Object Detection and Mapping for Service Robot Tasks

Staffan Ekvall, Danica Kragic, Patric Jensfelt

Computational Vision and Active Perception Laboratory

Centre for Autonomous Systems

Royal Institute of Technology, Stockholm, Sweden

{ekvall, danik, patric} @nada.kth.se

Abstract

The problem studied in this paper is a mobile robot that autonomously navigates in a domestic environment, builds a map as it moves along and localizes its position in it. In addition, the robot detects predefined objects, estimates their position in the environment and integrates this with the localization module to automatically put the objects in the generated map. Thus, we demonstrate one of the possible strategies for the integration of spatial and semantic knowledge in a service robot scenario where a simultaneous localization and mapping (SLAM) and object detection/recognition system work in synergy to provide a richer representation of the environment than it would be possible with either of the methods alone.

Most SLAM systems build maps that are only used for localizing the robot. Such maps are typically based on grids or different types of features such as point and lines. The novelty is the augmentation of this process with an object recognition system that detects objects in the environment and puts them in the map generated by the SLAM system. The metric map is also split into topological entities corresponding to rooms. In this way the user can command the robot to retrieve a certain object from a certain room. We present the results of map building and an extensive evaluation of the object detection algorithm performed in an indoor setting.

1 Introduction

The importance of robotic appliances both in economical and sociological perspective regarding the use of robotics in domestic environments as 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, detect as well as avoid obstacles, have a dialog with a user and manipulate objects. 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 mediating these two levels.

A successful coordination between these levels requires a well defined representation that facilitates anchoring of different processes. One of the proposed modeling approaches has been the use of *cognitive maps*, [2]. The cognitive map is the body of knowledge a human or a robot has about the environment. In [2], it is argued that topological, semantic and geometrical aspects are important for representation of spatial knowledge. This approach is closely related 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, [13]. Both of the above have strongly influenced our current work where the integration of object recognition and map building represents a basis for longterm reasoning and planning in an autonomous robot system. Some of our previous contributions related to different parts of a service robot system have been presented in [10, 13, 23–25, 32].

In this work, we concentrate on a particular problem related to an autonomous robot scenario which makes the basis for many of the tasks that involve manipulation of objects. In such a scenario, the robot has to be able to detect and recognize objects as well as estimate their pose. Object recognition is one of the main research topics in the field of computer vision. However, most methods typically assume that the object is either already segmented from the background or that it occupies a large portion of the image. In robotic applications, there is often a need for a system that can locate objects in the environment. This means that neither of the above assumptions are valid anymore since the distance to the object and thus its size in the image can vary significantly. Therefore, the robot has to be able to detect objects even when they occupy a very small part of the image. This requires a method that evaluates different parts of the image when searching for an object. This step is denoted 'object detection' in this work. Here, we use a method for object detection that is especially suitable for detecting objects in natural scenes, as it is able to cope with problems such as complex background, varying illumination and object occlusion. The proposed method uses a representation called Receptive Field Cooccurrence Histograms(RFCH) [5], where each pixel is represented by a combination of its color and response to different image filters. Thus, the cooccurrence of certain filter responses within a specific radius in the image serves as information basis for building the representation of the object. The specific goal for the object detection is an on-line learning scheme that is effective after just one training example but still has the ability to improve its performance with more time and new examples.

Both during the mapping phase and during robot task execution, object detection can be used to augment the map of the environment with objects' locations. We see several scenarios here: while the robot is building the map it will add information to the map about the location of objects. Later, the robot will be able to assist the user when s/he wants to know where a certain object X is. As object detection might be time consuming, another scenario is that the robot builds a map of the environment first and then when no tasks are scheduled for execution, it moves around in the environment and searches for objects.

The same skill can also be used when the user instructs the robot to go to a certain area to fetch object X. If the robot has seen the object before and it already has it in the map, the searching process is simplified to re-detection. By augmenting the map with the location of objects we also foresee that we will be able to achieve place recognition. This provides valuable information to the localization system as well as it greatly reduces the problem with symmetries in a simple geometric map. This would be an alternative approach to the visual place recognition presented in [25] and the laser based system in [16]. Furthermore, along the way by building up statistics about what type of objects typically can be 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.

This paper is organized as follows: we start by a short motivation and overview of related work in Section 2 and present our map building system in Section 3. In Section 4 we describe the object detection algorithm based on RFCHs and refer to its integration with SLAM the part. The object detection algorithm is then evaluated in Section 5 where we also show how we can augment our SLAM map with the location of objects. Finally, Section 6 concludes the paper and outlines avenues for our future research.

2 Motivation and Related Work

Although there exists a large body of work related to mobile robots, there are still no fully operational systems that can operate robustly and long-term in everyday environments. The current trend in development of service robots is reductionistic in the sense that the overall problem is commonly divided into manageable sub-problems.

During the last few years, there have been a few examples of systems where the robot is able to acquire and facilitate semantic information, [7, 31]. Different to our approach, the work presented in [31] 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 [7], deals mostly with the problem of integrating semantic and spatial levels where a special type of representation is used to achieve this. In the work reported here, we are primarily interested in the integration of geometric mapping 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 [24], human-robot interaction [32], SLAM [10], social aspect [23], dialog based robot interaction [13] and visual place recognition [25]. We expect our robot system to autonomously navigate through a home or an office environment and manipulate objects. Thus, object recognition has to be robust to outliers and changes commonly occurring in such a dynamic environment. The vision system design is based on the *active vision* paradigm, [1] where, instead of passively observing the world, viewing conditions are actively changed so that the best results are obtained given the task at hand. In our previous work [24], we showed how the robot can grasp objects given that the object is in front of the robot and the camera is centered on the object. With the work presented in this paper, those assumptions can be ignored. The robot is now able to autonomously find the object and thus position itself in front of the object.

In general, object recognition systems can roughly be divided into two major groups: global and local methods. Global methods capture the appearance of an object and often represent the object with a histogram over certain features extracted during the training process, e.g., a color histogram represents the distribution of object colors. In contrast, the local methods capture specific local details of objects such as small texture patches or particular features. For the robot to recognize an object, the object must appear large enough in the camera image. If the object is too small, local features cannot be extracted from it. Global appearance-based methods also fail, since the size of the object is small in relation to the background which commonly results in high number of false positives. As shown in Figure 1, if the object is too far away from the camera (left), no adequate local information can be extracted. Therefore, the main idea pursued in this work is to use a global 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. One of the contributions in this paper is that we make use of both approaches in a combined framework that let the methods complement each other.

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 its map. By augmenting the map with the location of objects we also 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, [7].

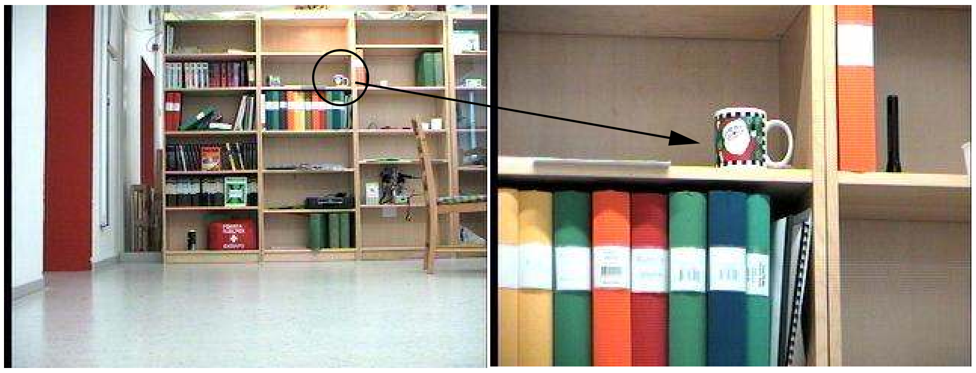


Figure 1: Left: The robot cannot recognize the cup located in the bookshelf. Right: Minimum size of the cup required for robust recognition.

