Integrating Active Mobile Robot Object Recognition and SLAM in Natural Environments

Staffan Ekvall, Patric Jensfelt, Danica Kragic
Computational Vision and Active Perception Laboratory and Centre for Autonomous Systems
Royal Institute of Technology, Stockholm, Sweden
ekvall, patric, danik@nada.kth.se

***Abstract*— Linking semantic and spatial information has become an important research area in robotics since, for robots interacting with humans and performing tasks in natural environments, it is of foremost importance to be able to reason beyond simple geometrical and spatial levels. In this paper, we consider this problem in a service robot scenario where a mobile robot autonomously navigates in a domestic environment, builds a map as it moves along, localizes its position in it, recognizes objects on its way and puts them in the map. The experimental evaluation is performed in a realistic setting where the main concentration is put on the synergy of object recognition and Simultaneous Localization and Mapping systems.**

I. Introduction

The importance of robotic appliances in terms of economical and sociological perspective regarding the use of robotics in domestic and office environments as well as a help to elderly and disabled has been well recognized. The AAAI Mobile Robot Challenge has demonstrated that the development of an interactive social robot represents a clear research challenge for the future. Such a robot should be able to easily navigate in dynamic and crowded environments, recognize and avoid objects and people and have a dialog with a human. It has been widely recognized that for such a system different processes have to work in synergy: high-level cognitive processes for abstract reasoning and planning, low-level sensory-motor processes for data extraction and action execution, and mid-level processes connecting these two. The coordination between these levels requires some form of representation that facilitates anchoring of different processes: one of the approaches has been the use of *cognitive maps*, [1]. The cognitive map is the body of knowledge a human or robot has about the environment. In [1], it is argued that topological, semantic and geometrical aspects are important for representation of spatial knowledge. This relates to Human-Augmented Mapping (HAM) where a human and a robot interact so to establish a correspondence between the human spatial representation of the environment and robot's autonomously learned one, [2]. These two approaches are the main motivation for our current work where the integration of object recognition and map building represents a basis for longterm reasoning and planning of a robot system. Our previous contributions related to different parts of a service robot system have been presented in [2]–[6].

A. Example Tasks and Experimental Platform

The specific problem considered in this work is a mobile robot platform that navigates autonomously in a domestic environment, builds a map as it moves along, localizes its position in it, recognizes objects on its way and puts them in the map. In our previous work, [3] we have demonstrated how robot localization and object manipulation can be performed once the robot knows an approximate position of the object *before* it is instructed to execute an object fetching task. In this paper, we are primarily concerned with a problem of how to autonomously build a map, detect object while doing this and automatically put them in the map. This then makes the basis for instructing the robot to fetch objects in a similar manner as we have demonstrated in [3]. Our experimental platform is a PowerBot from MobileRobots Inc., a non-holonomic differential drive platform with two rear caster wheels. The robot is equipped with a 6DOF robotic manipulators on the top plate. It has a SICK LMS200 laser scanner mounted low in the front, 28 Polaroid sonar sensors, a Canon VC-C4 pan-tilt-zoom CCD camera with 16x zoom on top of the laser scanner and a firewire camera on the last joint of the arm. The object recognition system presented in this work uses the Canon pan-tilt-zoom camera.

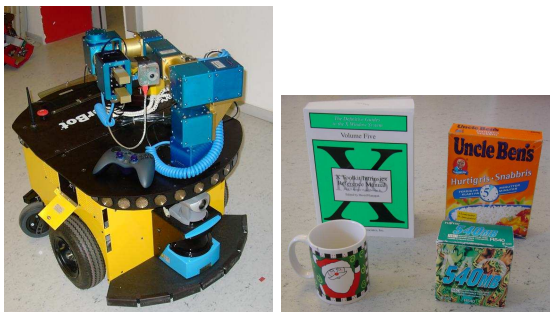

Fig. 1. left) The robot and right) Objects used in the experiments.

