# FUNCTIONAL MAPPING FOR HUMAN-ROBOT COLLABORATIVE EXPLORATION

Shanker Keshavdas & Geert-Jan M. Kruijff
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken
Germany
{Shanker.Keshavdas,gj}@dfki.de

**ABSTRACT**
Our problem is one of a human-robot team exploring a previously unknown disaster scenario together. The team is building up situation awareness, gathering information about the prescence and structure of specific objects of interest like victims or threats. For a robot working with a human team, there are several challenges. From the viewpoint of *task-work*, there is time-pressure: The exploration needs to be done efficiently, and effectively. From the viewpoint of *team-work*, the robot needs to perform its tasks together with the human users such that it is apparent to the users why the robot is doing what it is doing. Without that, human users might fail to trust the robot, which can negatively impact overall team performance. In this paper, we present an approach to the field of semantic mapping, as a subset of robotic mapping; aiming to address the problems in both efficiency (task), and apparency (team). First, we assess the situation awareness of rescue workers during a simulated USAR scenario and use this as an empirical basis to build our robots spatial model. The approach models the environment from a geometrical-functional viewpoint, establishing where the robot needs to be, to be in an optimal position to gather particular information relative to a 3D-landmark in the environment. The approach combines top-down logical and probabilistic inferences about 3D-structure and robot morphology, with bottom-up quantitative maps. The inferences result in vantage positions for information gathering which are optimal in a quantitative sense (effectivity), and which mimic human spatial understanding (apparency). A quantitative evaluation shows that functional mapping leads to significantly better vantage points than a naive approach.
**KEY WORDS**
Autonomous Robotics, Ontology, Semantic Mapping

## 1 Introduction

When a rescue team reaches a disaster environment, they seldom have information about the spatial organization of it. The tasks of the rescue team are then to typically explore the environment, identify objects of interest such as victims, fires, explosive risks; and perform actions such as rescuing victims and extinguishing threats. Among these tasks, exploration and identification of "objects of interest" such as victims, hazardous substances are tasks that are performable by the robot. See Fig. 1 for illustrations of environments in which we have deployed human-robot teams. For example, in responding to a tunnel traffic accident the priority is to search for victims (inside cars), whereas in a freight train accident we need to assess the presence of dangerous materials. Exploration of the environment helps build an awareness of the situation which proves invaluable to rescue workers. The traditional method of a robot building up it's own spatial awareness is by building a metric map i.e. of laser scans and visual information. However that alone is of limited use to a rescue worker.

Instead rescue workers might be more interested in a *semantic map*, which is described in [20] as a map which contains in addition to metric information, assignment of mapped features (laser, vision) to entities of known classes. Semantic maps allow users to communicate to the robot referring to entities that are present in the environment. The system matches *keywords* in this communication with discovered or added entities in the semantic map. This communication could be through a spoken dialogue system [1, 31] or through a user interface [25]. We discuss in more detail, the references made to objects in our environment by rescue workers in §3.

The mentioned approaches [25, 31] use knowledge bases with associated reasoning engines to gather further knowledge of such entities. Our approach differs from these in that, we perform more detailed functional-geometrical reasoning and our environment is largely unstructured. In the field of search and rescue, known or commonly expected entities in the case of a car crash would be cars, victims and so on. In our approach, we make use of a handwritten OWL/RDF-based ontology based upon objects of interest that may be observed in a disaster environment, and their relation to each other. Our approach is *functional* in the sense that, the system is adaptable to the functions that the robot and it's sensors may possess. We present this information in more detail in §4.3.

Our approach to semantic mapping address both efficiency (task), and apparency (team). Our focus is on the robot exploring and understanding the spatial structure of the disaster environment from the viewpoint of *information gathering*. Objects of interest often "contain" (in the topological sense) additional information that can be retrieved

from it. For example, a car might contain victims or a barrel might have a label identifying the explosive substances present within. In the former case, it would help for the robot to be in optimally computed position to gather information relative to the car i.e., the presence and locations of victims in the car. This is a process of inference and discovery. Upon the perception of a particular landmark, inference establishes whether the landmark might contain particular objects of interest. Gathering information then turns into verifying whether these hypotheses hold, and if verified, substantiating them as facts.

The context of our task is one of collaboration between humans and robots, with both being problem-holders. The humans need a robot to provide them with information about an environment which is too dangerous for them to (currently) enter, whereas a robot needs the humans to help it to make sense of the environment or to find its way around. Complications in this collaboration arise both in its *task-work* dimension, and the *team-work* dimension (cf. [13, 14]): Tasks typically need to be performed under time-pressure, requiring the robot to execute them efficiently and effectively; and, the way the robot does so needs to be understandable or *apparent* for the human user to trust the robot in determining and executing its own actions [6, 11].

(a) (b) (c)

Figure 1. Examples of where we have deployed human-robot teams: Tunnel accident (a), earthquake (b), train accident (c). (a) and (c) are at training areas, (b) is real-life (Mirandola, Italy; July'12).

Our approach achieves efficiency by considering how the structure of the landmark, the functional capabilities of the robot, and the actually observed situation around the landmark, all interact to establish positions where a specific action can be optimally executed. We refer to these positions as *vantage points*. We formulate optimality as a quantitative measure of the success of the action, e.g. maximum visibility into a landmark given position and sensor models. Apparency is achieved by basing vantage point selection on the kinds of the inferences humans tend to make about space and "affordances," i.e. from a functional-geometrical understanding of space [5]. For example, if the robot needs to look into a container-like object like a car, it makes more sense to be at openings (windows) rather than an arbitrary end (e.g. the tailpipe). Doing so naturally facilitates making better observations, but it also results in behavior which a human user can intuitively explain – and thus, possibly, trust.

An overview of the paper is as follows. §2 relates our approach to other work on knowledge gathering, and active visual search. §3 describes a specific search and rescue scenario, the "tunnel accident". We describe how rescue workers explore such an accident and the field data thus gathered forms the empirical basis for our approach. §4 describes the approach in more detail, including offline- and online workflows. §5 presents the experimental setup, and quantitative results comparing our approach to a naive one, on a tunnel accident use case. The paper ends with conclusions.

