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). 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 [17] as a map which contains in addition to metric information, assignment of mapped features (laser, vision) to entities of known classes. Further knowledge of these entities might be present in some knowledge base with an associated reasoning engine. 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. We present this information in more detail in §3.2.

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. [11, 12]): 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, 10].

(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 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 the approach in more detail, including offline- and online workflows. §4 presents the experimental setup, and first 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. Semantic mapping in the field of robotics is still curtailed largely to indoor and/or controlled environments. [17] 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 [8] and determining the type of room based on the objects found in them [21]. Such approaches use very basic semantics, unlike our approach which uses human readable ontologies based on task-specific knowledge of human beings. A precursor to our approach was [23], 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 [18], 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.

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 [22]. 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. [19]. 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 [7], an *indirect search* is suggested, where searching for one object helps restrict the search possibilities for a target object. However, Tsotsos [20] 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 [7]). 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 [23] in which we consider only ontological inference to establish functional aspects of space. In our previous paper, [13] we discussed 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. Our novelty in the current paper, is the 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 thus 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 first experimental evaluation results. The idea of deriving inferences from ontologies detailed with task-specific human knowledge comes from papers such as [9, 23]. Our approach makes a concise model of optimal positions for performing specific tasks similar to the approach [18]. Their work concerns pick and place operations in a kitchen scenario and the authors use point distribution models to reduce the dimensionality of successful poses, in order to sample from during testing. We make use of SVMs to concisely represent successful poses in functional mapping.

3 Approach

In the following subsections, we detail various aspects of our approach.

3.1 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 order to reduce the stress on the robot operator, we would like to provide any amount of autonomy that the robot can achieve, relevant to the task at hand. For example if the task was 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 teleoperation.

Our ideas stem from different *levels of autonomy* that [16] describes in a scale of autonomous human-robot collaboration. The authors explain how human-robot collaborative action can vary in degrees of autonomy as per the situation, from level 1 being complete teleoperation to level 10 being complete autonomy. We find this kind of a model to be very useful, and use it in different parts of our project as well. For example, when the vision component detects a crashed car with a low level of accuracy, it will ask the user to verify the detection, which falls under level 4 of "suggesting one alternative". In our approach to functional mapping and planning, we indicate to the human operator the poses that the robot proposes to visit and embed them in the real camera feed. The robot then executes that plan and allows the human operator the possibility of vetoing the suggestion. This falls under a high 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 human operator interface shown in Fig. 2, where a simulated rescue scenario is underway at the firefighter school (SFO) in Montelibretti, Italy.

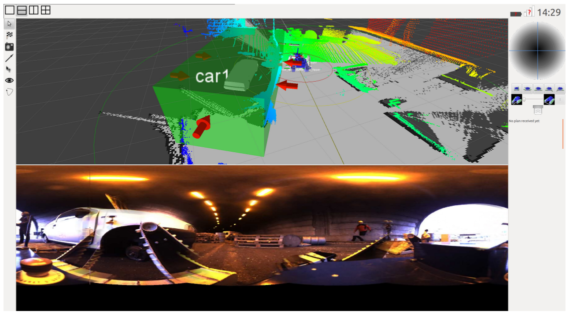

Figure 2. A screenshot of a simulated scenario with the NIFTi robot. The red arrows indicate the vantage point poses for looking into the detected car.

3.2 Ontology

Our use of semantic mapping is to attach meaningful categories to areas in the metrical map. In [24], 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. 3 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 [13], 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.

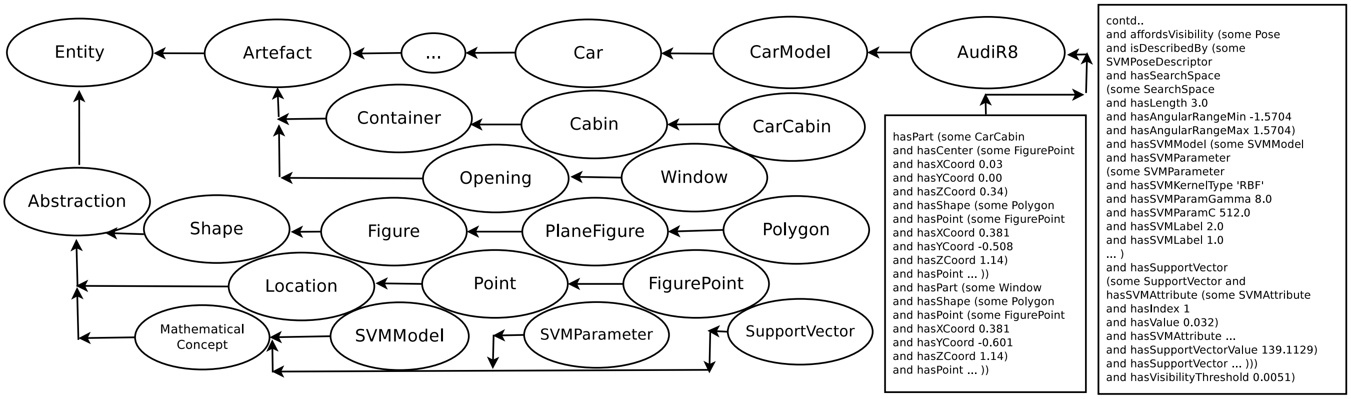

Figure 3. An excerpt of the car accident domain ontology. Details of a car with geometrical information as well as computed SVM models for visibility can be seen.