B. Motivation and Related Work

During the last few years, there have been a few examples of systems where the robot can acquire and facilitate semantic information, [7], [8]. Different to our approach, the work presented in [7] is mostly concentrated on linguistic interaction with a human and the robot is not using its sensors to retrieve

semantic information. The anchoring approach, presented in [8], deals mostly with the problem of integrating semantic and spatial levels where a special type of representation is used to achieve this. In this work, we are primarily interested in integration of SLAM and object recognition to acquire the semantic structure of the environment automatically and refer to our previous work regarding other aspects such as robot architecture [3], human-robot interaction [4], SLAM [5], social aspect [6] and dialog based robot instruction [2].

In service robot scenarios, we expect the robot to autonomously navigate through a home or an office and manipulate objects. For this purpose, we have developed an object recognition system that is effective based on just a few training images and also has the ability to learn incrementally as more training images are available. The vision system design is based on the *active vision* paradigm, [9] where, instead of passively observing the world, viewing conditions are actively changed so that the best results are obtained given the task at hand. The idea is to first use an appearance-based method to generate a number of hypotheses of the whereabouts of the object. The robot then investigates each of these hypotheses by moving closer to them, or as in our case, by zooming with a pan-tilt-zoom camera. Once the object appears large enough, it can be recognized with the local feature-based method. If the robot recognizes the object from two different locations, it can use geometric triangulation to calculate the approximate world position of the object and store it in the map. By augmenting the map with the location of objects, we foresee that we will be able to achieve place recognition in a longer run. Along the way by building up statistics about what type of objects are typically found in, for example, a kitchen the robot might not only be able to recognize a certain kitchen but also potentially generalize to recognize a room it has never seen before as probably being a kitchen, because of the objects found in it, [8].

II. Building a Map of the Environment

For automatic acquirement of semantic structure of the environment, automatic map building and its integration with object/place identification is a basic requirement. For increased flexibility, the robot should both be able to build a map and use it for localization. Many of the methods for SLAM (Simultaneous Localization and Mapping), including the one used in this paper, have their roots in the work by Smith et al. [10]. Much of the work in SLAM focus just on creating a map from sensor data and not on how to use the map later on. We want to use the map for tasks that require communication with the robot using common labels from the map. These labels are not only used for referring to objects, but also for certain areas and places. A natural way to achieve this is to let the robot follow the user around the environment. This allows the user to put labels on specific locations, areas or rooms.

Our SLAM algorithm uses a laser sensor and details can be found in [11]. A feature based map (e.g., 2D line map as in our case) is rather sparse and does not contain enough information for the robot to know how to move from one place to another. Furthermore, only structures that are modelled as features will be placed in the map and there is thus no explicit information about where there is free space such as in an occupancy grid based one. Here, we build a navigation graph while the robot moves around. When the robot has moved a certain distance, a node is placed in the graph at the current position of the robot. Whenever the robot moves between two nodes, these are connected in the graph. The nodes represent the free space and the edges between them encode paths that the robot can use to move from one place to another. The nodes in the navigation graph can also be used as references for certain important locations such as, for example, a recharging station. Fig. 2 shows an example of a navigation graph as connected stars. For a more detailed description of the navigation graph and how it can be used for space partitioning space, see [2].

III. Active Object Recognition

Despite the large body of work on vision based object recognition, few have investigated strategies for object recognition when the distance to the object (scale) changes significantly. Similarly, there are very few object recognition systems that have been evaluated in a mobile robot setting. In [12], a robot behavior similar to ours is presented, but with somewhat limited vision algorithms. A mobile, self-localizing robot wanders in an office environment and can learn and recognize objects encountered. However, the recognition algorithm cannot cope with a cluttered environment and it works only for a very few objects since the neural network based vision algorithm only uses object shape information as input.