## 2 Related Work

The basis of our research comes from the field of semantic mapping which is still at a nascent stage. Most approaches either use a complex spatial intelligence in structured environments or conversely a low-level spatial intelligence in unstructured environments. [20] provides several cases of semantic relations used to identify and label different planes of an indoor scenario based on their relative orientation. Using similar methods, the authors also demonstrate the identification of a ground plane in an outdoor scenario. Other indoor semantic mapping approaches include using laser scan patterns to classify rooms [9] and determining the type of room based on the objects found in them [28]. These approaches are a computationally interesting form of semantic mapping, but do not yield more complex spatio-cognitive structures. On the other hand, our approach uses human readable ontologies based on task-specific knowledge of human beings. A precursor to our approach was [30], where authors demonstrated a method for an indoor robot to recognize common indoor themes like doors, and the regions for interacting with them. The authors use spatial knowledge based on human interactions with doors to draw it's conclusions. A recent approach using spatial ontologies was [25], where an indoor robot observed the interaction of a human being with a kitchen environment and then uses an ontology derived from this knowledge to interact with the objects in that environment. These approaches are on the other end of the spectrum since the environment is indoor and controlled.

Another aspect of our approach comes from robotic exploration of unstructured and previously unknown environments. The current state of the art in mobile robotics is limited in terms of autonomous planning and exploration of such environments. We would like the robot to be capable of (collaborative) forms of exploration for information gathering, similar to those discussed by e.g. Wyatt et al [29]. We would like to cast exploration as a continual planning & execution process in which inferences are made over what information is missing, where such information might be gathered, and what actions to perform in order to gather this information.

This is different from an exhaustive search of the disaster scene, as would result from typical information-theoretic approaches to spatial exploration; cf. [26]. It is more similar to active visual search techniques, in which vantage points are planned in a particular space to search

for (or observe) a known object. This is potentially a hard problem to restrict. In [8], an *indirect search* is suggested, where searching for one object helps restrict the search possibilities for a target object. However, Tsotsos [27] showed even this problem to be NP-hard in the general case. Plan-based approaches like [1] then couple semantic knowledge of spatial structure, like basic containment relations, with search heuristics to help structure the search. A demonstration of this approach is even shown in large, unknown spaces [2, 3]. Our approach relates to work in active visual search, in that we reason about possibilities for information gathering in an "indirect search" way (like [8]). We then use continual planning to drive discovery i.e., we make inferences on the objects we observe which generates new plans for information gathering.

Functional mapping was coined in [30] in which we consider only ontological inference to establish functional aspects of space. In §3 we discuss empirical results from end-user experiments with human-robot teams [6] in which human users tele-operating a robot (UGV) displayed "exactly" the kind of behavior in selecting optimal viewpoints for exploration as predicted by our approach. In §4 we describe a combination of top-down ontological inferences about the structure of 3D landmarks, with Support Vector Machines(SVM)-based probabilistic inference for determining optimal positions relative to a given landmark and (inferences over) a given robot morphology including physical shape and sensor characteristics. We provide a more precise, functional-geometrical characterization of space in terms of the environment and the way the robot (given its configuration) can interact with it. Furthermore, we provide a setup for quantitative analysis of the approach (in simulation), and present experimental evaluation results. The idea of deriving inferences from ontologies detailed with task-specific human knowledge comes from papers such as [10, 30]. Our approach makes a concise model of optimal positions for performing specific tasks similar to the approach [25]. The authors have a concept similar to functional mapping called *action-related places*. Action-related places use training data to create point distribution models to reduce the dimensionality of successful poses to perform a task. These point distribution models are then sampled from during testing operations. Our approach performs more complex geometrical inferencing, over more generalized objects and robot types and stores the successful poses as SVMs in the ontology.

## 3 Empirical Basis

Our scenario is one that involves a human-robot team jointly exploring a traffic incident in a tunnel. Vision is impaired by smoke filling the tunnel. We have performed high-fidelity simulations of the disaster scenario, with robots and firefighters at the training site of the Italian National Fire Watch Corps (SFO at Montelibretti, Italy) and at the one of the Fire Department of Dortmund, Germany (FDDO). In the setup at SFO, we wanted to observe the visual points of attention that firefighters maintained during a rescue operation and match these with their spoken communication. For this reason, they were equipped with eye-gaze machines that track their visual attention [16], and their communication during several mock rescue operations was also recorded. A sample audio recording of a firefighter read as follows:

(1) A car, a civil car, with some people inside.

(2) A family. People. A woman drives. A person in *the front seat*. A child. Another child in *the rear seat*. Another child, a baby.

One thing that can be observed here is the felicitous use of hearer-new definite descriptions (marked in italics) [22,23]. Definite descriptions are supposed to refer to mutually known entities in the domain of discourse. The information of the structure of the car (eg: rear seat) is from the mental representation of the firefighter, where the representation of a car has been evoked by the indefinite description "a car" (the so-called *trigger entity*). And through his *prior knowledge* about cars he can be assumed to know that cars in general have (front and rear) seats. Such uses of a definite description to refer to an implicitly evoked entity that can be inferred based on background knowledge are called "inferrables" [23] or "bridging anaphora" [4]. The group of bridging anaphora that come into play in our recordings are the so-called "indirect references by association", which Clark explains with their predictability as being an associated part of the trigger entity. From the transcriptions, we observe that the firefighter's task is tightly correlated with the hierarchical composition of the spatial structure: the tunnel contains cars, which in turn contain victims; a truck, which typically contains goods; and barrels which usually contain (potentially hazardous) substances. It is generally assumed that humans adopt such a (partially) hierarchical representation of spatial organization [19]. This demonstrates the kind of inferences on background knowledge that the robot must perform, not only to autonomously determine a plan for locating victims but to produce and comprehend natural language scene descriptions.