2.1 Experimental Platform

The experimental platform is a PowerBot from MobileRobots Inc. It has a non-holonomic differential drive base with two rear caster wheels. The robot is equipped with a 6DOF robotic manipulator 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, see Figure 2.

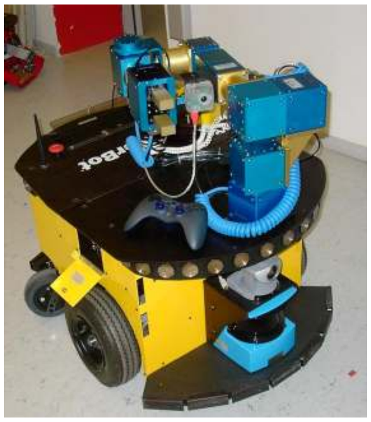


Figure 2: The experimental platform: PowerBot from MobileRobots Inc.

3 Simultaneous Localization and Mapping

One key competence for a fully autonomous mobile robot system is the ability to build a map of the environment from sensor data. It is well known that localization and map building has to be performed at the same time, which has given this subfield its name, simultaneous localization and mapping or SLAM. Many of todays algorithms for SLAM have their roots in the seminal work by Smith *et al.* [29] in which the stochastic mapping framework was presented. With laser scanners such as the ones from SICK, indoor SLAM in moderately large indoor environments is a mature technology today. In our previous work, we have focused primarily on the issue of the underlying representation of features used for SLAM [6] where it was demonstrated how maps can be built using data from a camera, a laser scanner or combinations thereof. Figure 3 shows an example of a map of an office environment where both laser and vision features are used. Line features are extracted from the laser data and can be seen as the dark lines in the outskirts of the rooms, mainly representing the walls and large furniture pieces such as shelves. In this particular example, the camera was mounted on the top of the robot with the optical axis pointing vertically so that it monitors the ceiling. From the camera images, we extract horizontal lines and circular features that correspond to the lamps placed in the ceiling. As it can be seen from the figure, the map of the environment is sparse. For environments with repetitive structures, e.g. when there is only a large hallway and most of the rooms are of the same size, such a map may be difficult to use for robot to localize itself it. Adding additional objects or specific places to the map, will add the robustness to the robot localization step as well as making the map more appropriate for interaction with humans.

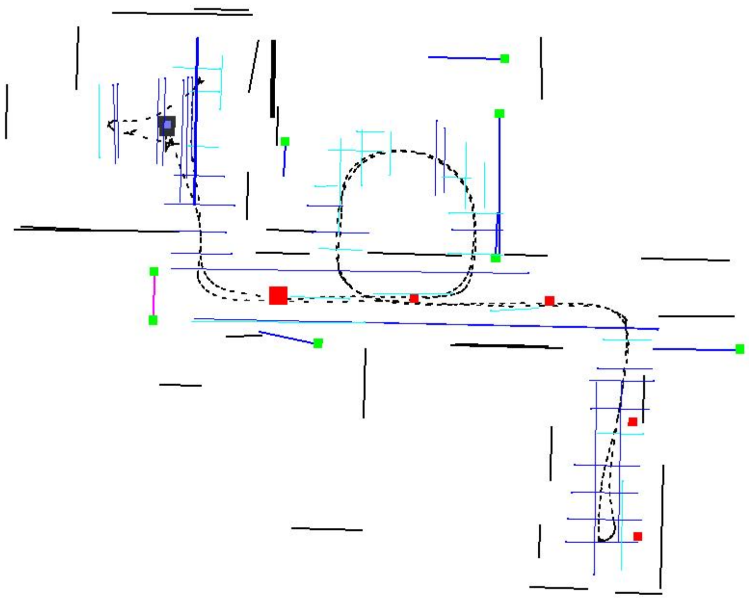


Figure 3: A partial map of the 7th floor at CAS/CVAP with both laser and vision features. The dark lines are the walls detected by the laser, the lighter ones are horizontal lines originating from the texture in the ceiling. The dark rectangular show the positions of ceiling lamps.

3.1 Building the Map

Most of the work in SLAM focus on creating a map from sensor data but not on how this data is created or how to use the map afterwards. In this paper, we want the robot to use the map to carry out different tasks. Some of these tasks may require that the user communicates with the robot using common labels from the map such as *“that room”* or *“that window”*. A natural way to achieve this is to let the robot follow the user around the environment in the initial stage when the map is being built. This allows the user to put labels on things and places of interest: certain locations, areas or rooms. This is convenient for example when instructing the robot later on to go to a certain area to fetch something or when asking the robot for information about where some thing might be.

As mentioned, our feature based map is rather sparse and does not contain enough information for the robot to know how to move from one place to another. 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. This is a well known weakness of feature based methods. In this work we use a technique inspired by [21] and 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. When the robot moves in areas where there already exist nodes close to its current position no new nodes will be created. Whenever the robot moves between two nodes they 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. Nodes can also be added to the navigation graph as references for certain important locations such as for example a recharging station. Figure 4 shows an example of a feature map with the navigation graph as connected stars.

3.2 Partitioning the Map

To be able to put a label on a certain area, the map has to be appropriately partitioned. One way to go about this is to let the user instruct the robot where the borders are between the different rooms. This may be feasible to do when the robot follows the user around. However, this is not so natural since the user can easily forget to notify the robot about it. Furthermore, not all rooms or areas are of interest for task execution.

For map partitioning, we use an automatic strategy that is based on detecting if the robot passes through a narrow opening. Whenever the robot passes a narrow opening it is hypothesized that a door is passed. This in itself will lead to some false doors in cluttered rooms. However, assuming that there are very few false negatives in the detection of doors we get great improvements by adding another simple rule. If two nodes that are thought to belong to different rooms are connected by the robot motion the door that separated them into different rooms was not a door. That is, it is not possible to reach another room without passing a door. In Figure 4 the larger stars denote doors or gateways between different areas/rooms.

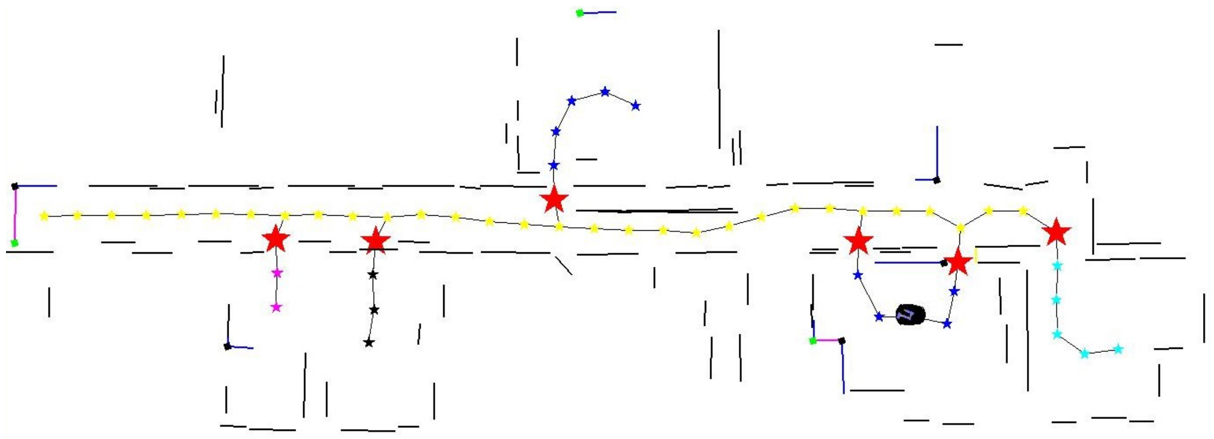


Figure 4: A partial map of the 7th floor at CAS/CVAP. The stars are nodes in the navigation graph. The large stars denote door/gateway nodes that partition the graph into different rooms/areas.

So far, we have presented how the robot is equipped with the ability to i) autonomously build the map of the environment, and ii) partition the map into different rooms. The next section describes the methodology for active object recognition and its use for augmenting the map of the environment.