The most significant drawback of the methods based on local features such as, for example, Scale Invariant Features (SIFT) [13] is that the reliable features can only be found when the object occupies a significant part of the image. It is very hard to recognize objects that are far away from the camera. We have solved this problem by both making the use of a pan-tilt-zoom camera and a global method prior to a feature based one to generate hypotheses. By zooming in on a number of a probable object locations provided by the hypotheses generation step, objects far away from the camera can be recognized. We propose to use Receptive Field Cooccurrence Histograms (RFCH) for generating hypotheses of object locations and then use a SIFT-based method for object recognition once the object is zoomed-in. RFCH [14] is an appearance-based method capable of detecting objects far away from the camera. The method itself does not outperform the SIFT based method proposed in [13] in terms of recognition rate but it is faster and thus more suitable for our scenario.

A. Object Training Procedure

For service robots, adding new objects to the database has to be easy and suitable for ordinary users. In our system, this is performed by simply showing the object to the robot - the user places the object in front of the camera and visual features are then automatically extracted from the image. During this "teaching" step, it is crucial that only features from the object are learned. If the background is visible in the image, that

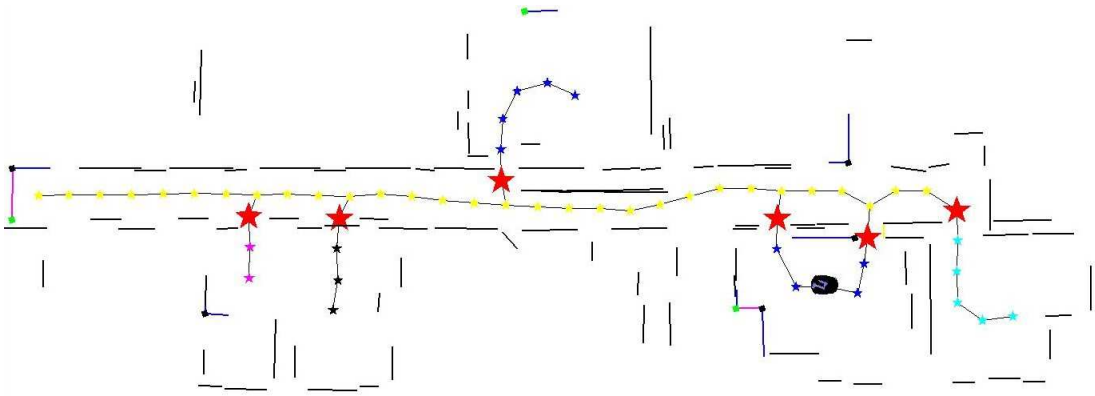

Fig. 2. A partial map of the 7th floor at CAS/CVAP. The stars are nodes in the navigation graph. The large stars denote door/gateway nodes that partition the graph into different rooms/areas.

information will be learned as well and will therefore increase the number of false positives in the online recognition stage. The common way of extracting the object is to manually process the image in an editor and crop the object from the background. However, this is a tedious step which may be difficult for a ordinary user to do. In our framework, objects are learned from human demonstrations in two steps. First, the robot is instructed to take an image of the scene with no object present in it. Then the operator places the object in front of the camera and image differencing is used to segment the object from the background. Since simple image differencing is prone to noise, a series of morphological operations is performed to achieve better segmentation (errode - dilate - errode). These operations are performed using information from the original image, i.e., a growing effect (covering holes) will not add black pixels but pixels from the original image. The result of this step can be seen in Fig. 3. The final image may still not have a perfect segmentation of the whole object but it is good enough for object recognition purpose.

Fig. 3. Left: The original image. Center: The result after image differencing. Right: The result after morphological operations

Another problem with image differencing is the choice of a threshold θ that determines if a pixel is part of the background or not. If θ is set too high, too much of the background will be kept and a too low θ will result in loosing significant parts of the object. In our work, we use an automatic adjustment of θ based on the result of the differencing performance. If image differencing was successful, the remaining pixels should be concentrated to a single area where the object has moved. If the differencing has failed, the pixels are mostly scattered around the entire image. Thus, the success is measured in terms of detection variance. In addition, a penalty that is linearly proportional to the number of remaining pixels is added to it. The reason for this is to cover the case of very few remaining pixels that have a low variance but are not sufficient for the object representation. In the run time, the algorithm tests every θ from 1 to 150 to find the optimal setting with the lowest score.