At another simulation scenario at FDDO, firefighters were given tele-operational control of the robot. The scenario was of an unknown smoke-filled environment and where they had to record the positions of vehicles, victims and hazardous material that they observed. Our interest in the experiment was to notice the vantage points the firefighters assumed when observing the inside of a car to look for victims, when looking at a motorbike, and explosive barrels. Once the trials were completed, we marked a boundary of 1 meter around the regions of interest (the car windows, the motorbike, and the barrel). We assumed that this was a sufficient visual range for affording the function of observing these regions of interest. We call the areas marked off by the boundaries as 'functional areas' – since these areas enable the function of observing these regions. In Fig. 2, we show the runs of three of the firefighters who participated in our experiment. Table 1 shows the percentage of 'observation time', or time spent inspecting the regions of interest. We further mention the percentage of the

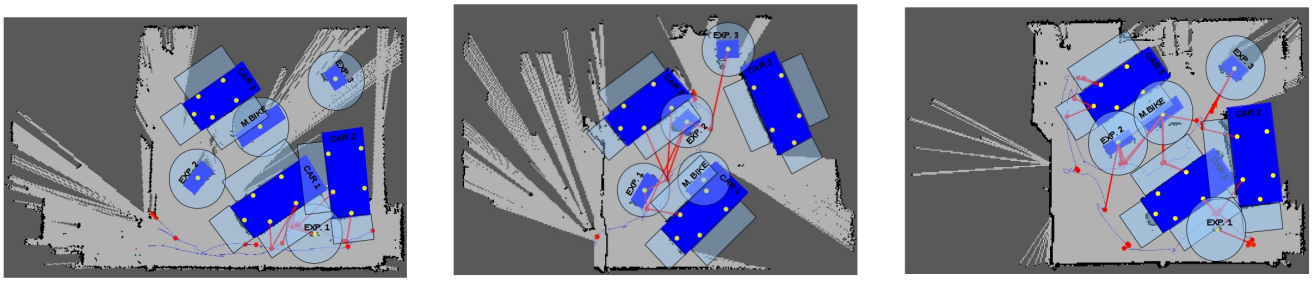


Figure 2. Maps acquired by tele-operation in FDDO, Germany, showing points from where observations/transcriptions were made(red), points of attention which they were observing(yellow), functional areas(light blue) and the path of the robot (blue trajectory).

| Participant | Percentage of observation time | Percentage of observation time in functional areas | Percentage of observation time in functional areas of different objects of interest | |
|---|---|---|---|---|
| | | | Vehicles | Expl. Barrels |
| 1 | 38.17 | 66.7 | 86.67 | 13.33 |
| 2 | 53 | 97.6 | 0 | 100 |
| 3 | 48 | 65.3 | 41.96 | 58.04 |

Table 1. An analysis of the time spent for the tele-operated runs shown in Fig. 2

observation time spent in functional areas of objects. From the data, we notice that Participant 1 and 3 spent over half, and Participant 2 spent nearly all observation time in the functional areas, divided into time spent observing vehicles and threats. This confirms our belief that rescue workers do employ strategic vantage points to observe regions of interest. We would like our robot to draw similar *human-compatible* spatial inferences to search for victims.

# 4 Approach

The following subsections detail various aspects of our approach. §4.1 describes the use of autonomous control in the system with a review of autonomy in HRI. §4.2 is a description of the semantic mapping system that supports our approach. §4.3 explains our use of ontology and subsequent reasoning. §4.4 explains the measure of visibility used for the search of victims in the car accident scenario. §4.5 and §4.6 explain the offline and online workflows used in our approach.

## 4.1 Link to Autonomy

Autonomous navigation of an unstructured disaster environment is a collaborative task, where full robot autonomy is currently beyond our scope. We have conducted simulated search and rescue exercises with firefighters in Germany and Italy and also been involved in a real rescue effort after the earthquake in Mirandola, Italy in 2012. We notice that rescue workers come under a lot of stress in such exercises and have to often conduct several tasks simultaneously e.g., rescuing victims, observing a scene, conveying information to superiors and discussing plans.

In a landmark report on the study of autonomy in human computer interaction, Sheridan [24] introduces the term *levels of autonomy* to indicate the different autonomy options that could be presented to the system operator. Further research in this field [7] studies the application of levels of automation on performance and cognitive workload in a dynamic control task. In a study on the effect of levels of automation on air-traffic control operators, Parasuraman [21] finds that varying the levels of automation appropriately can improve their working efficiency. The levels of autonomy should be varied according to task difficulty. Parasuraman uses the 10-point level of autonomy scale introduced by Sheridan from level 1 being complete teleoperation to level 10 being complete autonomy.

Our task of the control of a search and rescue robot is a cognitively demanding task like that of an air-traffic controller. Thus we choose to apply the same system of levels of autonomy. Traversal over a rough pile of rubble requires complete operator control. On the other hand, robot navigation over a relatively flat surface can be an autonomous task. Operator control can bypass the autonomous motion if the operator thinks the robot is performing the task wrong or if the robot requires help. In the task of locating victims in a car crash scenario, we would like the robot to proceed to the most viable points of gathering this information, provide continuous video feed to the operator and only have the operator intervene if the operator does not agree with the robot's plan or if the robot is stuck and requires teleop-

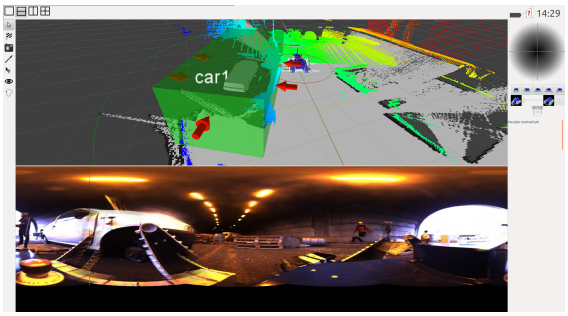


Figure 3. A screenshot of a simulated scenario with the NIFTi robot. The red arrows indicate the vantage point poses for looking into the detected car.

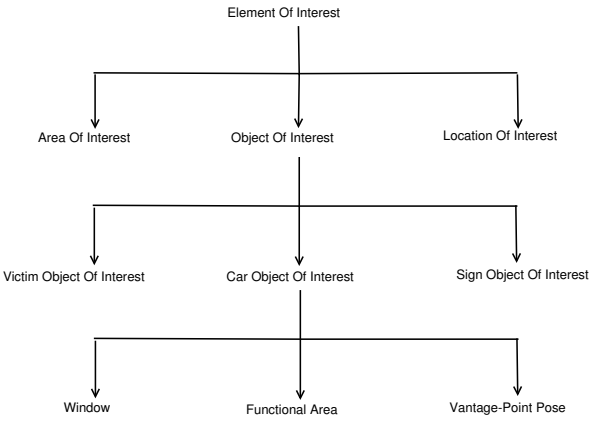


Figure 4. The semantic mapping entities present in the NIFTi system

eration.