4 Active Object Recognition

Object recognition algorithms are typically designed to classify objects to one of several predefined classes assuming that the segmentation of the object has already been performed. Test images commonly show a single object centered in the image and, in many cases, having a black background [19] which makes the recognition task simpler. In general, the object detection task is much harder. Its purpose is to search for a specific object in an image not even knowing before hand if the object is present in the image at all. Most of the object recognition algorithms may be used for object detection by scanning the image for the object. Regarding the computational complexity, some methods are more suitable for searching than others.

One of the early references related to our approach has been presented in [30] where it was proposed that an object should be represented by its color histogram. At the query stage, the objects are identified by matching a color histogram from an image region with a color histogram from a sample of the object using histogram intersection. It has been shown that this method is robust to changes in the orientation, scale, partial occlusion and changes of the viewing position. However, the major drawbacks of this method are its sensitivity to lighting conditions which we cope with in our method by combining the color with filter responses. This idea has been generalized by [27] through the introduction of multidimensional receptive field histograms to approximate the probability density function of local appearance. In their work, the recognition algorithm calculates probabilities for the presence of objects based on a small number of vectors of local filters at different scales.

Principal Component Analysis has also been investigated for appearance-based object recognition, [18]. This method has been found attractive mainly due to the representation of each image by a small number of coefficients, which can be stored and searched for efficiently. The main drawback of this and similar methods is that the representation is very sensitive to changes of individual pixel values, due to translation, scale changes, image plane rotation or light changes.

In [28] another method based on local characteristics has been developed and evaluated for object recognition in the case of partial visibility, image transformations and complex scenes. This approach is based on the combination of differential invariants computed at key points with a robust voting algorithm combined with some additional local constraints. The recognition is based on the computation of the similarity between the representation vectors for the key points. Many of the current work on object recognition considers different variants of Support Vector Machines, [26].

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 [9], 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 with very few objects since the neural network based vision algorithm only uses object shape information as input.

The idea of generating hypotheses and then zooming on them in the verification step to provide richer information has been used before, [11]. The authors use regular color histograms which only works for relatively simple objects and requires many training images. In the experimental section, we demonstrate that our object detection/recognition system is more powerful as it can learn to recognize a wide range of objects and find them in the environment even when there are object of similar colors in the field of view. In addition, our approach generates very few false positives due to the choice of combining both appearance and geometrical information.

Impressive recognition results have been achieved with methods based on Scale Invariant Features (SIFT) [14] and alike. The most significant drawback of these methods is that 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 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 RFCHs for generating hypotheses of object locations, and then use a SIFT-based method for object recognition once the object is zoomed-in. RFCH [5] 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 [14] in terms of recognition rate, but it is faster and more robust to effects of overall rotation and scale which is an important requirement in a service robot scenario.

4.1 Object Segmentation

For the robot to put a new object into the database, we have adopted a very natural approach with an ordinary end-user in mind: the object is shown to the robot by placing it in front of the camera. During this "teaching" step, features are extracted from the image and it is crucial that only features from the object are extracted and thus learned by the classifier. If the background is visible in the image, that information will be learned as well and will therefore increase the number of false positives in the online recognition stage. The common way of segmenting 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 regular user to perform. In our framework, training images are are generated from human demonstrations in two steps. First, the robot captures an image of the scene without the object being present in it. Then the operator places an object in front of the camera and the object is separated using image differencing. Thus, the user is relieved of the process of manually extracting the object. However, simple image differencing is prone to noise and the result is not a perfect image of the object as it contains many holes in the object and also some of the background. To cope with this problem, a number of morphological operations is performed to achieve better segmentation (errode - dilate - errode, [8]). 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 Figure 5. The final image may still not have a perfect segmentation of the whole object but this is not a problem for any of the vision algorithms we apply as long as most of the object is visible.

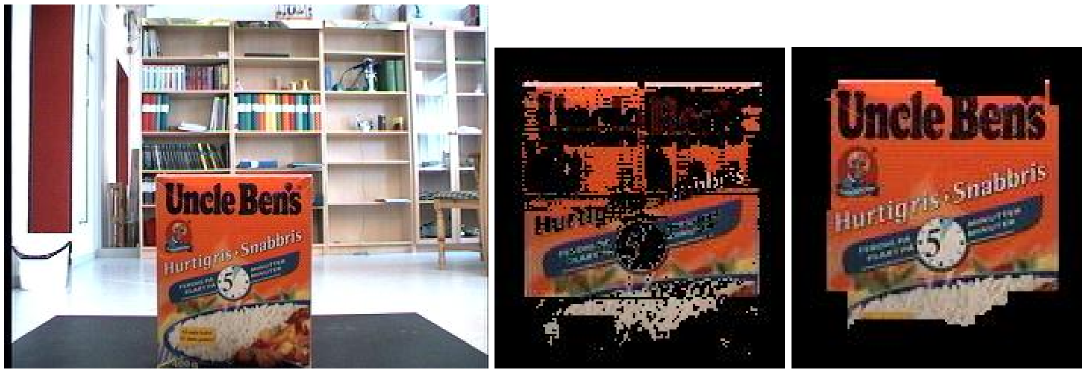


Figure 5: Left: The original image. Center: The result after simple image differencing (only a part of the image is shown). Right: The result after morphological operations.

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 remain. If θ is set too low, significant parts of the object may be omitted. 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 is added that is linearly proportional to the number of pixels remaining, to cover the case of very few remaining pixels that have a low variance but are not sufficient for the object representation. The algorithm tests every θ from 1 to 150, to find the optimal setting with the lowest score.

4.1.1 Quantization and Dimensionality Reduction

Once the object is segmented, it has to be represented for indexing. For this purpose, we have developed a method based on Receptive Field Cooccurrence Histograms. A Receptive Field Histogram is a statistical representation of the occurrence of several descriptor responses within an image.

Descriptors used in our work are gradient magnitude, response of a Laplacian, [8] and normalized colors. A Receptive Field Cooccurrence Histogram (RFCH) is able to capture more of the geometric properties of an object compared to classical color histograms. 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. This way, only pixel pairs separated by less than a maximum distance, d_{max} are considered.

Thus, the histogram represents not only how common a certain descriptor response is in the image but also how common it is that certain combinations of descriptor responses occur close to each other. We briefly present the descriptors:

- **Normalized Colors**

  The color descriptors are the intensity values in the red and green color channels, in normalized RG-color space, according to $r_{norm} = \frac{r}{r+g+b}$ and $g_{norm} = \frac{g}{r+g+b}$.

- **Gradient Magnitude**

  The gradient magnitude is a differential invariant, and is described by a combination of partial derivatives (L_x, L_y): $|\nabla L| = \sqrt{L_x^2 + L_y^2}$. The partial derivatives are calculated from the scale-space representation $L = g * f$ obtained by smoothing the original image f with a Gaussian kernel g of size 7, with standard deviation $\sigma = 2$ and $\sigma = 4$.

- **Laplacian**

  The Laplacian is an on-center/off-surround descriptor. Using this descriptor is biologically motivated, as it is well known that center/surround ganglion cells exist in the human brain. The Laplacian is calculated from the partial derivatives (L_{xx}, L_{yy}) according to $\nabla^2 L = L_{xx} + L_{yy}$. From now on, $\nabla^2 L$ denotes calculating the Laplacian on the intensity channel, while $\nabla^2 L_{rg}$ denotes calculating it on the normalized color channels separately.

Regular multidimensional RFHs [27] have one dimension for each image descriptor. This makes the histograms huge. For example, using 15 bins in a 6-dimensional histogram means 15^6 ($\sim 10^7$) bin entries. As a result the histograms are very sparse, and most of the bins have zero or only one count. Building a cooccurrence histogram makes things even worse. In that case we need about 10^{14} bin entries. By first clustering the input data, a dimension reduction is achieved. Hence, by choosing the number of clusters, the histogram size may be controlled. In this work, we have used 80 clusters resulting in that our cooccurrence histograms are dense and most bins have high counts.

