doi:10.2498/cit.1001182

Object Search and Localization for an Indoor Mobile Robot

Kristoffer Sjö, Dorian Gálvez López, Chandana Paul,
Patric Jensfelt and Danica Kragic

Centre for Autonomous Systems, Royal Institute of Technology, Stockholm, Sweden

In this paper we present a method for search and localization of objects with a mobile robot using a monocular camera with zoom capabilities. We show how to overcome the limitations of low resolution images in object recognition by utilizing a combination of an attention mechanism and zooming as the first steps in the recognition process. The attention mechanism is based on receptive field cooccurrence histograms and the object recognition on SIFT feature matching. We present two methods for estimating the distance to the objects which serve both as the input to the control of the zoom and the final object localization. Through extensive experiments in a realistic environment, we highlight the strengths and weaknesses of both methods. To evaluate the usefulness of the method we also present results from experiments with an integrated system where a global sensing plan is generated based on view planning to let the camera cover the space on a per room basis.

Keywords: spatial mapping, object search, visual search, view planning

1. Introduction

The field of mobile robotics is continuously expanding. The question is no longer if robots will take the leap out of the factories and into our homes, but when. The future applications of autonomous agents require not only the ability to move about in the environment and avoid obstacles, but also the ability to detect and recognize objects and interact with them. Yet most robotic applications tend to be one of the two: Either entirely blind to everything in their surroundings except what is required merely for navigating through the environment, or else designed to function in a fixed setting, where they have a well-known and unchanging frame of reference that they can relate objects to. Nevertheless, there are some recent attempts to overcome these limitations. For example, the robot league of the Semantic Robot Vision Challenge (SRVC) [12] is a promising attempt to advance understanding and development in this area. The topic in this paper contributes to the field in that it deals with object search and detection in realistic indoor environments and thus aims at reducing the aforementioned limitations.

For mobile robots operating in domestic environments, the distance to the objects varies significantly: often it is too large to perform reliable detection/recognition, especially when using low resolution images. Successfully recognizing an object requires that the robot moves closer to it or zooms in on it, which in turn assumes that the object has already been detected in the field of view. Different methods have been proposed to determine the area of interest of an image, so that the robot knows what to zoom at. In [14] a foveated dynamic attention system is demonstrated. The system uses edges and circular features to direct attention, though it is done in a non-specific fashion; this is also the case in [6], where a measure of feature saliency inspired by human cognition is used in order to provide a sequence of attentional saccades to potential interest areas in the image. The VOCUS system [4] is another example of a biologically inspired attention system. The top-placing entrants in [12] similarly use non-specific saliency to direct attention for object detection. An attention control method that uses contextual information is described in [10], although its specificity applies to the area surrounding objects rather than objects themselves. In [2] an object-specific attentional mechanism is described. It utilizes receptive field cooccurrence histograms (RFCH), which provide different hypotheses for any occurrence of each object in the image. Zooming in combination with controlling the pan/tilt-angles is used to provide a closer view of the objects for the later recognition step which is performed using SIFT feature matching.

To perform an efficient object search and detection in a realistically-sized environment, the robot needs the ability to plan when and from where to acquire images of the environment, as exhaustive search is unfeasible. View planning is a comparatively old, but still thriving research area. Early work on view planning was of a mostly theoretical nature, but as the field has matured more implementation-oriented results are emerging. [15] examines the problem of optimally covering the "view sphere", i.e. all angles that can be seen from a fixed point in space, given a probability distribution for the presence of the object. In [16], the approach is augmented with multiple viewpoints, each next point selected by a greedy policy. The general problem of finding a minimal set of viewpoints from which to observe all parts of the environment is called the *art gallery problem*. [8] proves that this problem is NP-hard, and thus approximate solutions are required. Using a polygonal map of the robot's surrounding, [5] uses a sampling scheme to find an approximate solution to the art gallery problem while additionally taking into account the practical limitations of sensors by postulating maximum and minimum distance and maximum viewing angle. However, parameters for only a single object are considered.

Another related view planning problem is the *watchman problem*, which entails computing a minimal continuous path through space from which all of the environment can be seen; here, the length of the path is what is crucial - in contrast to the art gallery problem, where the distance between viewpoints is immaterial. The watchman problem, too, is NP-hard when there are "holes" in the free space (as shown in [1]). Many different approaches exist to solving both the art gallery and watchman problems; [11] provides an extensive survey. In [13], the cost of moving and processing views is combined in a single planning task, approximated as an integer linear problem (ILP). A set of candidate view points is assumed to be provided.

1.1. Contributions

Using a combination of view planning and visual search, we show how existing computer vision methods can be used on a mobile robot platform to produce an autonomous system that is able to efficiently detect and localize different objects in a realistic indoor setting. The proposed system is implemented on a mobile robot and its practicability is demonstrated in experiments.