This model are applied to other functionalities of our scenario too. For example, when the robot's vision component detects a crashed car with a low level of accuracy, it will ask the user to verify the detection. This which falls under level 4 of the model i.e., "(the robot) suggests one alternative and awaits user input". When the robot decides upon a car to explore, we indicate through the semantic map, the goal poses of the robot's navigation and embed them in the graphical user interface. The robot then executes that plan and allows the human operator the possibility of vetoing the suggestion. This falls under a higher level 7 in the levels of autonomy, namely "(the robot) executes automatically, and necessarily informs the human". The descibed scenario is demonstrated in a screenshot of our graphical user interface shown in Fig. 3, where a simulated rescue scenario is underway at the firefighter school (SFO) in Montelibretti, Italy.

### 4.2 Semantic Mapping

As described in §1, we use semantic maps that are metric maps annotated with additional information. This information can include entities perceived by the robot, entities perceived by the user or entities derived by the robot. As will be described in §4.3 and §4.5, the robot queries it's ontology for information and derives relations based on the information.

Fig. 4 shows the entities that are present in our semantic map. The base entity is called an *Element Of Interest*. This can be subdivided further into areas, locations and objects. The areas would be defined in the message by a polygon, the location by a point and the object by a point and further properties. The *Car Object Of Interest* shows elements that are contained (in a topological sense), within the type. The *Window* is an element that is stored in, and can be derived directly from the ontology. The *Vantage-Point Pose* and *Functional Area* are elements that denote optimal viewpoints for inspection inside the car. They are computed by the robot through geometrical inferencing described in §4.5. These entities are displayed on the user interface as shown in Fig. 3, where a *Car Object Of Interest* and *Vantage-Point Poses* can be seen. These can then be interacted with by the user, talked about through spoken dialogue or used by the planning subsystem.

### 4.3 Ontology

Our use of semantic mapping is to attach meaningful categories to areas in the metrical map. In [31], a mobile robot drives around an indoor scenario and assigns labels to certain areas based on their physical charecteristics. It first generally labels all explored areas as ontological instances of the class *Area*. Based on further exploration, it is then able to further classify them as of class type *Room* or *Corridor* based on the analysis of the metrical map. It does so by using a hand-written ontology and by reasoning about categories based upon relations of specificity like *is-a* i.e., *Room is-a Area*. Further if an object of class *Couch* is found in this area, through a relation of object containment it could make an associative relationship e.g., *Room1 has-a Couch*. Fig. 5 shows a sample of our ontology of a car accident domain where similar relationships are shown. The arrows signify the classification relationship *is-a*, and several *has-a* relationships have been indicated for the class *AudiR8*. The *has-a* relationships specify for e.g., the geometrical structure of the car like the positions and dimensions of the windows and the car cabin.

We use a handwritten OWL/RDF-based ontology with manufacturer information about "car-accident" domain entities such as cars, robots, their sensors and so on. In our previous paper [15], we retrieved geometrical features of car models and functional and geometrical features of robot and sensor models from the ontology to use in our computation of optimal poses for finding victims.

In this paper, we extend the ontology to include information of the car cabin (i.e., the space where the passengers are seated). As will be explained in §4.4, we then compute information regarding the optimal "vantage points",

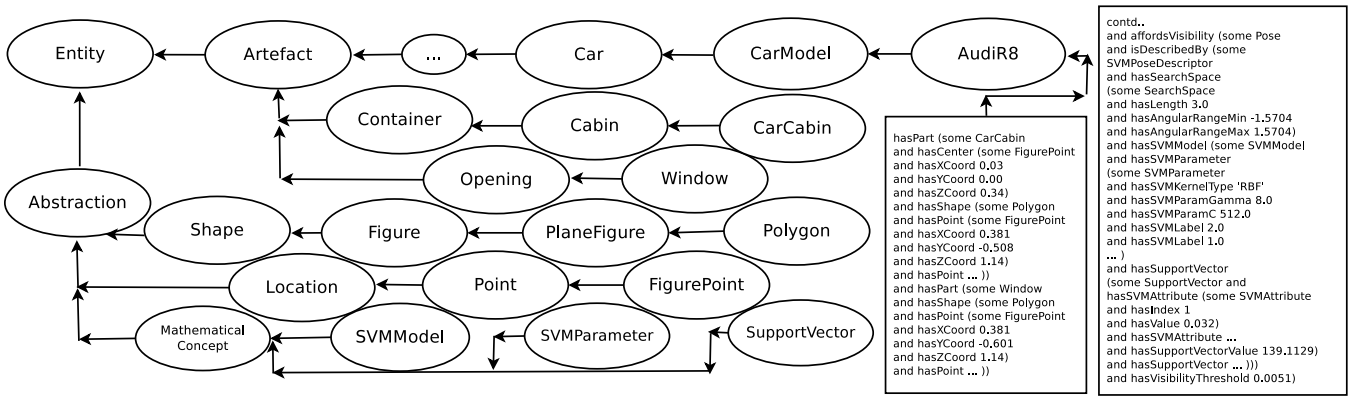


Figure 5. An excerpt of the car accident domain ontology. Details of a car with geometrical information as well as computed SVM models for visibility can be seen.

to look for victims inside the car cabin. We do so by querying the ontology for physical and functional parameters of the scene and use them as spatial parameters in our calculation of these vantage points. This is done during an offline step, and is added back to the ontology as explained in §4.5. We use SVMs to concisely represent the vantage points. The relationship *has-a* for representing the vantage points for the car Audi R8 in terms of SVM models can also be seen in Fig. 5. It is important to note that although we are computing the optimal positions for looking into a *Car* through a *Window*, our approach is relevant to any members connected by the *is-a* relation to these entitites i.e. *Container* and *Opening*. Thus it may be just as easily applied to inspection of a container on an automatic packaging line, robots working on automobile production etc.

To extract information from the ontology we submit queries to the HFC reasoning engine with a standard OWL-DL rule set and some custom rules [17, 18]. For example, to retrieve geometrical information about the corner points of the car cabin of the Audi R8, we submit the query:

*SELECT DISTINCT ?pnt WHERE <funcmap:AudiR8> <funcmap:hasPart> ?carcabin & ?carcabin <rdf:type> <funcmap:CarCabin> & ?carcabin <funcmap:hasShape> ?shape & ?shape <funcmap:hasPoint> ?pnt*

We submit similar queries to retrieve information about the SVM models that we store, robot and sensor information etc. The NIFTi robot possesses a detachable robot arm, that is seen in Fig. 6 used to look into a car. The ontology possesses information about the robot's morphology, any arms or cameras attached etc. If we would like to find out physical properties for e.g., the range of the camera that is mounted on the arm of the robot, we would submit the query:

*SELECT DISTINCT ?range WHERE <¡funcmap:NIFTiUGVWithArm> <funcmap:hasPart> ?arm & ?arm <rdf:type> <funcmap:RoboticArm> & ?arm <funcmap:hasPart> ?camonarm & ?camonarm <funcmap:hasRange> ?range*

Figure 6. The NIFTi robot looking into a car with the arm. The physical configuration of the arm is written into the ontology for use during reasoning.