Dimension reduction is done using K-means clustering [15]. Each pixel is quantized to one of N cluster centers. The cluster centers have a dimensionality equal to the number of image descriptors used. For example, if both color, gradient magnitude and the Laplacian are used, the dimensionality is six (three descriptors on two colors). As distance measure, we use the Euclidean distance in the descriptor space. That is, each cluster has the shape of a sphere. This requires all input dimensions to be of the same scale, otherwise some descriptors would be favored. Thus, we scale all descriptors to the interval [0,255]. The clusters are randomly initialized, and a cluster without members is relocated just next to the cluster with the highest total distance over all its members. After a few iterations, this leads to a shared representation of that data between the two clusters. Each object ends up with its own cluster scheme in addition to the RFCH calculated on the quantized training image.

4.2 Hypotheses Generation using Receptive Field Cooccurrence Histogram

When searching for an object in a scene, the image is quantized with the same cluster-centers as the cluster scheme of the object being searched for. Object hypotheses are generated by scanning the image with a small search window. At each step, a local RFCH is built and compared to the stored RFCH of the target object using histogram intersection

$$\mu(h_1,h_2) = \sum_{m=1}^{N^2} \min(h_1[n],h_2[n]) \tag{1}$$

where $h_i[n]$ denotes the frequency of receptive field combinations in bin n for image i quantized into N cluster centers. Note here that the histogram size is N^2 since we use cooccurrence histograms. The higher the value of $\mu(h_1,h_2)$, the better the match between the histograms. Prior to matching, the histograms are normalized with the total number of pixel pairs. This results in a vote matrix and if a vote is higher than a certain object-dependent threshold, the corresponding location for that vote is considered a hypothesis. An example of this is shown in Figure 8.

Another popular histogram similarity measure is the χ^2:

$$\mu(h_1,h_2) = \sum_{n=1}^{N^2} \frac{(h_1[n]-h_2[n])^2}{h_1[n]+h_2[n]} \tag{2}$$

In this case, the lower value of $\mu(h_1,h_2)$, the better the match between the histograms. The χ^2 similarity measure usually performs better than the histogram intersection method on object recognition image databases. However, we have found that χ^2 performs much worse than histogram intersection when used for object detection. We believe that this is because the background that is visible in the search window and not present during training, is not penalizing the match correspondence as much as with the χ^2. Histogram intersection focuses on bins that represent the searched object best, while χ^2 treats all bins equally.

4.3 Hypotheses Evaluation Strategy

Given our pan-tilt-zoom camera and a set of hypothesized object locations, the task is to efficiently determine and zoom in 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 Figure 6 for an example. This approach has proven to work well and is much more efficient compared to evaluating all of the hypotheses.

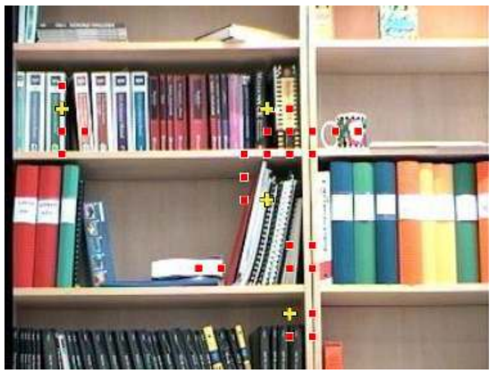


Figure 6: An example of the greedy search strategy used while searching for the cup at the intermediate zoom level. Squares represent possible object locations, and crosses are the calculated zoom locations that cover all the hypotheses. There are 4 zoom locations and 30 hypotheses in this example.

The method is used both for the high and the intermediate zoom levels. The zoom factor z decides how much to zoom in 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 a predefined threshold, we perform a simple matching based on SIFT features to verify the hypothesis, [14]. The more SIFT-matches found in a part of the image, the more likely it is that it 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.

4.4 Integrating SLAM and Object Recognition

As pointed out in [12, 17] among 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 [32], 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 [23], 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 [13], we concentrate on a dialog based interaction for resolution of ambiguities in Human-Augmented Mapping. Systems integration, object manipulation and grasping using visual servoing have previously been demonstrated in [24] and are outside the scope of this paper.

In the current scenario, we focus on the integration of SLAM and object recognition modules that will on a longer run facilitate all of the above problems. We note here that the main strength of this work is the ability of the system to provide richer maps of the environment than it would be possible to achieve with any of the techniques alone. In addition, it definitely facilitates mobile manipulation tasks where the robot is able to move from its current position to another place or room in order to fetch an object. Many of the current object manipulation systems assume that the object is already in the field of view or that it can be easily found from the robot's current position. Our main goal with the current system is to make this step as flexible as possible and allow for a more autonomous mobile object manipulation. Including object into the map also serve to address another important problem in SLAM, namely, loop closing. In [20], a laser range scanner is used to build a map and vision based Maximally Stable Extremal Regions (MSER) are used to detect loop closing situations. The objects in our map could be used in a similar way.

In the current system, the robot first follows the user through a new environment so that the user can show the robot around. The robot is considered to be a *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, [13]. 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 which allows for re-detection. Objects are stored on an area by area basis. We should stress that we do not propose to try to estimate the exact position of an object here. When the time comes to interact with the object, for example to pick it up, it has to be re-detected anyway to confirm that it is still there and then use visual servoing techniques to pick it up using the same technique that has been proposed in [24].

5 Experimental Evaluation

First, we present evaluation of the object detection system. The proposed technique is intended for use primarily in a service robot scenario and the experimental evaluation is strongly motivated by this. We believe that evaluating the method on some of the computer vision databases is not equally relevant for the considered scenario but it will be considered in our future work. Again, the main reason is that the design of the currently available databases has not been governed by a service robot application in mind and it does not include the variability that usually arises in such applications. In addition, the evaluation on databases does not capture the problem of actively searching for objects.

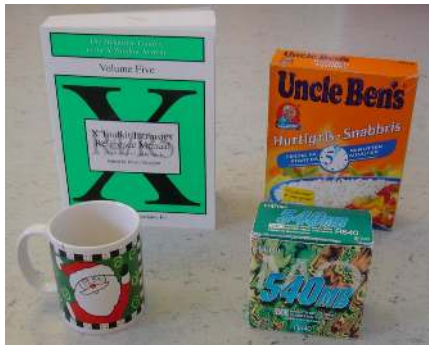


Figure 7: The four objects tested in the SLAM scenario.

5.1 Evaluation of Object Detection Using RFCHs

In case of object detection, the object occupies usually only a small area of the image and classical object recognition algorithms cannot be used directly since the entire image is considered. Instead, the image is scanned using a small search window. The window is shifted such that consecutive windows overlap to 50 % and the RFCH of the window is compared with the object's RFCH according to (1). Each object may be represented by several histograms if its appearance changes significantly with the view angle of the object. However, in this work we only used one histogram per object.

The matching vote $\mu(h_{object}, h_{window})$ indicates the likelihood that the window contains the object. Once the entire image has been searched through, a vote matrix provides hypotheses of the object's location. Figure 8 shows a typical scene from our experiments together with the corresponding vote matrix for the yellow soda can. The vote matrix reveals a strong response in the vicinity of the object's true position close to the center of the image. The vote matrix may then be used to segment the object from the background, as described in [4], or just provide an hypothesis of the object's location. The most probable location corresponds to the vote cell with the maximum value.

The running time can be divided into three parts. First, the test image is quantized. The quantization time grows linearly with the number of clusters, N. Second, the histograms are calculated and this calculation time grows with the square of pixel distances, d_{max}. Last, the histogram similarities are calculated. Although histogram matching is a fast process, its running time grows with the square of N. The algorithm is very fast which makes it applicable even on mobile robots. Depending on the number of descriptors used and the image size, the algorithm implemented in C++ runs at about 3-10 Hz on a 3 GHz regular PC.