B. Hypotheses Generation

Once the object is segmented from the background, it has to be represented in a compact way for future indexing. We shortly overview the methodology and refer to our previous work for more details, [14]. We start by extracting visual features (gradient magnitude and Laplacian response) and use them to build Receptive Field Cooccurrence Histograms (RFCH). A RFCH is able to capture more of the geometric properties of an object compared to a regular histogram since, instead of just counting the descriptor responses for each pixel, the histogram is built from *pairs* of descriptor responses. The pixel pairs can be constrained based on, for example, their relative distance where only pixel pairs separated by less than a predefined distance, d_{max} are considered. Thus, the histogram represents not only how common a certain descriptor is but also how common certain combinations of descriptors are.

In the run time, the robot observes the environment and object hypotheses are generated by scanning the image with a small search window. At each step, a RFCH is built and compared to the stored RFCH of the target object using histogram intersection resulting in a vote matrix, see Fig. 4. If a vote is higher than a certain object-dependent threshold, the corresponding location for that vote is considered a hypothesis. The threshold value provides a tradeoff between search time and detection probability. If the threshold is low, many hypotheses are generated and evaluating them is time consuming. On the other hand, if the threshold is high, there is a risk that the object is missed. An extensive experimental evaluation has shown that the method is not very sensitive to the value of

the threshold, [14]. We have found that $T = 0.2$ was suitable for relatively small objects (approximately 8-10 cm in height), while the larger objects could use $T = 0.25$ for a faster search.

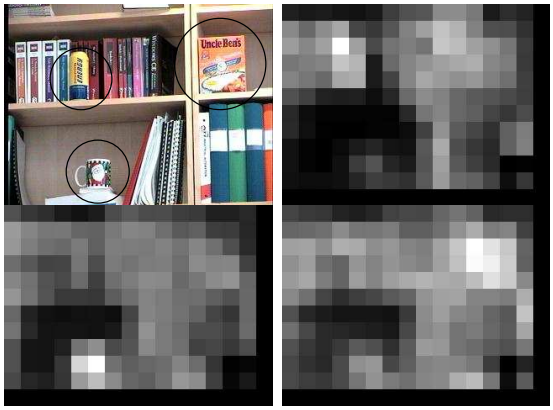

Fig. 4. Searching for a soda can, cup and rice package in a bookshelf. Light areas indicate high likelihood of the object being present. Vote matrices: Upper right - soda, lower left - cup, lower right - rice package.

C. Hypotheses Evaluation Strategy

Given a pan-tilt-zoom camera and a set of hypothesized object locations, the task is to efficiently determine and zoom on the generated hypotheses. To speed up the process, we decided to first use an intermediate level of zoom and then use the appearance-based object detector for final verification. Finding the best image locations to zoom on is not a trivial task. We quantize the view space into the same size as the vote matrix. For each vote cell, we calculate which hypotheses would still be visible if one would zoom in on that location using zoom factor z. Then, the problem is to find the smallest set of zoom locations that cover all hypotheses. Here, a simple greedy approach is followed: Select the location that covers most hypotheses, then remove these from the list and calculate the zoom locations once again. Continue until all hypotheses are covered. See Fig. 5 for an example. This approach has proven to work well and is much more efficient compared to evaluating all of the hypotheses. The method is used both for the high and the intermediate zoom levels. The zoom factor z decides how much to zoom in so to reach the next zoom level. A large z makes objects larger in the image and thus gives the detector more information but it in turn means that fewer hypotheses can be evaluated simultaneously. To account for this, we set z based on the distance measured by the laser scanner in the direction of the object. If the distance is small, the far zoom level is skipped, and the algorithm starts at the intermediate level. If the distance is very small, about 1 m or less, the intermediate level is also skipped. Experimental evaluation will show that the object recognition method works well even if the object appears larger in the image compared to training images.