We build further on the ideas presented in [2]. For the local visual search we add a vision-based object distance estimate that facilitates better zooming capabilities. In [2] distances to objects were estimated using a laser scanner. We also improve the way multiple objects can be searched for at the same time through better utilization of shared zooming. We also add a more efficient view planning strategy that takes into account the layout of the environment and the specific constraint of the individual objects. Finally, we present results from an extensive experimental evaluation of the vision based distance estimation and show examples of runs with the entire integrated system.

1.2. Hardware

The robotic platform used is a Performance PeopleBot. It is equipped with a SICK laser rangefinder with a 180 degree field, positioned near the floor (at about 30 cm), and with a Canon VC-C4R video camera, able to acquire low resolution images (320×240 pixels) with pan/tilt functionality and up to $13\times$ magnification. The camera is mounted about 1m above the floor. The robot has a differential drive and a wireless LAN connection.

2. Navigation

The robot is provided with a metric 2D-map, consisting of line features representing structures in the environment such as walls as well as with its own location in this map. The map is generated in advance by the robot using laser data and standard SLAM methods as presented in, for example, [3]. The robot is also given a set of nodes in 2D-space with edges between them, constituting a *navigation graph* which represents known robot-navigable space. Similar ideas have been presented in [7]. This is performed in the mapping step; as the robot moves into unexplored areas, it drops nodes at regular intervals, and when it moves between existing nodes it connects them in the map.

For example, consider the situation in Figure 1. It represents the map of a room built by means of laser data, where stars and lines represent the navigation graph created during exploration. The figure also shows some objects placed at different positions in the room. The objective of the object search algorithm presented further on in the paper is to detect all the objects, without prior information about their location, while keeping the trajectory traveled during the search short.

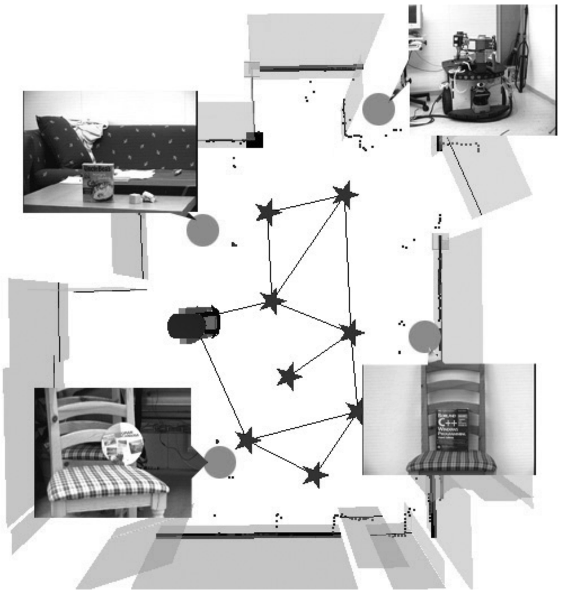


Figure 1. Example of distribution of objects in a room. Stars represent nodes, and circles, the actual positions of objects.

The object search begins with a planning step used to determine an efficient movement policy for exploring the map. In the current system, only the navigation nodes are considered during planning, as they are the only parts of the map guaranteed to be reachable. This way, the process is also computationally cheaper.

The map used in the search procedure represents a room, i.e. a more or less convex space, that consists of different types of geometrical structures such as walls, furniture, etc. Starting originally with a more complex map of the environment, the maps of single rooms are obtained by dividing the navigation graph into subgraphs by cutting out all the *door nodes* [7]. All features of the map are assigned to the room which has the nearest navigation node. Planning efficient movement between rooms is currently performed using a next-closest-room-first strategy.

The resulting navigation plan provides the robot with a list of nodes it needs to visit. For each node, there is also a list of viewing directions or views that it needs to process as well as the list of objects it has to look for in each view. The navigation plan is defined so that all parts of the room are searched for all the objects, while keeping the number of visited nodes and visual searches as low as possible. Additional object constraints must also be fulfilled as, for example, a small object can only be detected/recognized from a short distance. Finally, uniqueness must be taken into account: objects should be discarded once they are found, which means the exploration plan will need to be updated.

2.1. Grid-based View Planning

2.1.1. Occupancy Grid

The metric map built using SLAM is not geometrically perfect. Features extracted from laser data do not form a clean, continuous outline. Typically, different and commonly overlapping line features explain the same sensor data thus making the resulting clutter to increase planning complexity. For this reason, a simpler occupancy grid-based representation is used as the base for the view planning. The occupancy grid can be acquired either directly from laser data or by rasterizing an existing feature map by simply marking a cell as occupied if it contains a feature.