Naturally the queries to determine camera properties, could be modified if we queried and found out that there was no arm on the robot. This helps the flexibility of our approach, as we can query variable configurations and perform our geometrical reasoning on the basis of different configurations. We use the same method to find out the camera's field of view, the degrees of freedom and reach of the arm etc.

### 4.4 Measure Of Visibility

The measure of visibility is a measure of the likelihood of a human operator succesfully locating a victim through looking at the robot's camera feed from a certain position around a car. In [15], we used the area of the car window visible in the visualization cone of the robot's camera(the viewable volume in front of a camera) as shown in Fig. 7, comparing it to the average size of a human face, which would be detectable by a vision component running face detection algorithms. However, we found that face detec-

tion is unreliable in smoky environments that we typically find in such disasters, and the measure was not very accurate as it ignored the rest of the car cabin where victims may also be found. As mentioned in our discussion about *sliding autonomy* in §4.1, our scenario is one where the operator is overviewing the video feed provided by the robot. We feel the operator is in a better position to do a critical task such as determining if a certain area in a smoky video feed contains a part of a human being, thus removing the autonomy from the robot in that particular task.

In our current approach, we feel a better measure of visibility would be the volume of the car cabin, which is where the passengers are located, that is visible from a certain robot position. The idea is that, if we then plot a path of such viable locations, we want to maximize the volume of the car cabin visible in the robot's camera feed while the robot maneuvers through that path. That will then give the human operator the highest possible chance of locating a victim in the region of the car cabin. To have an idea of the volume of the car cabin, we fill the model of the car cabin with equal radii packing spheres in a hexagonal close-packing arrangement, as shown in Fig. 7. This arrangement has the highest packing density. We then calculate the visibility measure from any robot position around the car as, the ratio of the packing spheres visible from that location to the total number of packing spheres. The algorithm for computing the visibility measure is then given by Algorithm 1.

To demonstrate the visual region that can be seen by a camera, we use a visualization cone that contains the regions that a camera can see from a certain position. In Fig. 7, we can see the visualization cone from a certain position around the car model looking towards it. Let the visualization cone have a horizontal angle $H$ and a vertical angle $V$, $r_{hv}$ be a ray of the cone with angles $h$ and $v$, $R$ be the reliable range of vision for the camera in this scenario, $p_{cam}$ be the camera, $S$ be the set of packing spheres, $r_{sph}$ be their radii and $W$ be the set of window polygons. Let the expressions *ray(A,B)* be a ray from point $A$ to $B$, *dist(A,B)* be the orthogonal distance between $A$ and $B$, *proj(A,B)* be the projection of ray $A$ on ray $B$, and *int(A,B)* be the point of intersection of ray $A$ on polygon $B$. In Algorithm 1, equation 6 isolates only those spheres that the current ray passes through. Equations 7 and 9, then are criteria on whether the length of intersection of the ray and the point, and that of the ray and window; are less than the visual range of the camera. The visibility measure is then calculated in Equation 19.

### 4.5 Offline Workflow

We use Support Vector Machines (SVMs) to form concise models of high-visibility yielding vantage point poses. We use the RBF kernel and our 3 input parameters are the X, Y coordinates and the 2D angle of the robot with respect to the detected car($\theta$). In Fig. 7, we can see car windows and *search spaces* corresponding to each of them. Using linear

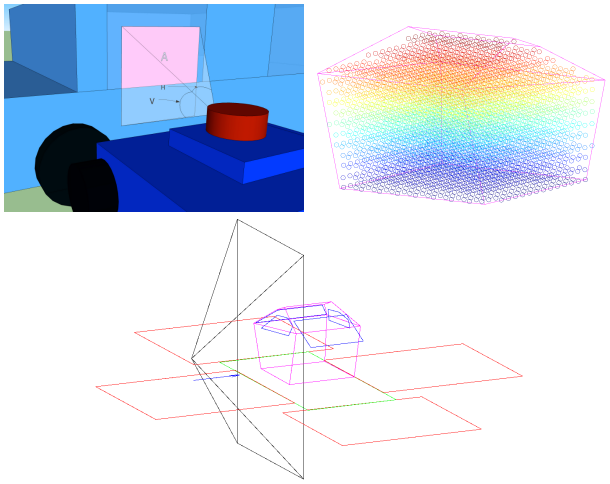

Figure 7. (clockwise from top-left) :(a) A sketch of a robot looking inside a car, the camera's visualization cone with horizontal angle H and vertical angle V is able to view an area A of the car window;(b) a car cabin(pink frame) with packing spheres (multi-coloured);(c) a sketch of a visualization cone of a robot's camera(black) looking at a car cabin (pink) through car windows (blue polygons), also shows search spaces(red), car outline(green) and robot pose(blue arrow). Figures (b) & (c) are constructed from information extracted from the ontology for the Ladybug3 camera and an Audi R8 car

iterators(*lin*), say of $10^{-2}$ of searchspace side length, and angular iterators(*ang*), say ($\pi$/16); for each search space we generate a search set($S$) of $16 * 10^4$ robot poses. With the addition of the arm, we simply increase the number of iterations in the searchspace, including iterations from the minimum to maximum reach of the arm. For each of these poses, we then compute the measure of visibility(*vismes*) as described in §4.4.

Of these values, a large number give zero visibility. The others give varying amounts of positive visibility. We use a system of set of varying thresholds($T$), based upon the top percentile of positive visibilities. We think this will give the human operator, a choice between very high visibility (say top 10%) or a larger range of visibilities (say top 50%). The online step of the algorithm is very fast, so the human operator may switch between various ranges of visibilities very easily if desired. The thresholds also help in the evaluation of the method in §5. We choose a specific threshold thus classifying the set $S$ into 2 classes. We choose the best RBF kernel parameters ($c$ and $\gamma$) by performing cross validation through coarse($CG$) and fine grid($FG$) search parameters on these 2 classes. We then use these *best parameters* to create an SVM model ($M_{sp,t}$) for this particular search space and threshold, consisting of about 500 support vectors. The robot would finally store the SVM model along with the search space parameters and the threshold to the ontology.