Once a hypothesis is zoomed in, we again use RFCH for matching. If the match value exceeds the threshold, we perform SIFT-matching to verify the hypothesis. The more SIFT-matches found in an image, the more likely it is that the image contains the object. If the number of matches exceeds an object-dependent threshold, the object is considered recognized. Some objects have more features than others and are thus easier to recognize. To minimize the number of false positives, the threshold depends on the number of features found during training. If multiple objects are being searched for, the hypotheses for each object may be combined at each zoom level. This way, the number of zoom-in steps can be reduced, compared to searching for the objects in sequence. For each zoom-in operation, only those objects that generated the visible hypotheses are considered.

Fig. 5. An example of the greedy search strategy used while searching for the cup at the intermediate zoom level. Squares represent possible object locations, and crosses are the calculated zoom locations that cover all the hypotheses. There are 4 zoom locations and 30 hypotheses in this example.

IV. Integrating SLAM and Object Recognition

As pointed out in [15], [16] and others, robot architecture design and modules such as navigation, localization, vision based object and person recognition, speech recognition and dialog processing are just some of the key research problems that have to be considered in a development of a service robot. In our previous work, we have already demonstrated some of these. In [4], we present an interactive interface for a service robot based on multi sensor fusion where speech, vision and laser range data are integrated and show the benefit of sensory integration in the design of a robust and natural interaction system using a set of simple perceptual algorithms. In [6], we deal with the problem of embodied interaction between a service robot and a human where a control strategy based on human spatial behavior that adopts human-robot interaction patterns similar to those used in person-person encounters was studied. In [2], we concentrate on a dialog based interaction for resolution of ambiguities in Human-Augmented Mapping with special focus on spatial organization and localization. Systems integration have previously been demonstrated in [3].

In the current scenario, we focus on the integration of SLAM and object recognition modules. Here, the robot follows the user through a new environment so that the user can show the robot around. The robot is considered to be

our *guest* that is getting a tour of the environment. The user can attach labels to areas/room, i.e. instruct the robot that *this* is the living room, *this* is the kitchen, [2]. These labels can then be associated with a part of the navigation graph. As the object recognition is moderately fast, we let the robot add objects to the map after the user has shown it the extent of the environment. This is then carried out fully autonomously. Some objects can be detected from more than one position. This allows for triangulation to estimate not only the bearing to the object but also the approximate position. Even though an object has only been detected once, the map contains information from where each object has been detected and in what direction.

V. Experimental Evaluation

We have evaluated the effectiveness of our system in an office environment. Here, the robot was presented four objects (see Fig. 1, left) which were then placed in a room in six different positions. In the training stage, the robot was given two views of each object: one close-up view for SIFT-training and one at a smaller scale for RFCH-training. For this specific experiment, the navigation graph was limited to four nodes in a single room. At each node, the robot performed a search for the objects. This search was done with two different robot rotations, separated by 180°. Four different pan angles for the camera were used to cover the field of view for each of the two robot orientations.

For each object, we measured the average time for detection (ADT), average total search time (ATST) and the number of detections. Not all object locations were visible from all node positions so we counted the number of times the robot missed the object completely, i.e. did not see it from any of the node positions. As seen in Table I, this only happened three times. The rice package and book were the easiest to detect, which can be seen from the average time for detection. This is not because of their appearance but rather due to the size: the rice package and book are both large, so their features are easier to detect when the object is far away. To spot the zip-packet or the cup, the robot usually had to be less than 3 m away from the object. We found that setting the SIFT-threshold to $1/20$ of the number of features found during training, a high detection rate and no false positives were achieved. The main reason for the required recognition time and failure is that the camera sometimes needed several seconds to focus with the result that the images were blurry, causing the robot to miss the objects. In Fig. 6, the map of the room is shown with one of the object configurations with all four objects detected. Since all objects were detected from several nodes, we were able to estimate their positions.