Note that the occupied cells are not assumed by the algorithm to obstruct field of view of the camera in any way. As the data originates from the laser, which is placed low on the robot, an occupied cell may not correspond to occluded field of view for the more highly placed camera.

Grid cell size is a tuning parameter; a small cell size will result in a lot of points to cover, which means higher accuracy, but also higher computational cost. Small cells will be very closely packed and thus grouped into the same viewing directions for the robot to process. On the other hand, a too-large cell size will lead to insufficient detail in the plan and the robot may miss parts of the map to explore. In the current system, a fixed cell size of 0.5m is used.

2.1.2. Views

Using the grid, *views* can be calculated. A view is a triplet consisting of the map node to which the robot has to travel, the direction it should point its camera to and the list of objects to be searched for in the resulting image. In order to simplify the calculations, grid cells are considered visible in a view if their center point is inside the field of view of the robot.

2.1.3. Object Constraints

There are various objects the robot has to look for and their attributes must be taken into account; specifically, their size, since it affects the distance from which an object can be detected/recognized. Hence, for each object a minimum and a maximum distance value is defined; the robot should attempt to find it only at distances in this interval.

There are separate distance constraints for object recognition and object detection. For recognition, the minimum distance is simply defined as the range at which the object would fill an entire image with a default zoom. The maximum distance is defined as the range at which the object would occupy an entire image if maximum zoom was used. The minimum distance for purposes of detection is given by the parameters of the detection algorithm, explained in more detail in Section 3.2.

Figure 2 shows an example of two potential views of a set of cells that, in this case, originate from a wall. Large circles represent nodes; dots, grid cell centers; the numbers close to them, the nodes they are associated with; and the shaded area, the views (along with objects planned for in each view). Note how views from both the nodes 15 and 18 are needed in order to cover the right-hand wall due to the different sizes of the objects. Other parts of the map are covered by different sets of nodes.

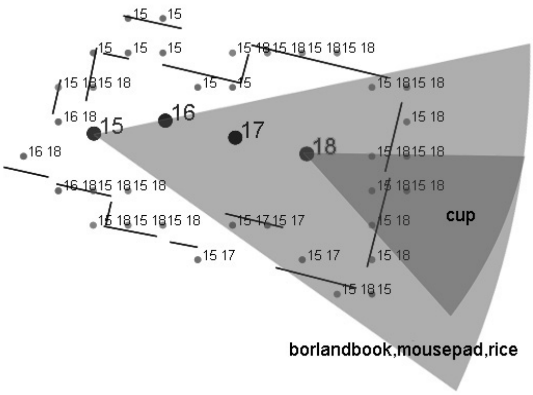


Figure 2. Example of the effect of distance constraints on view planning.

2.1.4. Planning Strategy

The objective of the algorithm is to ensure that any of the sought objects would be seen in at least one of the planned views regardless of which cell it was in – in other words, each possible object-cell combination must be covered by some view.

After generating the grid, the view covering the most object-cell pairs is chosen iteratively until no pair remains that is not covered by any view, or no view remains that covers further pairs. In the latter case, the pairs are impossible to cover given the provided set of nodes. The object-cell pairs covered by each view chosen are removed from subsequent iterations when they are covered by the desired number of views, which is 1 in the simplest case.

The plan is executed by visiting the closest navigation graph node that has a view being part of the list, performing object search for all its views, then moving on to the next closest node and so on. If an object known to be unique is found during search, it is eliminated from the plan; if any views in the plan are rendered empty by this, they can be removed as well. Removing an object involves simply checking which views contain it – without performing any new geometrical calculations – which operation has linear time complexity. As it is evident from the above, the algorithm proposed is greedy in terms of nodes and map cells. Although it does not ensure an optimal solution, it allows for obtaining a low number of views in polynomial time.

2.1.5. Tilt Angle Selection

Since a 2D map of the environment is used, there is no direct information that could help in deciding how to use the tilt angle of the camera. Yet, the objects being sought for might be at any height, and thus some thought must be given to covering the vertical dimension as well as the horizontal ones.

Those grid cells which are closer to a given view's associated node than a set threshold (here set to 2 meters) generate new views that cover the vertical extent of the objects' possible locations. Using the average distance for those grid cells, together with an upper and a lower boundary for objects' positions, one or more tilt angles are selected (with as little overlap as possible) and the resulting views are added to the plan.

3. Vision

To detect objects, the system uses receptive field cooccurrence histograms (RFCH) as described in [2]. As potential objects are detected, the system calculates suitable interest regions for the camera to zoom on. Here, the system needs an estimate of the distance to the object in order to decide whether to proceed with recognition given the current camera parameters or if zooming is needed. In [2] this estimate was taken from the laser scanner; we, instead, obtain an estimate through the RFCH procedure itself.