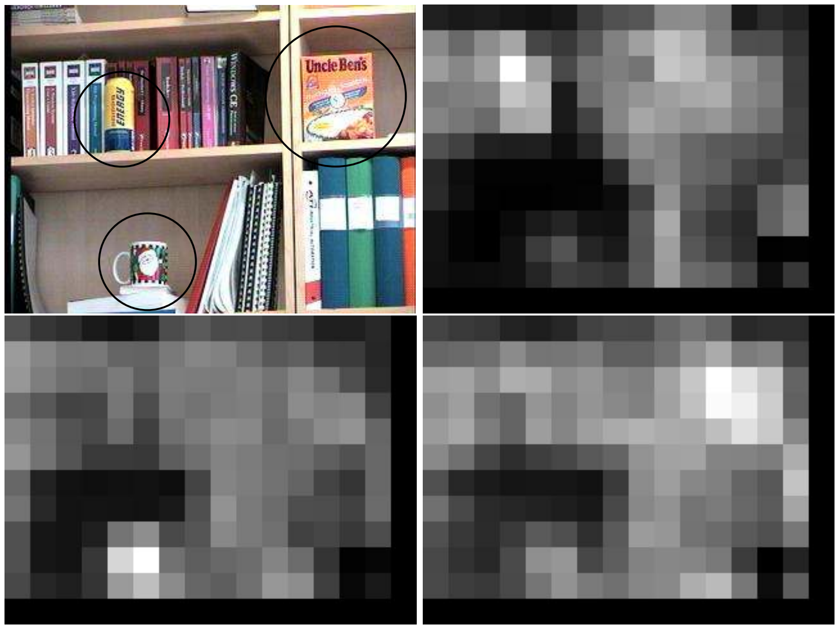


Figure 8: Searching for a soda can, cup and rice package in a bookshelf. Light areas indicate high likelihood of the object being present. Vote matrices: Upper right - soda, lower left - cup, lower right - rice package.

In this section, we evaluate six different descriptor combinations. The descriptor combinations are chosen to show the effect of the individual descriptors as well as the combined performance. The descriptor combinations are:

- 2D: $[R,G]$ - Capturing only the absolute values of the normalized red and green channel. Corresponding to a color cooccurrence histogram. With $d_{max} = 0$ this means a normal color histogram (except that the colors are clustered).

- 4D: $[R,G,\ \nabla^2 L_{rg}\ \sigma = 2]$ - The above combination extended with the Laplacian operator at scale $\sigma = 2$. As the operator works on both color channels independently, this combination has dimension 4.

- 5D: $[R,G,\ |\nabla L_{rg}|,\ \nabla^2 L_{rg}\ \sigma = 2]$ - The above combination extended with the gradient magnitude information on each color channel, scale $\sigma = 2$.

- 5D: $[|\nabla L|\ \sigma = 1,2,4, |\nabla L_{rg}|\ \sigma = 2]$ - Only the gradient magnitude, on the intensity channel and on each color channel individually. On the intensity channel, three scales are used, $\sigma = 1,2,4$. For each of the color channels, scale $\sigma = 2$ is used. 5 dimensions.

- 6D: $[\nabla^2 L\ \sigma = 1,2,4, \nabla^2 L_{rg}\ \sigma = 2]$ - The same combination as above, but for the Laplacian operator instead.
- 10D: $[R, G,\ |\nabla L_{rg}|,\ \nabla^2 L_{rg}\ \sigma = 2,4]$ - The combination of colors, gradient magnitude and the Laplacian, on two different scales, $\sigma = 2,4$. 10 dimensions.

All descriptor combinations were evaluated using CODID - CVAP Object Detection Image Database, [3].

5.1.1 CODID - CVAP Object Detection Image Database

CODID is an image database designed specifically for testing object detection algorithms in a natural environment. The database contains 40 test images of size 320x240 pixels, and each image contains 14 objects. The test images include problems such as object occlusion, varying illumination and textured background. Out of the 14 objects, 10 are to be detected by an object detection algorithm. The database provides 10 training images for this purpose, i.e. only one training image per object. The database also provides bounding boxes for each of the ten objects and each scene and an object is considered to be detected if the algorithm can provide pixel coordinates within the object's bounding box for that scene. In general, detection algorithms may provide several hypotheses of an object's location. In this work, only the strongest hypothesis is taken into account. The test images are very hard from a computer vision point of view, with cluttered scenes and objects lying rotated behind and on top of each other. Thus, many objects are partially occluded in the scene. In total, the objects are arranged in 20 different ways and each scene is captured under two lighting conditions. The first lighting condition is the same as during training, a fluorescent ceiling lamp, while the second is a closer placed light bulb illuminating from a different angle.

5.1.2 Training

For training, one image of each object is provided. Naturally, providing more images would improve the recognition rate but our main interest is to evaluate the proposed method using just one training image. The training images are shown in Figure 9. As it can be seen, some objects are very similar to each other, making the recognition task non-trivial. The histogram is built only from non-black pixels. In these experiments, the training images have been manually segmented. For training in robotic applications, we assume that the robot can observe the scene before and after the object is placed in front of the camera and perform the segmentation based on image differencing as described in Section 4.1.

5.1.3 Detection Results

The experimental evaluation has been performed using six combinations of feature descriptors. As can be seen in Table 1, the combination of all feature descriptors gives the best results. The color descriptor is sensitive to

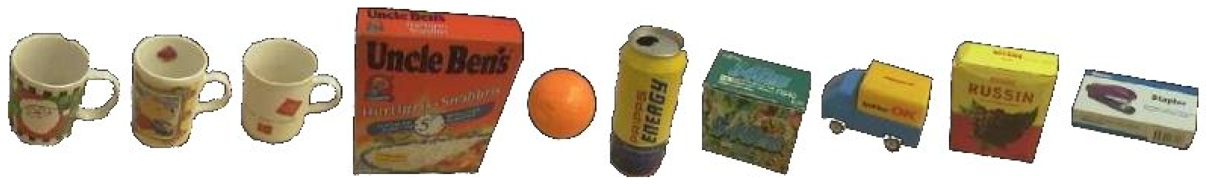


Figure 9: There are currently ten objects in the database.

changing lighting conditions, despite the fact that the images were color normalized prior to recognition. On the other hand, the other descriptors are robust with respect to this problem and the combination of descriptors performs very well. We also compare the method with regular color histograms which show much worse results.

Adding descriptors on several scales does not seem to improve the performance in the first case, but when lighting conditions change, some improvement can be seen. With changed illumination, colors are less reliable and the method is able to benefit from the extra information given by the Laplace and gradient magnitude descriptors on several scales. All descriptor combinations have been tested with $N = 80$ cluster-centers, except the 10-dimensional one which required 130 cluster-centers to perform optimally. Also, $d_{max} = 10$ was used in all tests, except for the color histogram method, which of course use $d_{max} = 0$.

All detection rates reported in Table 1 are achieved using the histogram intersection method (1). For comparison, we also tested the 6D descriptor combination with the χ^2 method (2). With this method, only 60 % of the objects were detected, compared to 95 % using histogram intersection.

Table 1: The detection rate of different feature descriptor combinations with the histogram intersection method (1) in two cases: i) same lighting conditions as when training, and ii) changed lighting conditions.

	Lighting Conditions:	
Descriptor Combination:	Same	Changed
2D: Color histogram	71.5	38.0
2D: $[R, G]$ (CCH)	77.5	38.0
4D: $[R, G,\ \nabla^2 L_{rg}\ \sigma = 2]$	88.5	61.5
5D: $[\lvert\nabla L\rvert\ \sigma = 1,2,4, \lvert\nabla L_{rg}\rvert\ \sigma = 2]$	57	51
5D: $[\nabla^2 L\ \sigma = 1,2,4, \nabla^2 L_{rg}\ \sigma = 2]$	77.5	62.0
6D: $[R, G,\ \lvert\nabla L_{rg}\rvert,\ \nabla^2 L_{rg}\ \sigma = 2]$	95.0	80.0
10D: $[R, G,\ \lvert\nabla L_{rg}\rvert,\ \nabla^2 L_{rg}\ \sigma = 2,4]$	93.5	86.0

5.1.4 Free parameters

The algorithm requires setting a number of parameters which were experimentally determined. However, it is shown that the detection result are not significantly affected by the values of parameters. The parameters are:

- **Number of cluster-centers, *N***

  We found that using too few cluster-centers reduces the detection rate. From Figure 10 it can be seen that feature descriptor combinations with high dimensionality require more cluster centers to reach their optimal performance. As seen, 80 clusters is sufficient for most descriptor combinations.

