mathtt{C}am}$ maps a perceived color $\mathcal{C}_1$ into a fuzzy set on the space S_1; and
- $g_1^{\mathtt{S}ym}$ maps a predicate $\mathcal{P}_1$ representing a color into another fuzzy on the space S_1.

With respect to the figure, $g_1^{\mathtt{C}am}$ maps the color of the box as measured by the camera to the large ellipsoid in the upper part of the (hue,saturation) plane, while $g_1^{\mathtt{S}ym}$ maps the predicate `green` to the smaller ellipsoid.

C. Fusion and Matching

Fusion of information is implemented using fuzzy intersection, which is the standard way of combining information represented by fuzzy sets. If μ_1 and μ_2 are two fuzzy sets on the same space S, representing two distinct items of information, then the result of fusing these two items is represented by the fuzzy set μ given, for any $s \in S$, by

$$\mu(s) = \mu_1(s) \otimes \mu_2(s),$$

where $\otimes$ is a triangular norm, or T-norm [21]. The most common T-norms used in fuzzy logic are the minimum and product operators. In the experiments reported below, we use the product operator.

In the example shown in Figure 3, the result of fusing the three items of information in the shared anchoring space would be represented by the fuzzy set that corresponds to the fuzzy intersection of the three areas shown in the shared space between the two PEIS. Figure 4 shows an example of two 1-dimensional fuzzy sets being fused.

Matching of information is also implemented using the mechanisms of fuzzy logic: if μ_1 and μ_2 represent two items of information as above, then the degree of matching between these two items is given by:

$$\text{match}(\mu_1, \mu_2) = \sup_{s \in S} (\mu_1(s) \otimes \mu_2(s)) ,$$

where $\sup$ denotes the supremum (least upper bound) operator. Other definitions of matching have been proposed in the fuzzy logic literature [10], and other measures of matching could be useful, depending on the context in which the matching is done. For example, some measure of the overlap between fuzzy sets might be used instead. However, the above suffices for the experiments used in this work. In Figure 4, the result of matching two fuzzy sets according to the above equation is 0.8, as shown by the dotted line.

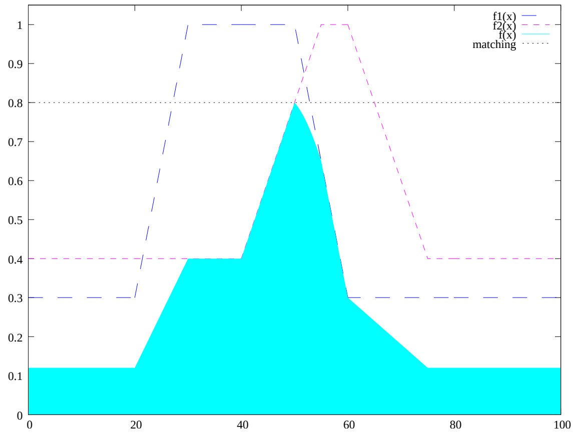

Fig. 4. The fuzzy set f(x) is the result of fusing f1(x) and f2(x) using the product T-norm. The result of the matching is the value of the maximum point of f(x), which is approximately 0.8.

VI. EXPERIMENTS

In order to illustrate the above framework, we present two simple experiments. The first experiment illustrates the *matching* process and is run on-line in a physical PEIS-Ecology. The second experiment illustrates the *fusion* process and is run offline, using artificial data.

The working environment we use for developing and testing PEIS-Ecology in various configurations is a small bachelor apartment (about $25\,\mathrm{m}^2$), which we call the PEIS-Home. The first experiment we will present was performed online in the PEIS-Home, and the second experiment uses a coarse model of the apartment. The PEIS-Home has a living room, a bedroom, and a small kitchen. Figure 5 shows some views of the PEIS-Home.

A. A matching experiment

The purpose of the matching experiment is to demonstrate how information of different types (in this case symbolic and perceptual), and from different sources, can be compared in order to determine which items of information pertain to the same physical object. In particular, this is used to perform perceptual anchoring.

Fig. 5. The living room and kitchen in the PEIS-Home.

The physical experiment was carried out using a PEIS-Ecology containing the following three PEIS.[2]

1) *Pippi*: a Magellan Pro robot equipped with odometry, sonars, and a camera. *Pippi* is the main actor in this scenario.
2) *Emil*: a second Magellan Pro robot. *Emil* is not performing any actions in this scenario, but it is still a member of the PEIS-Ecology, able to communicate information via the tuple-space.
3) *PeisBox*: a box containing a small computer running a single PEIS-component that provides static information about the box and its properties.