If the distance allows for reliable recognition immediately, the system uses SIFT feature matching in order to recognize the object [9]. Otherwise, the interest regions for the different objects are merged as far as possible and the camera zooms in on each region in turn, repeating the procedure until all objects have either been found or eliminated. A detailed description of the above procedure is presented in the following sections.

3.1. Object Search Algorithm

Figure 3 presents the object search procedure as a whole. Starting with an image at 1× magnification, each object is processed independently, whereupon the resulting zoom windows are merged and each gives rise to a new, zoomed image and the procedure repeats for each of them.

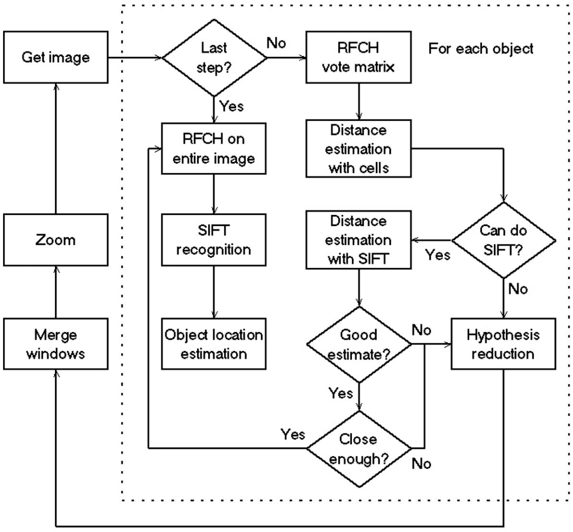


Figure 3. Object search algorithm.

The algorithm has three steps: initial, middle and final. It progresses through them according to the following:

- Initial: No magnification used. After distance estimation and zooming, it proceeds to the middle step.
- Middle: Magnification given by output from zoom window sharing (Subsection 3.4.2). If new distance estimate indicates current magnification is too small (not within 1.8× of new desired middle magnification), this step is repeated. If, on the other hand, it is within 1.2× of the desired final magnification, it skips straight to recognition. Otherwise, it moves to final step without further zooming.
- Final: Magnification in accordance with Eq. 1. Recognition performed.

Typically, each step will run once only.

In the first two steps, an RFCH vote cell grid is created and used to extract a set of hypotheses. Then, distance is estimated using the strongest hypothesis. If the distance found is small enough (here, such that it would require less than 3× the current magnification), SIFT matching is performed for a more accurate distance measure. If this estimate in turn says that the object is sufficiently magnified, the algorithm jumps to recognition; otherwise, the most reliable distance is used to produce a zoom window for the next step. Hypothesis grouping and reduction (Subsection 3.4.1) prunes the result for each object; then, hypothesis sets for the different objects are merged.

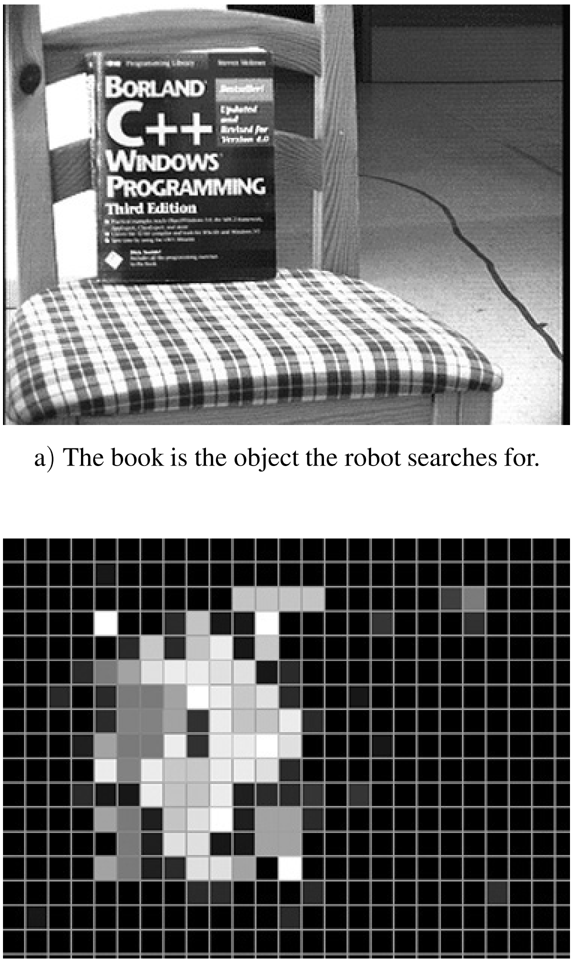
The magnification required for the algorithm to jump to the recognition stage is a tuning parameter; in the current system, it is set to 1.2× of the final. There is also a "short-cut" that lets the object go straight from step 1 to step 3 if the current magnification is found to be essentially equal to the middle magnification (within 1.8×).