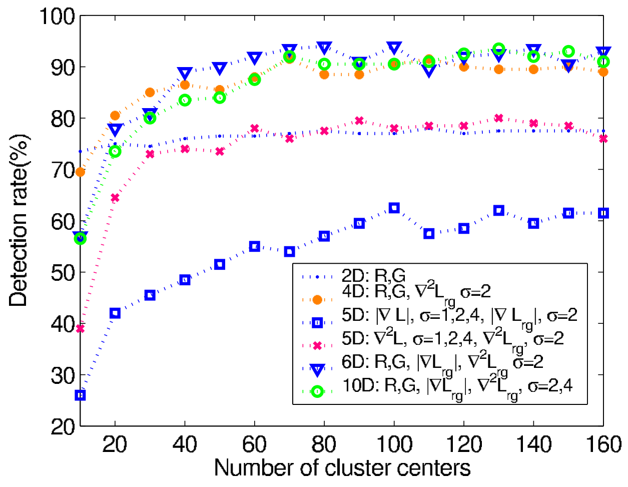


Figure 10: The importance of the number of cluster-centers for different image descriptor combinations.

- **Maximum pixel distance, d_{max}**

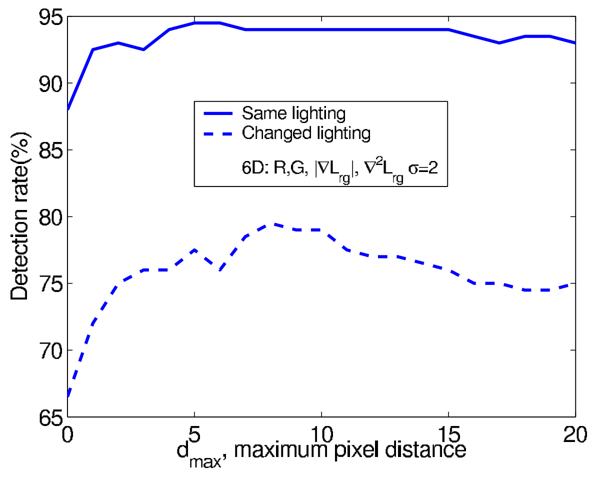
  The effect of cooccurrence information is evaluated by varying d_{max}. Using $d_{max} = 0$ means no cooccurrence information. As seen in Figure 11, the performance is increased radically by just adding the cooccurrence information of pixel neighbors, $d_{max} = 1$. For $d_{max} > 10$ the detection rate starts to decrease. This can be explained by the fact that the distance between the pixels is not stored in the RFCH. Using a too large maximum pixel distance will add more noise than information, as the likelihood of observing the same cooccurrence in another image decreases with pixel distance. As seen in Figure 11, the effect of the cooccurrence information is even more significant when lighting conditions change.

- **Size of cluster-centers, α**

  We have investigated the effect of limiting the size of the cluster centers. Pixels that lie outside all of the cluster centers are classified as background and not taken into account. As seen in Figure 11, the algorithm performs optimally when $\alpha = 1.5$, i.e. the size is 1.5 times the average distance to the cluster center used during training. Smaller α removes too many of the pixels and, as α grows, the effect described in Section 4.1.1 starts to decrease.

- **Search window size**

  We have found that some object recognition algorithms require a search window size of a specific size to function properly for object detection. This is a serious drawback, as the proper search window size is



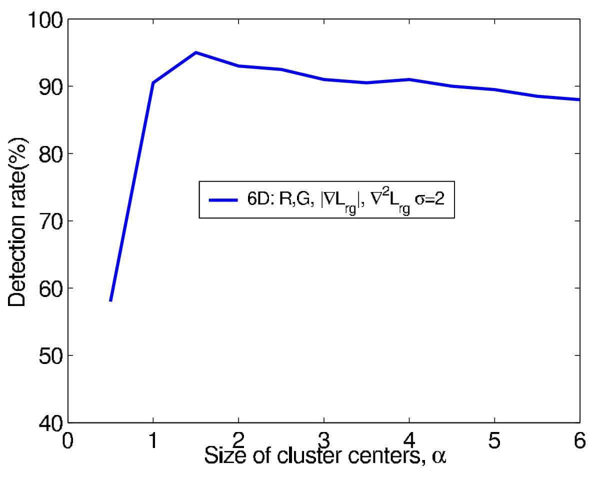
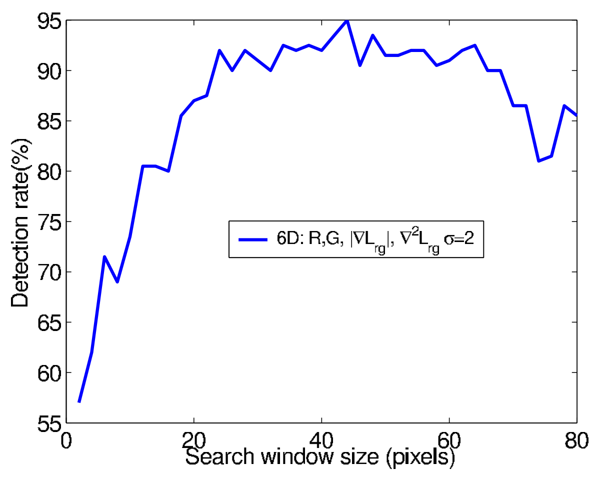


Figure 11: Left: The detection rate mapped to the maximum pixel distance used for cooccurrence, d_{max}, in two cases: Same lighting as when training, and different lighting from training. Right: The detection rate mapped to the size of the cluster centers, α.

commonly not known in advance. Searching the image several times with different sized search windows is a solution, although it is quite time consuming. As Figure 12 shows, the choice of search window size is not crucial for the performance of our algorithm. The algorithm performs equally well for window sizes of 20 to 60 pixels. In our experiments, we have used a search window of size 40.



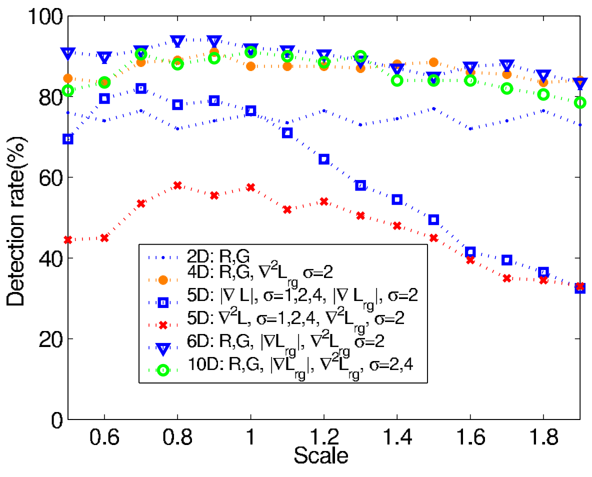


Figure 12: Left: The detection rate mapped to the size of the search window. Right: The effect on detection rate when the training examples are scaled, for six different image descriptor combinations.