Typically a robot ($r$) would perform these steps offline

**Algorithm 1** Computing visibility of car cabin from single robot pose

$n_{sph} \leftarrow 0$
$N_{tot} \leftarrow |S|$
**for** $h \leftarrow 0, H$ **do**
  **for** $v \leftarrow 0, V$ **do**
    **for all** $s_c \in S$ **do**
      **if** $dist(r_{hv}, s_c) < r_{sph}$ **then**
        **if** $proj(ray(p_{cam}, s_c), r_{hv}) < R$ **then**
          **for all** $win \in W$ **do**
            **if** $dist(int(r_{hv}, win), p_{cam}) < R$ **then**
              $n_{sph} \leftarrow n_{sph} + 1$
              $S \leftarrow S - s_c$
            **end if**
          **end for**
        **end if**
      **end if**
    **end for**
  **end for**
**end for**
$vismes \leftarrow n_{sph}/N_{tot}$

on all car models (*V*) present in it's ontology. For a car model(*v*) we get the search space set (*SP*) and from both robot and car we get the physical (window, car cabin dimensions) and functional (camera range, view angles) parameters ($param_{phy,fun}$). These steps are shown in Algorithm 2.

One iteration of the offline workflow takes about 6 hours on a fairly powerful computer (8 core, 2.8GhZ). We argue that this is acceptable, since this offline process has to be performed only once on every robot for every car model present in the ontology. The SVMs also clearly reduce the dimensionality of the vantage point poses, enabling them to be stored easily in an ontology. Retrieval of SVM model can be easily done through queries, and once retrieved computing the classification for a test pose is a simple process.

### 4.6 Online Workflow

Fig. 8 shows the online workflow, which takes place after the offline workflow has been completed for every car model. In step 1, when a car is detected, the robot retrieves from the ontology for that particular car model each search space and threshold, the SVM model and linear and angular iterators it had stored in the last step of the offline process. In step 2, a particular search space and threshold are chosen. All search spaces may be chosen one at a time, or a certain search space may be chosen for proximity to the robot to have a quick look. The thresholds are chosen according to the operators choice, based on the type of visibility desired, i.e. high visibility or a broad range of visibilities. In step 3, using the linear and angular iterators of the search space, a random robot pose is generated. If the robot has an arm, this is detected by looking into the ontology. If so the length of the arm is also randomized through iterators. Then the robot pose is appended with this arm

**Algorithm 2** Offline workflow

**for all** $v \in V$ **do**
  $SP \leftarrow$ GetSearchSpaces$(v)$
  $param_{phy,fun} \leftarrow$ GetParameters$(r, v)$
  **for all** $sp \in SP$ **do**
    $S \leftarrow$ GenerateSearchSet$(sp, lin, ang)$
    **for all** $s \in S$ **do**
      $vismes \leftarrow$ MeasureOfVisibility$(s)$
    **end for**
    $S \leftarrow (S, vismes)$
    **for all** $t \in T$ **do**
      $S \leftarrow$ ClassifyThreshold$(S, t)$
      $CG \leftarrow (..2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2, 4, 8, 16..)$
      **for all** $cg \in CG$ **do**
        $(c_{cg}, \gamma_{cg}) \leftarrow$ CrossValidation$(cg, S)$
      **end for**
      $(c_b, \gamma_b) \leftarrow max(c_{cg}, \gamma_{cg})$
      $FG \leftarrow (..(c_b, \gamma_b).2^{-0.3}, (c_b, \gamma_b).2^{-0.1}..)$
      **for all** $fg \in FG$ **do**
        $(c_{fg}, \gamma_{fg}) \leftarrow$ CrossValidation$(fg, S)$
      **end for**
      $(c_b, \gamma_b) \leftarrow max(c_{fg}, \gamma_{fg})$
      $M_{sp,t} \leftarrow$ CreateSVMModel$(c_b, \gamma_b, S)$
      AddToOntology$(M_{sp,t}, sp, t, lin, ang)$
    **end for**
  **end for**
**end for**

length. Next, the robot pose is checked against the SVM model to see if it is classified in the class of visibility above the threshold. This process is repeated till a suitable pose is found. For all the cases that we have tested, this takes a very short amount of time, upto 5 seconds. We believe this is a reasonable amount of time for getting a pose that might yield good visibility. Additionally, it is also possible to check the measure of visibility for this pose against the car model, which can be evaluated very quickly. However, this is usually not necessary as the cross-validation performed in the offline step usually produces a very high rate ($> 95\%$), as the successful cases are well ordered and can easily be clustered. Finally, this vantage point pose can be used as a planning coordinate.

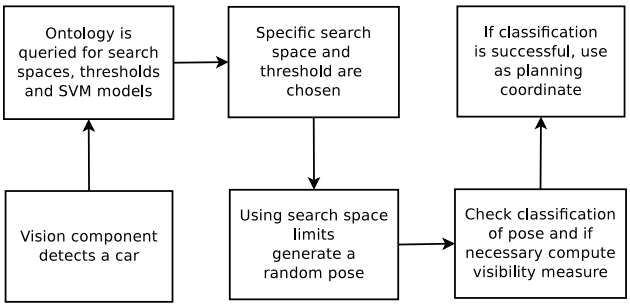


Figure 8. Schematics of the online functional mapping workflow

# 5 Experiment

## 5.1 Impact of Environment on Real Trials

Performing experimental trials in our setup requires a robustly working mapping, vision detection and planning interfaces. However the state of the art in these components as applied to outdoor scenarios does not ensure a robust system. The vision detection system used in NIFTi is described in [12]. The real scenario consists of a mixture of unpredictable lighting and prescence of smoke that impairs robustness in live experiments. The state of the art in path planning in unstructured environments is not reliable enough for live experiments either. In most current approaches, it is required at the very least that the area be premapped. This is not a realistic scenario.

In §2, we explain the *Action-Related Places* approach in comparison to ours. For this approach, autonomous motion could be implemented since the environment was largely static, indoors and planar. The object detection was robust since the authors were detecting kitchen items in indoor environments as compared to vehicles in our case. For these reasons, we have chosen to demonstrate our results in simulation.