The last step in the object search consists simply of recognition, wherein SIFT matching is preceded by a "sanity check" RFCH match on the entire image (see Subsection 3.5). If the object is found, its location in space is computed from its position in the image and the distance estimate. The output of the algorithm is a list of objects that were found in the current view and their calculated locations and distances.

3.2. Object Detection

The detection process works on a per-object basis and consists of several steps: an image is first taken by the camera and divided into cells. For each cell, RFCHs are computed using clusters learned from each respective object in a training phase. These are then matched against the training images' histograms, resulting in a similarity value for each cell and object. This set of cell values is called the object's *vote matrix*. An example is shown in Figure 4. Higher value of cells (represented by a lighter shade in the image) denotes a greater degree of correspondence between test image and training image.

Next, object hypotheses are generated. A cell is a hypothesis if its value is higher than those of its 8-connected neighbors, as well as higher than an object-dependent threshold. The size of the vote cells in the above algorithm is a tuning parameter. Large cells mean faster histogram matching; however, they decrease the detection rate when they are larger than the objects in the initial image. Also, maximum distance allowed for object detection during the planning step is set to the distance at which an object would occupy a single cell at no zoom, which decreases as cell size grows. The value used in our work, 15 × 15 pixels, is a compromise between these considerations.



a) The book is the object the robot searches for.

b) The vote matrix for the book in the above image.

Figure 4. Vote matrix (b) generated from image (a) where the object occupies large portion of the image. The lighter a cell, the higher probability of the object.

3.3. Distance Estimation

In [2], the distance estimate used to determine zoom levels was based directly on the robot's laser sensor. However, the distance provided by the laser is often misleading, as Figure 5 shows: the laser sensor is placed about 30 cm above the floor and if an object is not at that height, the estimate may be wrong. The approach works only for objects that are placed on the floor or are located close to walls (for example, in a bookshelf). If the distance estimate is wrong, the final zoom may either not be sufficient to make the object occupy enough of the image, or otherwise may be too large causing only a small part of the object to be seen. Furthermore, even if the object is recognized, its estimated position might be inaccurate. To address these issues, in this work we use two alternative ways for distance estimation.

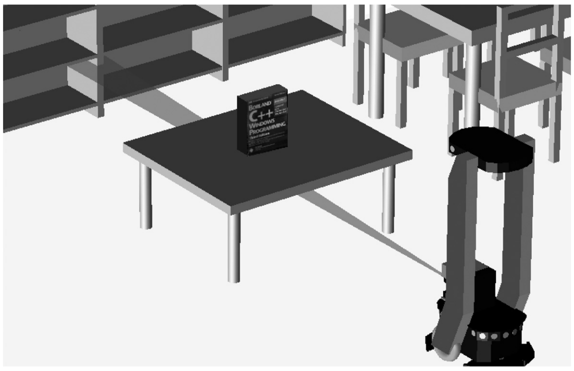


Figure 5. Distance estimation provided by the laser may not be reliable: instead of the distance to the object on the table, the distance to the shelf is measured.

3.3.1. Using the Vote Matrix

Using the RFCH vote matrix for distance estimation consists of measuring how many cells are part of the object and treating the area they occupy in the image as an approximation of the object's size. Here, cells are considered to be associated with a hypothesis if their degree of match is above the threshold and if there is an 8-connected path to the hypothesis with cells of monotonically increasing value. Only the strongest hypothesis and its associated 8-connected cells are taken into account, because it is likely to be the most reliable.

Given the object's actual size stored in the training database, the distance is then computed as:

$$D = \frac{W_{real}\dfrac{W_{im}}{2D_{vote}}}{\tan\left(\dfrac{\alpha}{2}\right)}$$

where D stands for the estimated distance (meters); W_{real}, for the real width of the object (meters); W_{im}, for the width in pixels of the camera image; D_{vote}, for the width in pixels of the bounding box of the cells associated with a hypothesis and α, the horizontal viewing angle. This procedure is fast and approximate, but sufficiently accurate to allow the object search algorithm to assign a valid zoom.

3.3.2. Using SIFT

SIFT produces a scale parameter for each key point extracted. For each matched pair of key points in the training and recognition image, the quotient of the keys' scale parameter gives an estimate of their relative apparent size and hence their distance, according to:

$$D = \frac{W_{real}\dfrac{W_{im}}{2W_{tr}}\dfrac{S_{tr}}{S_{real}}}{\tan\left(\dfrac{\alpha}{2}\right)}$$

where S_{tr} denotes the scale of the point extracted from the training image; S_{real}, the scale of the point extracted from the recognition image, and W_{tr}, the width of the object in the training image in pixels.