5.1.5 Scale Robustness and Re-detection

We have also investigated how the different descriptor combinations performs when the scale of the training object is changed. The training images were rescaled to between half size and double size, and the effect on detection

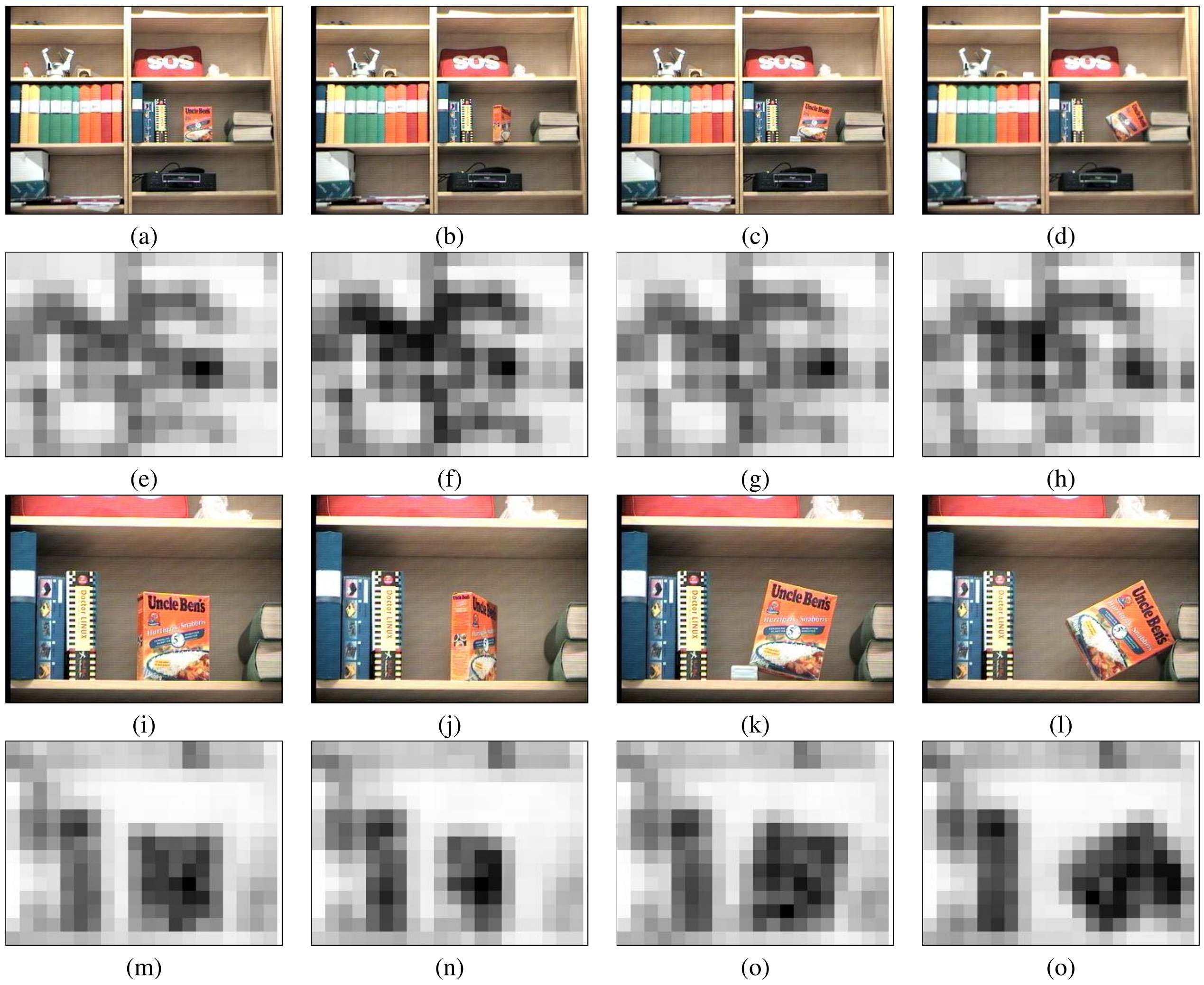


Figure 13: A few example images that demonstrate the outcome of the hypotheses generation process for cases when the rotation of the object relative to the camera changes. It is evident that the rotation around the vertical axis affects the results (figure (b) and (f)) for cases when the object is far away from the camera. A more detailed discussion is provided in the text.

performance was investigated. As seen in Figure 12, the color descriptors are robust to scaling, while the other descriptors types decrease in performance as the scale increase. However, when the descriptor types are combined, the performance is relatively robust to scale changes. To improve scale robustness, the image can be scanned at different scales.

To visualize the performance of the hypothesis generation step we have generated a few test images in which we varied both the initial distance to the object and object's rotation relative to the camera. In this scenario, the robot is searching for the rice package. The odd rows in Figure 13 show the example images, while even rows show the vote matrices used for generate the hypotheses for the attention process. Strong hypotheses are shown with darker colors. We see that the rotation of the object around the camera axis does not effect the outcome of the hypotheses generation significantly. However, the rotation around the vertical axis does effect the hypothesis generation (evident in Figure 13(b) and (f)). The reason for this is that the object is viewed from the side and the stored representation of the object uses the frontal image. However, the object is still one of the strongest hypothesis generated even if we, on purpose, placed it in an environment where there are other items with similar textural properties presents (orange folders). It can also be seen in Figure 13(j) and (n) that once the hypothesis is examined more closely, the votes are stronger and facilitate correct detection. One of the limitations of the current method is thus not so much in the hypotheses generation but in hypotheses verification since SIFT points are invariant to rotations of up to 30-35°, [14] which means that even if the correct hypothesis is generated the system may fail to detect the object when the SIFT based hypothesis generation process fails. This can be addressed by using more than one view of the objects.

5.2 Evaluating the Search Effectiveness

We started by testing the effectiveness of our system in an office environment. Here, the robot was presented four objects (see Figure 7) which were then placed in a room in six different configurations. In the training stage, the robot was given two views of each object: one close-up view for SIFT-training and one at a lower scale for RFCH-training. We note here that still a single image is used for the object, just at different scales.

The robot performed a search for the objects at each node in the navigation graph. 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. In this specific experiment, the navigation graph consisted of four nodes in a single room (see Figure 14). We note here that this strategy is not optimal and that there is room for improvement by, for example, considering some of the work on view planning, [22].

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 2, 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 after moving the camera or the robot, with the result that the images were blurry, causing the robot to miss the objects. In Figure 14, 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, their position estimates can be improved using triangulation.

Table 2: 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

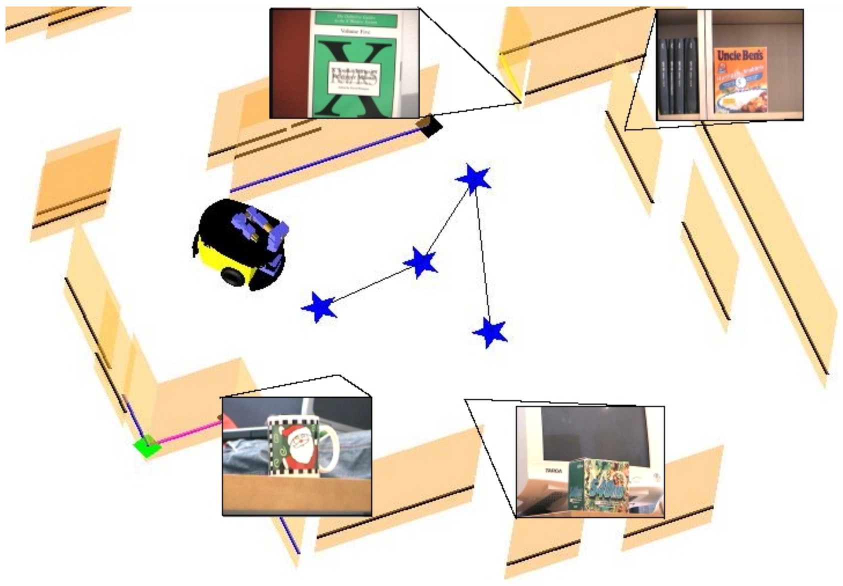


Figure 14: The results of a robot detecting four objects in a living room and estimating their positions.

5.3 Searching for Objects in Several Rooms

In this experiment, the search for objects is not limited to a single room. However, we skip door nodes and nodes directly adjacent to doors as stopping in these places blocks the way for people moving around. We used only two objects in this experiments, a soda can and the rice package. Figure 15 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 and thus correctly partitioned the space into rooms. This demonstrates the possibility of instructing the robot to fetch object X in room Y.