The scenario unfolds as follows. Initially, *Pippi* is in the living room, *Emil* is in the kitchen, and the *PeisBox* is in the doorway between the living room and the bedroom. *Pippi* has a scheduled task, which is to wake up the occupant of the PEIS-Home, who is asleep in the bedroom. *Pippi* uses a high level planner to generate a plan for this task; the first sub-task of this plan is to navigate to the bedroom. This is the only part of the plan which is relevant to this discussion. See Figure 6 (top) for a graphical view of the experimental setup.

While navigating to the bedroom, *Pippi* observes the *PeisBox*, and creates a percept for it using visual data. The properties of this percept include its color, and shape. Soon thereafter *Pippi* discovers that there is a problem: the sonar sensors detect that there is an obstruction in the doorway to the bedroom. After the normal obstacle avoidance behaviors fail to find a path through the doorway, an exception is sent to the planner.

Upon receipt of this exception, the planner develops a possible contingency plan which involves pushing the obstruction. In order to meet the pre-requisites of this plan, the planner must determine if the obstruction is indeed pushable. The position of the obstruction as determined by the sonars and the position of the percept created upon observing the *PeisBox* coincide, and so the description of the percept is applied to the obstruction.

Pippi knows that there are many PEIS in the environment. Hoping that one of them matches the description of the

[2]This is actually part of a larger experiment, reported in [2], which included more PEIS and was aimed at verifying the functionality of the PEIS-Ecology middleware.

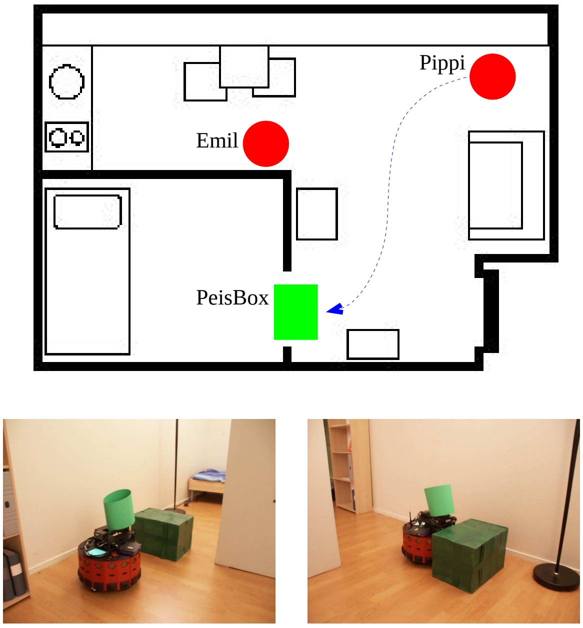

Fig. 6. A visual depiction of the matching experiment. Top: the initial phase of the matching experiment. Bottom: The execution of the box pushing.

object currently blocking the doorway, *Pippi* subscribes to the appearance tuples of all PEIS in the ecology. This is where the matching occurs. The percept for the *PeisBox* includes a color, which is mapped to a region in the HSV color space using a g-function. The shape is also mapped, onto a shape dimension (which in this case is a discrete dimension, with an enumeration of possible shapes).

Emil returns an appearance tuple which includes the color `red` and the shape `cylinder`. The *PeisBox* returns an appearance tuple which describes it as being `green`, and having a `box` shape. All this symbolic information is mapped into the shared anchor space, and the match only succeeds for *PeisBox*. *Emil*'s appearance fails in both the shape dimension, and in the three color dimensions, while the *PeisBox*'s description matches in all dimensions.

The *PeisBox* is thus anchored to the percept corresponding to the observed box. The remaining steps are easy to envision: the planner queries the tuple-space for the property *weight* of the *PeisBox*, which was just anchored. Upon finding that the object is relatively light, the planner decides to use the recovery plan involving the push behavior, which succeeds (see Figure 6 (bottom)).

B. *A fusion experiment*

The purpose of the fusion experiment is to demonstrate how information of different types (in this case symbolic and perceptual), and from different sources, can be fused using a common representation space. This yields two main benefits in general, both of which are demonstrated here:

- *disambiguation*: when multiple hypotheses about the state of an object are maintained, or when multiple percepts could be associated with a certain object, complementary information from another source can aid in isolating the correct hypothesis;
- *improved estimates*: when multiple sources of information are considered, redundancy in information can be used to reduce imprecision and inaccuracy.

The domain considered in this experiment is position, and only the (x, y) dimensions are used. The fusion experiment is carried out using artificial data. The data is based on a coarse model of the PEIS home, shown in Figure 7(a).