As mis-matched key point pairs can produce incorrect scale parameters, the final estimate of the object distance is taken as the median of the distance estimates from all matches. Experiments indicate that an adequate estimate is obtained given 10 or more SIFT matches. With 4 matches or more, a passable rough estimate is typically obtained (within about 30%). If there are fewer than 4 matches, the result is likely to be very poor (most likely based on some other structure than the object) and is not used.

The drawback of the above method is that extracting SIFT features from an image is computationally expensive, and using it to guide the zoom process may take too long to be feasible. Another problem is the number of SIFT features required to obtain a robust estimation; when the object is small in the image (i.e. resolved by few pixels), it is unlikely that enough matches will be available.

3.4. Calculation of Zoom

Given a training image of an object, its size, the distance to the object and the camera field of view, we want to calculate the magnification needed to make it fill the image as much as possible. The size of the object is approximated by the size of its bounding box. In order to make the object fill the image, the desired horizontal angle of view (α), as well as the vertical (β), are calculated as:

$$\begin{aligned} \alpha &= 2 \arctan \left(\frac{W_{real}}{2D} \right) \\ \beta &= 2 \arctan \left(\frac{H_{real}}{2D} \right) \end{aligned} \quad (1)$$

where W_{real} and H_{real} represent the known width and height of the object.

Since the object will typically not have the same aspect ratio as the image, only one of the angles α and β can be used to select the magnification parameter. Therefore, of the two levels of magnification suggested by the height and the width respectively, the lower level is selected, as given by the following rule:

If $\frac{W_{im}}{W_{tr}} < \frac{H_{im}}{H_{tr}}$ (H_{im} and H_{tr} being the heights analogous to the widths W_{im} and W_{tr}), α is used; otherwise, β.

3.4.1. Hypothesis Grouping and Reduction

Even with the threshold, there are typically too many hypotheses to evaluate one-by-one. In order to avoid excessive zooming and processing, hypotheses are grouped together into *zoom windows*, which are regions of the image to be magnified and processed. The size of these windows is determined by the magnification recommended by the distance estimate of the strongest hypothesis. The position of the window is chosen to cover the maximum number of hypotheses, and this is repeated until all hypotheses are covered.

In cases when the distance parameter is not very accurate, an error propagates into the calculation of the magnification parameter and into the size of the zoom windows. This may lead to generating more zoom windows than is warranted and, consequently, lengthening the search process. Thus, as a second step it is desirable to remove those windows which do not contribute information to the search, as they either contain too few hypotheses or are located close to “richer” zoom windows. Therefore, zoom windows which overlap more than 20% with another containing at least 3 times its number of hypotheses are removed. These conditions are quite conservative in order to ensure that no potentially important zoom windows are removed.

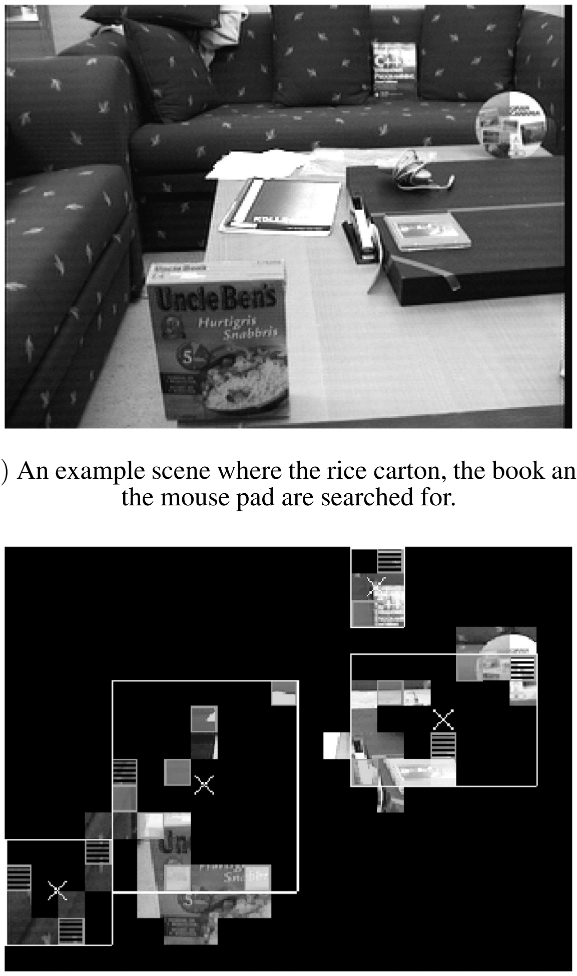
3.4.2. Zoom Window Sharing

When searching for several objects, the set of zoom windows obtained for each object is computed separately. After this is done, the combined set of windows needs to be merged to reduce redundant steps. Here, we look for instances of a zoom window encompassing that of another object, in which case we can remove the latter.