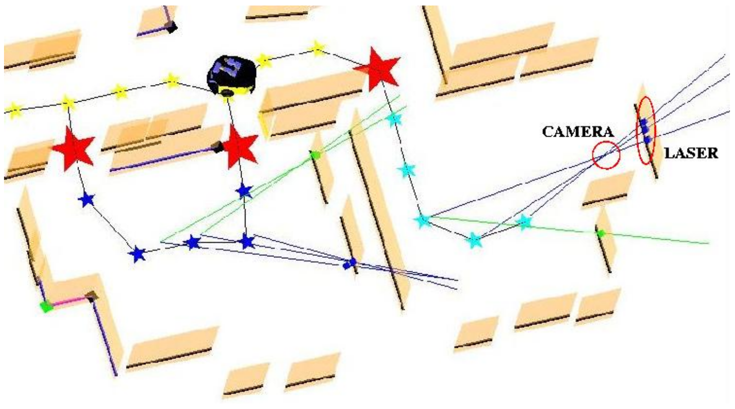


Figure 15: Searching for two objects in two rooms: a rice package placed on a dinner table is precisely localized by camera compared to laser.

6 Conclusion

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. In this paper, this has been demonstrated in a object detection/recognition scenario. We have shown how we build upon a SLAM system using different types of features (lines, points, SIFT) and sensors (laser, camera) and how we construct a navigation graph that allows the robot to find its way through the feature based map. We have also shown how we can partition this graph into different room with a simple strategy. A new method suitable for object detection has been presented. It is based on Receptive Field Cooccurrence Histograms that uses different image filter responses and normalized colors.

The experimental evaluation shows that the object detection method is able to successfully detect objects in cluttered scenes by comparing the histogram of a search window with the stored representation. An extensive experimental evaluation has shown that the method is robust. The representation itself is invariant to translation and rotation and robust to scale changes and illumination variations. The algorithm is able to detect and recognize many objects with different appearance, despite severe occlusions and cluttered backgrounds. The performance of the method depends on a number of parameters but we have shown that the choice of these are not crucial. On the contrary, the algorithm performs well with a wide variety of parameter values.

The strength in the algorithm lies in its applicability to object detection for robotic applications. There are several object recognition algorithms that perform very well on object recognition image databases assuming that the object is centered in the image on a uniform background. For an algorithm to be used for object detection, it has to be able to recognize the object although it is placed on a textured cloth and only partially visible. The CODID image database was specifically designed for testing these types of natural challenges, and we have reported good detection results on this database. The algorithm is fast and fairly easy to implement. Training of new objects is a simple procedure and, as it has been demonstrated, only one or, in cases with large differences in depth, two training images are sufficient for a good representation of the object.

There is still place for improvement. The cluster-center representation of the descriptor values is not ideal, and more complex quantization methods are to be investigated. In the experiments we recognized only 10 objects. More experiments are required to evaluate how the algorithm scales with an increasing number of objects, and also to investigate the method's capability to generalize over a class of objects, for example *cups*. In this work, we have used three types of image descriptors considered on several scales, but there is no upper limit of how many descriptors the algorithm may handle. There may be other types of descriptors that would improve results, and additional types of descriptors will be considered in the future.

Finally we shown how to augment a SLAM map with information about objects positions. One of the limitations of the current approach is that the system does not use the knowledge of the latest position of the object when searching for it in the frames node. This put some of the unnecessary burden to the search process and we will optimize this process in the future. In this work, we have not presented the full six-dimensional pose estimation of the object or online pose tracking, two methods that will be used for object manipulation and fetch-and-carry types tasks. In a longer run, we believe that the system will also allow us to determine what type of objects are typically found in certain types of rooms which can help recognizing the function of a room that the robot has never seen before such as a kitchen or a workshop.

References

[1] D. H. Ballard. Animate vision. *Artificial Intelligence*, 48(1):57–86, 1991.

[2] 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,*, pages 345–359, 1983.

[3] CODID - CVAP Object Detection Image Database. http://www.nada.kth.se/˜ekvall/codid.html.

[4] S. Ekvall, F. Hoffmann, and D. Kragic. Object recognition and pose estimation for robotic manipulation using color cooccurrence histograms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'03*, pages 1284 – 1289, 2003.

[5] S. Ekvall and D. Kragic. Receptive field cooccurrence histograms for object detection. In *Proc. IEEE/RSJ International Conference Intelligent Robots and Systems, IROS'05*, pages 84–89, 2005.

[6] J. Folkesson, P. Jensfelt, and H. I. Christensen. Vision SLAM in the measurement subspace. In *Proc. of the IEEE International Conference on Robotics and Automation, ICRA'05*, pages 18–22, April 2005.

[7] G. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernndez-Madrigal, and J. Gonzlez. Multi-hierarchical semantic maps for mobile robotics. In *Proc. IEEE/RSJ International Conference Intelligent Robots and Systems, IROS'05*, pages 2278 – 2283, 2005.

[8] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison Wesley Publishing Company, 1992.

[9] A. Gopalakrishnan and A. Sekmen. Vision-based mobile robot learning an navigation. In *IEEE International Workshop on Robot and Human Interactive Communication, RO-MAN'05*, pages 48–53, 2005.

[10] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman. A framework for vision based bearing only 3D SLAM. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'06)*, pages 1944 – 1950, Orlando, FL, 2006.

[11] T. Kawanishi, H. Murase, and S. Takagi. Quick 3D object detection and localization by dynamic active search with multiple active cameras. In *IEEE International Conference on Pattern Recognition, ICPR'02*, pages 605–608, 2002.

[12] 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*, volume 4, pages 3469–3655, October 2004.

[13] 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, March 2006.

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

[15] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 1:281–297. University of California Press, 1967.

[16] O. Martínez Mozos, C. Stachniss, and W. Burgard. Supervised learning of places from range data using adaboost. In *Proc. of the IEEE International Conference on Robotics and Automation, ICRA'05*, pages 1742–1747, Barcelona, Spain, 2005.

[17] 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. RO-MAN*, pages 172–177, August 2005.

[18] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[19] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library: Coil-100. In *Technical Report CUCS-006-96, Department of Computer Science, Columbia University*, 1996.

[20] P. Newman and K. Ho. SLAM-loop closing with visually salient features. In *IEEE International Conference on Robotics and Automation, ICRA'05*, pages 635 – 642, Barcelona, Spain, 2005.

[21] P. Newman, J. Leonard, J.D. Tardós, and J. Neira. Explore and return: Experimental validation of real-time concurrent mapping and localization. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'02)*, pages 1802–1809, Washinton, DC, USA, May 2002.

[22] M. K. Reed P. K. Allen and I. Stamos. View planning for site modeling. In *Proc. DARPA Image Understanding Workshop*, pages 1181–1192., November 21-23 1998.

[23] E. Pacchierotti, H.I. Christensen, and P. Jensfelt. Embodied social interaction in hallway settings: a user study. In *IEEE Workshop on Robot and Human Interactive Communication (RO-MAN'05)*, pages 164–171, Nashville, TN, August 2005.

[24] 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'02*, volume 3, pages 2500 – 2505, 2002.

[25] A. Pronobis, B. Caputo, P. Jensfelt, and H.I. Christensen. A discriminative approach to robust visual place recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'06*, 2006.

[26] D. Roobaert. *Pedagogical Support Vector Learning: A Pure Learning Appr oach to Object Recognition*. PhD thesis, Computatioal Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology, Stockholm, Sweden, May 2001.

[27] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

[28] C. Schmid and R. Mohr. Combining grayvalue invariants with local constraints for object recognition. In *International Conference of Computer Vision and Pattern Recognition*, pages 872–877, 2002.

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

[30] M. Swain and D. Ballard. Color indexing. *International Journal of Compter Vision*, 7:11–32, 1991.

[31] 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*, pages 1338–1343, 2002.

[32] 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)*, pages 3469–3475, New Orleans, April 2004.