## 5.2 Simulation Experiments

We found our method difficult to evaluate during real experiments, due to unreliable results from the vision and navigation components, which are managed by other partners in our project. This is expected as given the severe environmental conditions (uncertain lighting, smoke, rough and uneven terrain, unexpected obstacles) in these scenarios, the current state of the art approaches in these fields do not perform robustly. Thus it is difficult to obtain test data from a real scenario. Instead we run the offline workflow as usual, and generate the poses from the online workflow. We then check the measure of visibility obtained from these poses on a simulated car model which is generated from the car dimensions of the ontology.

We compare the visibility obtained from these poses to pose obtained from a more naive approach. For the naive approach, we wanted to choose poses that do not consider the structure of the car but are aware of the position and size of the car. These dimensions can easily be seen from a 2D occupancy map, like one that is generated from a laser scan with 2D mapping. The positions of the naive approach were random points around the car up to a distance of the search space length of 3m. The directions of the robot for the naive approach, were chosen such that they pointed to any point on the model of the car. Thus the robot in the naive approach has an understanding of where the car is, but does not know what parts it is composed of e.g., windows.

We calculated the measures of visibility obtained from 5000 robot poses generated from the functional mapping approach and the naive approach. We performed experiments with 2 robot models and 2 car models and got consistent results for all the cases.

| Case | Threshold Visibility Percentage | Naive Algorithm Visibility | Functional Mapping Visibility |
|---|---|---|---|
| 1 | 50% | 1.3732 % | 2.6416% |
| | 25% | 1.3012 % | 3.4687% |
| | 10% | 1.3623 % | 4.6090% |
| 2 | 50% | 0.9352 % | 1.5547% |
| | 25% | 0.9107 % | 1.6379% |
| | 10% | 0.8997 % | 2.0271% |
| 3 | 50% | 6.9358 % | 11.6519% |
| | 25% | 7.4886 % | 15.8252% |
| | 10% | 7.3456 % | 22.5355% |
| 4 | 50% | 2.5736 % | 5.3523% |
| | 25% | 2.7935 % | 6.0341% |
| | 10% | 3.0426 % | 6.9374% |

Table 2. Comparison of achieved positional visibility by naive algorithm and functional mapping. Case 1 was with the NIFTi robot and the Audi R8, case 2 with the NIFTi robot and the BMW 3Series Sedan, case 3 with the Pioneer PeopleBot and the Audi R8 and case 4 with the NIFTi robot with the arm and the Audi R8

Table 2 summarizes the results. We used as robot models the robot developed during the NIFTi project which is equipped with a Ladybug 3 omnicamera at a height of 40 cm and the popular Pioneer PeopleBot equipped with two Flea 2 cameras fitted on the top of the robot at a height of about 145 cm. We have an additional configuration of the NIFTi Robot equipped with a customized arm. The arm only has a degree of freedom in the vertical direction. It is mounted at 34 cm and has a vertical reach of 112 cm. The arm has a Flea 2 camera mounted for the simulation. From the results, we see that even using a poor threshold of $50\%$ i.e., using $50\%$ of non-zero visibility poses as a basis for the SVM model yields almost twice as good visibility of the car cabin as the naive approach. As we reduce the successful visibility threshold percentage to $25\%$ and $10\%$ we get even better results with about thrice as good visibility as the naive approach. We see a similar trend among all the robot and car models tested. Though, we do notice that the NIFTi robot with the arm does not perform as well as the PeopleBot which is slightly taller and has 2 cameras. This could also be because the visibility poses from the NIFTi robot with the movable arm are distributed along different heights. Since the PeopleBot is at a fixed height the distribution of the poses would be more uniform. Also, the visibility from the naive approaches are rather uniform in all the cases demonstrating that 5000 poses are enough for a reasonable comparison. The difference in height and the use of an additional camera would explain the much higher visibility for the Pioneer PeopleBot. In our computation of visibility measure, we only add the shared visibility of any attached cameras once. Fig. 9 shows 300 poses generated from the functional mapping workflow and the naive algo-

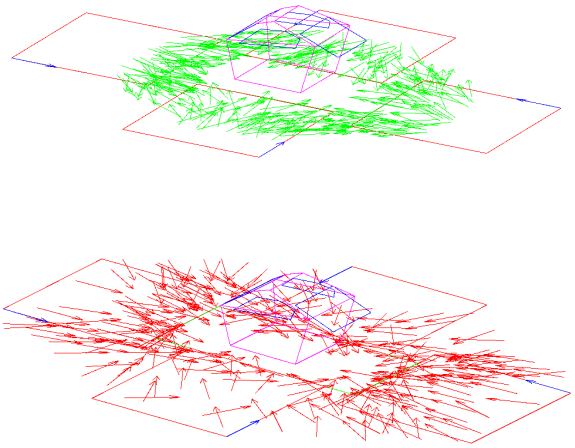

Figure 9. 300 poses generated for a test case by the (top) Functional Mapping and (bottom) Naive algorithms. The red and green arrows are the poses, pink frame in the center is the car cabin, blue polygons are windows and red polygons are the search spaces

rithm for case 1. We choose 300 as it is not as crowded as 5000 poses and the directionality of the generated poses of the functional mapping approach are clear and evident.

## 6 Conclusions

We demonstrated a method for the interaction of a robot with 3D landmarks in a search and rescue environment, based upon ontological knowledge, both pre-existing and additionally computed, as an aid to collaborative efforts by human-robot rescue teams. In particular, we analyzed the case of victim search inside crashed cars. We developed a workflow that concisely represents successful poses of looking into cars (of the order of 100s of thousands) into 200-500 3-attribute SVM vectors per opening that affords such visibility. We store these SVM vectors and the corresponding search spaces into the ontology, which is retrievable during real-time operation. The time taken to generate a successful pose from these SVM models is about 1-5 seconds which is acceptable in real-time. We performed experiments on some car models and robot configurations and found that poses thus generated by the functional mapping workflow perform far better than those by an algorithm naive of the ontological knowledge.

In the future, we plan to perform experiments with a navigating robot, with a camera on a movable arm and plan trajectories around several crashed cars that optimize the amount of visualization inside these cars. Further, we plan to extend the notion of openings and containers to other use cases e.g., entering a hole into a room of known dimensions, climbing a known stairway and so on.