Not all the zoom windows that overlap can be merged, as straying too far from each object’s target magnification may cause object detection to fail. The object search process has three steps: the first step without zooming, the second step with a middle-level zoom and the third step with large zoom; see Subsection 3.1. It is not as important that the middle-level zoom is exact, since it is only used for hypothesis finding with RFCH. Thus, a maximum and a minimum magnification is defined for the middle-level detection step, allowing some flexibility in selecting the windows to be used in this step. It is most important to get the minimum zoom right: the lower it is, the more objects we can look for at one time, but the higher the risk that objects are missed because they appear too small in the image.

The algorithm works as follows: first all zoom windows are shrunk to their minimum size. Then, each zoom window associated with an object A is compared with those of an object B. If the hypotheses contained by one of B’s windows can be made to be contained by one of A’s – expanding the latter if needed, while conforming its maximum allowed size – then the B window is removed and object B is added to the A window’s list of candidate objects to look for in the next step. This procedure is repeated for each pair of objects. Tests have shown that too flexible a window size tends to be harmful to detection; in this work the maximum middle-level magnification is set to 0.75 of the final zoom level, and the minimum to 0.7 of the final zoom level.

Given the object distribution shown in Figure 6. a), Figure 6. b) shows an example of this process. Small squares represent hypotheses, whereas big rectangles are the zoom windows they are grouped into. The brightest hypotheses



a) An example scene where the rice carton, the book and the mouse pad are searched for.

b) Shared zoom windows.

Figure 6. Shared zoom windows of three objects placed at different distances.

arise from the rice carton; the dark ones, from the book, and the striped ones from the mouse pad. Note that three of the windows contain hypotheses for more than one object.

3.5. Object Recognition

The final object recognition is done once the object occupies either the whole image or a large portion of it. It consists of extracting SIFT features from the current image and matching them with the SIFT features in the training image. SIFT features are scale-, position- and rotation invariant up to a certain level, meaning that many of the features will match even if the object is seen from a different angle or under different lighting conditions. However, it is usually the case that the number of SIFT matches during the search is much lower than the number extracted from the training image, due to changes of viewing angle and background. Because of this, we consider an object to have been found if at least a 5% of the SIFT features match. This value has previously been demonstrated to result in few false positives in [2].

Once an object is recognized, its position in the environment is calculated from the pan and tilt angles of the camera, the estimated position of the object inside the image and the distance calculated by the system; see Subsection 3.3. Because of the large variation present in the images, it is very probable that false positives reach the last step of the visual search. In order to reduce the amount of unnecessary extraction of SIFT features, the same RFCH algorithm that is used for detection is used one last time on the fully zoomed image before running the recognition algorithm. The SIFT-based recognition is performed only if this match is successful.

4. Experimental Evaluation

Several experiments were performed to evaluate the proposed algorithms. Test objects used in the experiments are: a book, a rice carton, a printed mouse pad, a printed cup, a box for a trackball, and a large robot. The size of the forward face of the objects varied, from the cup at 14 × 10 cm to the robot at 63 × 55 cm. Color and shape were likewise diverse, providing a highly heterogeneous sample.

4.1. Object Detection Using RFCH

The robustness of RFCH object detection was evaluated in the following way:

Five test objects[1] (cup, trackball box, rice carton, book, mousepad) were placed at eight different distances, between 0.5 m and 4 m, from the robot's camera, using two different backgrounds (a plain white wall and a typical office scene). The robot was excluded in this evaluation, as the ranges where it is detected differed too much from the other objects. Five images per position were obtained, introducing some perturbation in the object between each.

[1] The robot was excluded in this evaluation, as the ranges where it is detected differ too much from the other object.

As described previously, for each image, RFCH was used to calculate the similarity value for each vote cell, and this was thresholded according to an object dependent threshold. The vote cells whose values were above the threshold were segmented into 8-connected regions and the local maxima of these regions were extracted as hypotheses. The hypotheses were manually labeled as true if they overlapped with a part of the object in the test image, otherwise they were considered false. Figure 7 shows the ratio of false hypotheses generated in the set of test images as a function of distance.

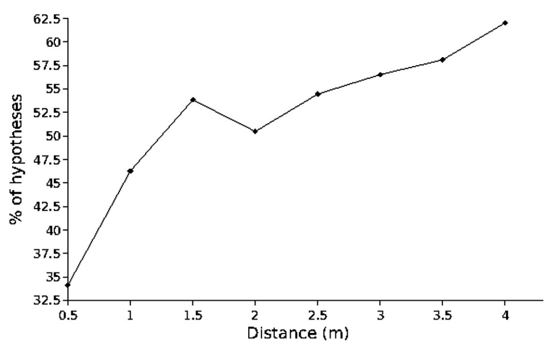


Figure 7. Percentage of false hypotheses generated by the RFCH attention mechanism.