TABLE I
Object Recognition Results

Object	ATD (min)	ATST (min)	Detect
Rice	1:10	3:52	6/6
Book	0:40	2:55	6/6
Cup	3:52	11:22	5/6
Zip-disks	3:28	7:01	4/6

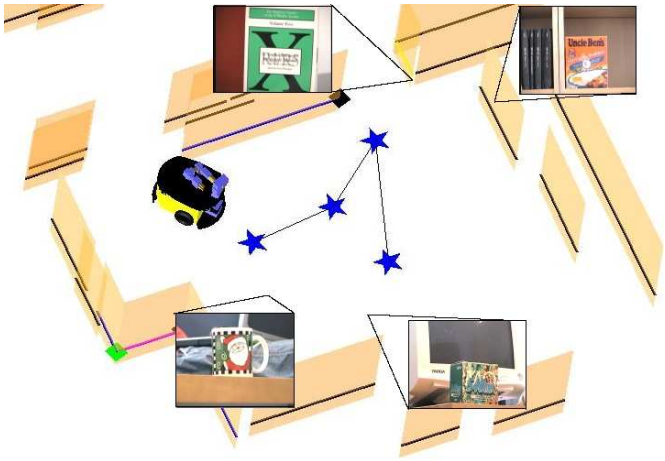

Fig. 6. The results of a robot detecting four objects in a living room and estimating their positions.

A. Experiment 2: Searching in Several Rooms

In this experiment, the search for objects is not limited to a single room but to nodes generated based on the map that are not door nodes or directly adjacent to doors. We used only two objects in this experiment and Fig. 7 shows the situation after the robot has visited two of the rooms. One instance of each of the objects were placed in these two rooms. The lines extending from close to the graph nodes starts in the camera position at the time of detection and is directed toward the observation of the object. As can be seen the objects are often spotted from more than one location. A rice package showed to the far right in the map was placed on a table. It has been detected three times and it can be seen that a triangulation would place the object closer to the camera than the laser which is only able to detect the distance to the wall behind the table. In this figure, it can also be seen that the robot has correctly detected the three doors in this part of the environment (marked as large stars), two of them leading to the same room from the corridor. This demonstrates the

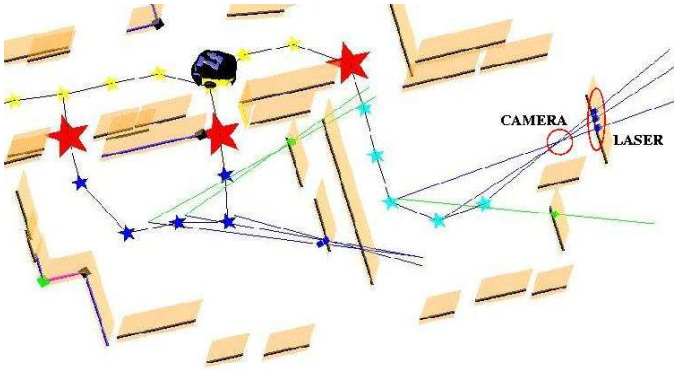

Fig. 7. Searching for two objects in two rooms: a rice package placed on a dinner table is precisely localized by camera compared to laser.

Fig. 8. The robot is instructed to fetch the rice package from the living room. It first plans a path and then follows it while avoiding obstacles.

possibility of instructing the robot to fetch object X in room Y, a task presented in the next section.

B. Experiment 3: Fetching an Object

We shortly demonstrate here how the robot may use the acquired world knowledge. After the map is built and the robot has performed a multi-room search, it is instructed to go to a specific room and pick up an object. The initial position of the robot is the room shown on the right in Fig. 7 called *the manipulator lab* and the robot has to fetch the rice package from the living room (the left room on the map). The robot first plans a path using the navigation nodes and starts moving through the door to the hallway. It then enters the living room and moves to the closest point from which it has previously seen the object. At this point, it verifies that the object is still there. Then, it raises its arm towards the object, signaling that the object has been found. If the object was not found at the expected location, a new search is initiated. A few example images taken during robot task execution are shown in Fig. 8.