This model includes two rooms (a kitchen and a bedroom), as well as two tables. One of the tables *Table-01* is a PEIS, and possesses an RFID reader which can read the RFID tags of any object placed on its surface. There are two other PEIS in the model. One is an overhead camera, placed in the kitchen, called *Camera-01*. The other is the robot *Pippi*, initially assumed to be in the bedroom. The scenario unfolds as follows.

- **Information 1 (Symbolic):** Initially, a human called *Alex*, and *Pippi*, are in the bedroom. *Alex* tells *Pippi* to get his cup from the *kitchen*. *Pippi* looks into a database, and finds that Alex's cup is *Cup-22*, and that it is a PEIS. *Pippi* now knows that *Cup-22* is in the kitchen, and proceeds in that direction. Figure 7(b) shows the fuzzy set used by *Pippi* to represent this information. White areas indicate possible positions (i.e. $\mu(x, y) = 1.0$), black areas are impossible (i.e. $\mu(x, y) = 0.0$), and gray values are somewhere in between. The dot in the middle of the kitchen is the center of gravity of the fuzzy set. In this case, positions in the kitchen are possible, all other positions are impossible.
- **Information 2 (Symbolic):** *Pippi* subscribes to *Cup-22*'s appearance tuple, and finds that the cup is `green`. This information can now be used as a matching criteria, as in the previous experiment.
- **Information 3 (Perceptual):** *Pippi* now subscribes to all information about cups, to see if anyone has information regarding the positions of cups in the PEIS home. *Camera-01* sees three cups in the kitchen. However one of the cups is red, and can be ignored, since it does not pass the matching test (due to information 2). There could also be information from other sources about cups in other rooms; however, these would not pass the matching test either, since *Pippi* knows that the desired cup is in the kitchen. The perceived positions of the cups which are seen by *Camera-01* (and which could possibly be *Cup-22*) are shown in Figure 7(c). The result of fusing this information with previous information is shown in Figure 7(d).
- **Information 4 (Symbolic):** *Table-01* detects that *Cup-22* is somewhere on its surface (Figure 7(e)), since *Cup-22* has an RFID tag. The result of fusing this information with previous information is shown in Figure 7(f). Note that this has led to disambiguation, as only one of the cups observed by the camera has a position which coincides with that of the table.
- **Information 5 (Perceptual):** Armed with all the previous information, *Pippi* can now navigate to the position of *Table-01*, and perceive the position of *Cup-22* (Figure 7(g)) using an on-board camera. The result

of combining all the information together is shown in Figure 7(h). Note that camera observations (from both *Camera-01* and *Pippi*'s camera) have less bearing uncertainty than range uncertainty. Since the observations come from different directions, this allows the effective range uncertainty to be drastically reduced, yielding a more accurate estimate. This final position estimate is accurate enough to allow *Pippi* to grasp the cup, and bring it to *Alex*.

VII. CONCLUSIONS

In this work we have described our approach to including robotic technologies into common environments, using PEIS-Ecologies. We also recalled the *perceptual anchoring problem*, which is the problem of creating and maintaining the connections, inside a robotic system, between symbols and sensor data which refer to the same physical object. We then extended the concept of anchoring inside a single robot to the case of anchoring in PEIS-Ecologies. In PEIS-Ecologies, multiple PEIS need to agree about which symbols and percepts refer to the same objects. In addition, some of the objects might themselves be PEIS. For these objects, the robot-object relation should be seen as a robot-robot relation, and it should be addressed using the tools of cooperative robotics.

We perform cooperative anchoring in PEIS-Ecologies using individual and shared *anchor spaces*, which provide a common reference frame in which items of information can be compared and combined. In particular, both symbolic and perceptual information, coming from various sources, can be mapped into the same space, using grounding functions. Having the information in the same shared anchor space allows matching of percepts and symbols to be performed. Fusion of this information (again, of different types and from different sources), can also be performed using the shared anchor space, allowing complementary and redundant information to be exploited in order to improve estimates of object properties. We then illustrated how matching and fusion can be performed in our framework experimentally.

Many open questions yet remain. The matching and fusion functionalities described here are fundamental enough that it might be useful to have them as part of the PEIS-Ecology middleware. However this might not always be useful or possible, depending on the processing capabilities of the individual PEIS. Perhaps the matching and fusion could be offered as a service, provided by certain PEIS.

Another open question is where to store the shared anchor spaces. If the object to which the anchor refers is a PEIS, then it might make sense to store it (at least virtually) in the object. But what if the object is not a PEIS? Distributed or centralized approaches might both be useful in some contexts.

The cooperative anchoring problem is relatively untouched in the literature, and as such we intend to test various ideas using a progression of scenarios, which look at different types of information, and possibly different approaches to the matching and fusion operations.