## Acknowledgments

The research reported in this paper was supported by NIFTi, "Natural human-robot coordination in dynamic environments." NIFTi is funded by the European Union through its Cognitive Systems & Robotics unit, grant #247870 (Jan.2010-Dec.2013). The authors would like to thank Hendrik Zender for discussions.

## References

[1] A. Aydemir, M. Göbelbecker, A. Pronobis, K. Sjöö, and P. Jensfelt. Plan-based object search and exploration using semantic spatial knowledge in the real world. In *Proceedings of the European Conference on Mobile Robotics (ECMR 2011), Orebro, Sweden*, 2011.

[2] A. Aydemir and P. Jensfelt. Exploiting and modeling local 3d structure for predicting object locations. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[3] A. Aydemir and P. Jensfelt. What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[4] H. H. Clark. Bridging. In R.C. Schank and B.L. Nash-Webber, editors, *Theoretical Issues in Natural Language Processing*. Association for Computing Machinery, New York, NY, USA, 1975.

[5] K. R. Coventry and S. C. Garrod. *Seeing, Saying and Acting: The psychological semantics of spatial prepositions*. Psychology Press, Taylor & Francis, Hove and New York, 2004.

[6] G.J.M. Kruijff *et al*. Experience in system design for human-robot teaming in urban search & rescue. In *Proceedings of Field and Service Robotics (FSR) 2012*, 2012.

[7] M.R. Endsley. Theoretical underpinnings of situation awareness: A critical review. In M. R. Endsley and D. J. Garland, editors, *Situation awareness analysis and measurement*. Lawrence Erlbaum, 2000.

[8] T.D. Garvey. Perceptual strategies for purposive vision. Technical report, AI Center, SRI International, Menlo Park, CA, September 1976. Technical Report 117.

[9] N. Goerke and S. Braun. Building semantic annotated maps by mobile robots. In *Proceedings of TAROS*

*2009: Toward Autonomous Robotic Systems*, pages 149 – 56, 2009.

[10] M. Hanheide, N. Hawes, J. Wyatt, M. Göbelbecker, M. Brenner, K. Sjöö, A. Aydemir, P. Jensfelt, H. Zender, and G.J. Kruijff. A framework for goal generation and management. In *Proceedings of the AAAI Workshop on Goal-Directed Autonomy*, 2010.

[11] R. R. Hoffman, J. D. Lee, D. D. Woods, N. Shadbolt, J. Miller, and J. M. Bradshaw. The dynamics of trust in cyberdomains. *IEEE Intelligent Systems*, pages 5–11, November/December 2009.

[12] D. Hurych, K. Zimmermann, and T. Svoboda. Fast learnable object tracking and detection in high-resolution omnidirectional images. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, March 2011.

[13] G.A. Kaminka and I. Frenkel. Flexible teamwork in behavior-based robots. In *Proceedings of AAAI 2005*, 2005.

[14] G.A. Kaminka and I. Frenkel. Integration of coordination mechanisms in the BITE multi-robot architecture. In *Proceedings of ICRA 2007*, 2007.

[15] S. Keshavdas, H. Zender, G.J.M Kruijff, M. Liu, and F. Colas. Functional mapping: Spatial inferencing to aid human-robot rescue efforts in unstructured disaster environments. In *Proceedings of the 2012 AAAI Spring Symposium on Designing Intelligent Robots*, 2012.

[16] H. Khambhaita, G.J.M. Kruijff, M. Mancas, M. Gianni, P. Papadakis, F. Pirri, and M. Pizzoli. Help me to help you: How to learn intentions, actions and plans. In *Proc. of the AAAI Spring Symposium 2011*, March 2011.

[17] H.U. Krieger. A temporal extension of the hayes/ter horst entailment rules and an alternative to w3c's n-ary relations. In *7th International Conference on Formal Ontology in Information Systems*. IOS Press, 2012.

[18] H.U. Krieger and G.J.M. Kruijff. Combining uncertainty and description logic rule-based reasoning in situation-aware robots. In *Proceedings of the AAAI 2011 Spring Symposium "Logical Formalizations of Commonsense Reasoning"*, volume SS-11. AAAI Press, 3 2011.

[19] Timothy P. McNamara. Mental representations of spatial relations. *COGNITIVE. PSYCHOL.*, 18:87–121, 1986.

[20] Andreas Nüchter and Joachim Hertzberg. Towards semantic maps for mobile robots. *Robot. Auton. Syst.*, 56(11):915–926, November 2008.

[21] R Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans*, 30:286–297, 2000.

[22] Massimo Poesio and Renata Vieira. A corpus-based investigation of definite description use. *COMPUT. LINGUIST.*, 24(2):183–216, 1998.

[23] Ellen Prince. The ZPG letter: subjects, definiteness, and information status. In Sandra Thompson and William Mann, editors, *Discourse Description: diverse analyses of a fund raising text*, pages 295–325. John Benjamins, 1992.

[24] T. B. Sheridan and W. L. Verplank. Human and computer control of undersea teleoperators (Man-Machine Systems Laboratory Report). Technical report, Massachussets Institute of Technology, 1978.

[25] M. Tenorth and M. Beetz. Towards practical and grounded knowledge representation systems for autonomous household robots. In *Proceedings of the 1st International Workshop on Cognition for Technical Systems*, 2008.

[26] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. The MIT Press, Cambridge, MA, 2005.

[27] J. K. Tsotsos. On the relative complexity of active vs. passive visual search. *International Journal of Computer Vision*, 7(2):127–141, 1992.

[28] Shrihari Vasudevan, Stefan Gächter, Viet Nguyen, and Roland Siegwart. Cognitive maps for mobile robots-an object based approach. *Robot. Auton. Syst.*, 55(5):359–371, May 2007.

[29] J.L. Wyatt, A. Aydemir, M. Brenner, M. Hanhiede, N. Hawes, P. Jensfelt, M. Kristan, G.J.M. Kruijff, P. Lison, A. Pronobis, K. Sjöö, D. Skočaj, and A. Vrečko. Self-understanding and self-extension: a systems and representational approach. *IEEE Transactions on Autonomous Mental Development*, 2(4):282–303, 2010.

[30] H. Zender, P. Jensfelt, and G.J.M. Kruijff. Human- and situation-aware people following. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2007)*, pages 1131–1136, August 2007.

[31] Hendrik Zender, Óscar Martínez Mozos, Patric Jensfelt, Geert-Jan M. Kruijff, and Wolfram Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, June 2008.