VI. Conclusions

In this paper, we have presented our current efforts toward integrating spatial and semantic information in a service robot scenario that allows the robot to reason beyond simple geometrical level. At this stage, we are primarily interested in using different learning techniques to acquire semantic structure of the environment automatically. The approach taken is the integration of SLAM and object recognition systems where the map of the environment, built automatically during navigation, is augmented by detecting objects in it and then using this augmented map to perform fetching tasks. We have also presented a method for active object recognition which integrates both local and global information about the object. Finally, we have presented some initial results on how we augment our map with information about where objects are. Our current work deals with using this information in object fetch-and-carry tasks. We believe that, in a longer run, the proposed methodology will allow us to determine what objects are typically found in certain types of rooms thus facilitating the recognition of rooms' functionality.

Acknowledgement

This work has been supported by EU through the project PACO-PLUS, FP6-2004-IST-4-27657 and Swedish Research Council.

References

[1] B.J.Kuipers, "The cognitive map: Could it have been any other way?," *In H. L. Pick, Jr. and L. P. Acredolo (Eds.), Spatial Orientation: Theory, Research, and Application, New York: Plenum Press,*, pp. 345–359, 1983.

[2] G.-J. M. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen, "Clarification dialogues in human-augmented mapping," in *Proc. of the 1st Annual Conference on Human-Robot Interaction (HRI'06)*, (Salt Lake City, UT), Mar. 2006.

[3] L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic, and H. I. Christensen, "Systems integration for real-world manipulation tasks," in *IEEE International Conference on Robotics and Automation, ICRA 2002*, vol. 3, pp. 2500 – 2505, 2002.

[4] E. A. Topp, D. Kragic, P. Jensfelt, and H. I. Christensen, "An interactive interface for service robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04)*, (New Orleans), pp. 3469–3475, Apr. 2004.

[5] P. Jensfelt, J. Folkesson, D. Kragic, and H. I. Christensen, "Exploiting distinguishable image features in robotic mapping and localization," in *1st European Robotics Symposium (EUROS-06)* (H. I. Christensen, ed.), (Palermo, Italy), Mar. 2006.

[6] E. Pacchierotti, H. Christensen, and P. Jensfelt, "Embodied social interaction in hallway settings: a user study," in *IEEE Workshop on Robot and Human Interactive Communication (ROMAN)*, (Nashville, TN), pp. 164–171, Aug. 2005.

[7] C. Theobalt, J. Bos, T. Chapman, A. Espinosa, M. Fraser, G. Hayes, E. Klein, T. Oka, and R. Reeve, "Talking to godot: Dialogue with a mobile robot," in *In IEEE International Conference on Intelligent Robots and Systems, IROS'02*, pp. 1338–1343, 2002.

[8] G. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernndez-Madrigal, and J. Gonzlez, "Multi-hierarchical semantic maps for mobile robotics," in *IROS*, 2005.

[9] D. H. Ballard, "Animate vision," *Artificial Intelligence*, vol. 48, no. 1, pp. 57–86, 1991.

[10] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *4th International Symposium on Robotics Research*, 1987.

[11] J. Folkesson, P. Jensfelt, and H. Christensen, "Vision slam in the measurement subspace," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'05)*, Apr. 2005.

[12] A. Gopalakrishnan and A. Sekmen, "Vision-based mobile robot learning an navigation," in *RO-MAN*, 2005.

[13] D. Lowe, *Perceptual Organisation and Visual Recognition*. Robotics: Vision, Manipulation and Sensors, Dordrecht, NL: Kluwer Academic Publishers, 1985. ISBN 0-89838-172-X.

[14] S. Ekvall and D. Kragic, "Receptive field cooccurrence histograms for object detection," in *IEEE International Conference on Intelligent Robots and Systems*, 2005.

[15] F. Michaud, Y. Brosseau, C. Cote, D. Letourneau, P. Moisan, A. Ponchon, C. Raievsky, J.-M. Valin, E. Beaudryy, and F. Kabanza, "Modularity and integration in the design of a socially interactive robot," in *IEEE International Workshop on Robot and Human Interactive Communication, 2005. ROMAN*, pp. 172–177, Aug. 2005.

[16] M. Kleinehagenbrock, J. Fritsch, and G. Sagerer, "Supporting advanced interaction capabilities on a mobile robot with a flexible control system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, vol. 4, pp. 3469–3655, Oct. 2004.