ACKNOWLEDGMENTS

This work is supported in part by CUGS (the Swedish computer graduate school), and in part by ETRI (Electronics and Telecommunications Research Institute, Korea) through the project "Embedded Component Technology and Standardization for URC (2004-2008)". Our thanks to Mathias Broxvall and Marco Gritti for helping with the concepts and implementation of a PEIS-Ecology.

REFERENCES

[1] N.Y. Chong an H. Hongu, K. Ohba, S. Hirai, and K. Tanie. A distributed knowledge network fo real world robot applications. In *Proc of the IEEE/RSJ Int Con on Intelligent Robots and Systems*, pages 187–192, 2004.

[2] M. Broxvall, M. Gritti, A. Saffiotti, B.S. Seo, and Y.J. Cho. PEIS ecology: Integrating robots into smart environments. In *Proc of the IEEE Int Conf on Robotics and Automation*, Orlando, FL, 2006.

[3] J-P. Cánovas, K. LeBlanc, and A. Saffiotti. Robust multi-robot object localization using fuzzy logic. In D. Nardi, M. Riedmiller, and C. Sammut, editors, *RoboCup 2004: Robot Soccer World Cup VIII*. Springer-Verlag, Berlin, DE, 2004.

[4] A. Chella, S. Coradeschi, M. Frixione, and A. Saffiotti. Perceptual anchoring via conceptual spaces. In *Proc. of the AAAI-04 Workshop on Anchoring Symbols to Sensor Data*, San Jose, CA, 2004.

[5] S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: preliminary report. In *Proc. of the 17th AAAI Conf.*, pages 129–135, Austin, TX, 2000.

[6] S. Coradeschi and A. Saffiotti. An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2-3):85–96, 2003.

[7] M. Dietl, J.-S. Gutmann, and B. Nebel. Cooperative sensing in dynamic environments. In *Proc of the IEEE/RSJ Int Con on Intelligent Robots and Systems*, pages 1706–1713, 2001.

[8] F. Dressler. Self-organization in autonomous sensor/actuator networks. In *Proc of the 19th IEEE Int Conf on Architecture of Computing Systems*, 2006.

[9] J.H. Kim, Y.D. Kim, and K.H. Lee. The third generation of robotics: Ubiquitous robot. In *Proc of the 2nd Int Conf on Autonomous Robots and Agents*, Palmerston North, New Zealand, 2004.

[10] G. J. Klir and T. A. Folger. *Fuzzy sets, uncertainty, and information*. Prentice-Hall, 1988.

[11] N. Kubota and K. Nishida. Cooperative perceptual systems for partner robots based on sensor network. *Int Journal of Computer Science and Network Security*, 6(11):19–28, 2006.

[12] J.H. Lee and H. Hashimoto. Intelligent space – concept and contents. *Advanced Robotics*, 16(3):265–280, 2002.

[13] A. Loutfi, S. Coradeschi, and A. Saffiotti. Maintaining coherent perceptual information using anchoring. In *Proc of the Int Joint Conf on AI*, Edinburgh, UK, 2005.

[14] F. Michahelles, A. Schmidt, B. Schiele, and M. Beigl. Towards distributed awareness – an artifact based approach. In *Proc of the 6th IEEE Workshop on Mobile Computing Systems and Applications*, pages 82–91, 2004.

[15] Network Robot Forum. www.scat.or.jp/nrf/English/.

[16] The PEIS ecology project. Official web site. www.aass.oru.se/˜peis/.

[17] A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, 1(4):180–197, 1997.

[18] A. Saffiotti and M. Broxvall. PEIS ecologies: Ambient intelligence meets autonomous robotics. In *Proc of the Int Conf on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, pages 275–280, Grenoble, France, 2005.

[19] D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proc of the IEEE Int Conf on Robotics and Automation*, Seoul, Korea, 2001.

[20] L. Steels. *The Talking Heads Experiment*, volume 1. Words and Meanings. Laboratorium, Antwerpen, 1999.

[21] S. Weber. A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms. *Fuzzy sets and systems*, 11:115–134, 1983.

[22] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[23] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

(a) A coarse model of the PEIS apartment.

(b) **Information 1:** *Cup-22* is in the kitchen.

(c) **Information 3:** *Camera-01* sees two green cups.

(d) **Information 1,3:** Combination of items 1 and 3.

(e) **Information 4:** *Table-01* detects *Cup-22*.

(f) **Information 1,3,4:** Combination of items 1, 3 and 4.

(g) **Information 5:** *Pippi* sees *Cup-22*.

(h) **Information 1,3,4,5:** Final position estimate.

Fig. 7. Progression of estimates for the position of *Cup-22*, as new items of information arrive. The darker areas are more possible, the light areas are less possible. The dots indicate the center of gravity of the possible areas.