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 §3.3, we then compute information regarding the optimal "vantage points", 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 §3.4. 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. 3. 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 [14, 15]. 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.

3.3 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 [13], 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. 4, 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 detection 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 §3.1, 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. 4. 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. 4, 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

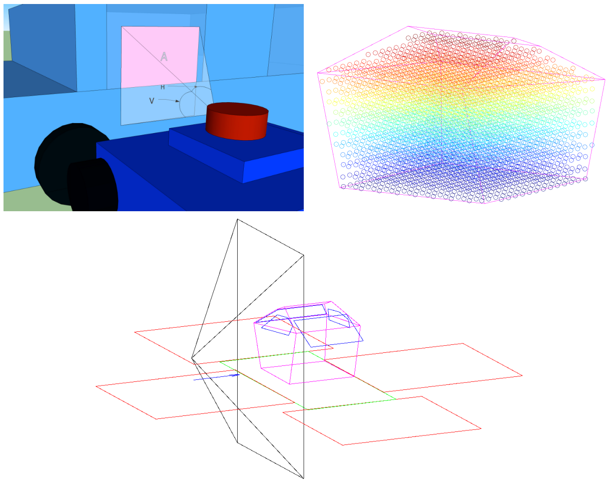

Figure 4. (clockwise from top-left) :(a) A sketch of a robot looking inside a car, the camera's visualization cone with horizontal angle H and vertical angle V is able to view an area A of the car window;(b) a car cabin(pink frame) with packing spheres (multi-coloured);(c) a sketch of a visualization 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

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.

3.4 Support Vector Machines (SVM)

We use 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(θ). In Fig. 4, we can see car windows and *search spaces* corresponding to each of them. Using linear 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. For each of these poses, we then compute the measure of visibility(*vismes*) as described in §3.3.

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 §4. We choose a specific threshold thus classifying the set *S* into 2 classes. We choose the best RBF kernel parameters (*c* and γ) 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.

Algorithm 1 Computing visibility of car cabin from single robot pose

```
 1: $n_{sph} \leftarrow 0$
 2: $N_{tot} \leftarrow |S|$
 3: for $h \leftarrow 0, H$ do
 4:   for $v \leftarrow 0, V$ do
 5:     for all $s_c \in S$ do
 6:       if $dist(r_{hv}, s_c) < r_{sph}$ then
 7:         if $proj(ray(p_{cam}, s_c), r_{hv}) < R$ then
 8:           for all $win \in W$ do
 9:             if $dist(int(r_{hv}, win), p_{cam}) < R$ then
10:               $n_{sph} \leftarrow n_{sph} + 1$
11:               $S \leftarrow S - s_c$
12:             end if
13:           end for
14:         end if
15:       end if
16:     end for
17:   end for
18: end for
19: $vismes \leftarrow n_{sph}/N_{tot}$
```

Typically a robot (*r*) would perform these steps offline 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.

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
```

3.5 Online Workflow

Fig. 5 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. 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.

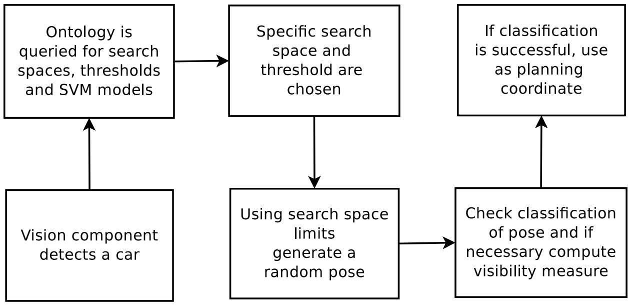
Ontology is queried for search spaces, thresholds and SVM models
Specific search space and threshold are chosen
If classification is successful, use as planning coordinate
Vision component detects a car
Using search space limits generate a random pose
Check classification of pose and if necessary compute visibility measure

Figure 5. Schematics of the online functional mapping workflow

4 Experiment

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.

Table 1 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

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%

Table 1. 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 and case 3 with the Pioneer PeopleBot and the Audi R8.

about 145 cm. 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. 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. 6 shows 300 poses generated from the functional mapping workflow and the naive algorithm 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.

5 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.

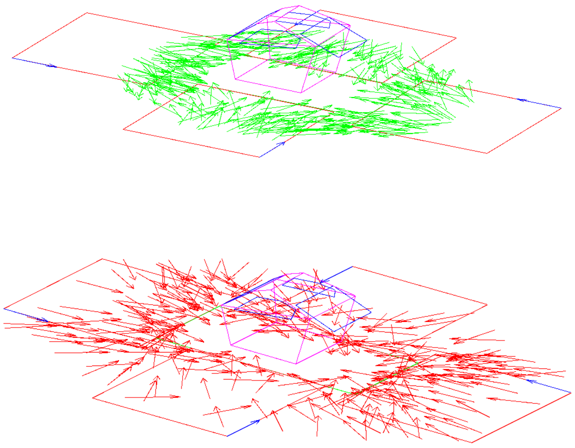

Figure 6. 300 poses generated for a test case by the (top) Functional Mapping and (bottom) Naive algorithms. The red and green arrows are the poses, pink frame in the center is the car cabin, blue polygons are windows and red polygons are the search spaces

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] C.C. Chang and C.J. Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

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

[8] 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.

[9] 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.

[10] 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.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] C. Miller and R. Parasuraman. Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human Factors*, 49:57–75, 2007.

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

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.