Generally, at larger distances less pixels of the object are visible and it is less distinct from the background. Thus, it can be seen that a larger number of false hypotheses are generated at larger distances. The rate of detection for the objects selected ranges from approximately 65% at close distances, to 35% at longer distances. This sensitivity affects the efficiency of the visual search, as false hypotheses can give rise to unproductive zoom positions, but it also ensures that true hypotheses are very rarely neglected.

4.2. Initial Distance Estimation

As mentioned in Subsection 3.3, initially in the visual search an approximate distance estimate is required in order to direct zooming actions and determine when SIFT extraction may be performed. Below are the results highlighting the properties of RFCH and SIFT, respectively, when used for this purpose. The same set of images was used as in Subsection 4.1, and the distance was computed using both RFCH and SIFT separately for every image. The distance estimate from RFCH is based on the strongest hypothesis. The performance of distance estimation is thus affected by how often a correct hypothesis is selected for distance estimation. Figure 8 presents statistics on the likelihood of selecting the correct hypothesis for distance estimation as a function of distance.

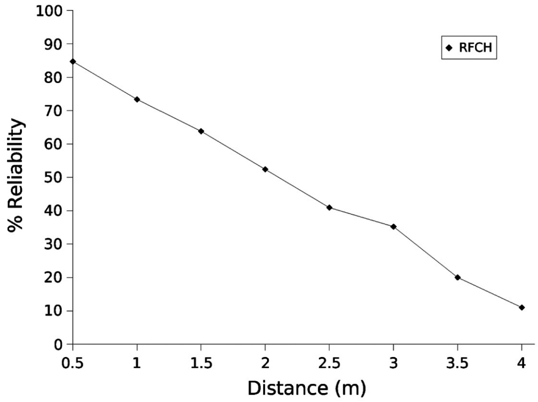
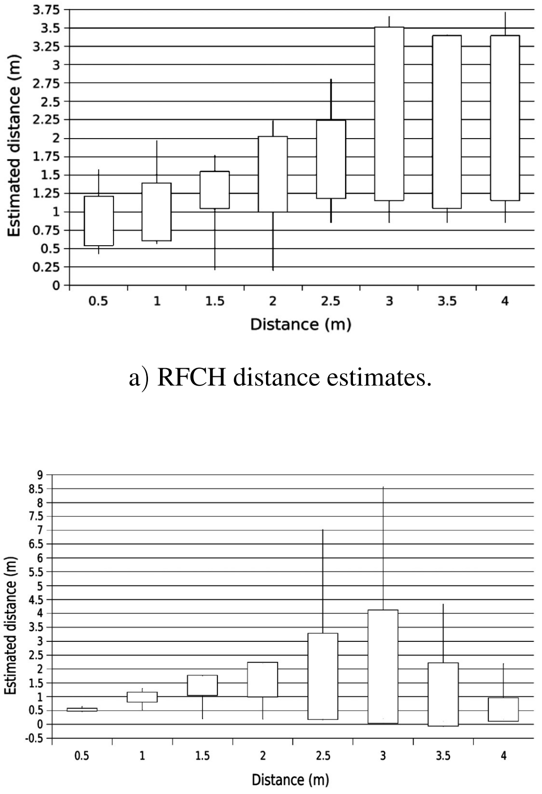


Figure 8. Percentage of correctly chosen hypotheses for distance estimation.

It is evident that reliability decreases with distance as the signal-to-noise ratio of the image drops. At 2.5 m and above, less than half of the distance estimates are based on the actual object in the image, and performance continues to drop until, at 4 meters, the area used becomes nearly random.

Figure 9 presents the results of distance estimation using RFCH and SIFT without magnification, performed on five different test objects. As expected, performance deteriorates for both methods at long range, due to the decreased size of the object in the image, and for RFCH also partly to the discretization of the vote cells.

It is notable that the values obtained through both methods tend towards the low end. The reason for this are mainly outliers, erroneously assigned values of 0.5–1 m, caused by large background structures being mistaken for a close-up object. Compared to RFCH, SIFT exhibits a far more accurate and dependable estimate at short range. However, its quality rapidly deteriorates at longer distances, as can be seen by



a) RFCH distance estimates.

b) SIFT distance estimates.

Figure 9. Distance estimation results; all objects. Top image RFCH, bottom SIFT. Boxes signify one standard deviation about the average for each distance; lines signify the most extreme values.

inspecting the average value of the estimates beyond 2.5 m in Figure 9. b). This is because a certain level of detail is needed to extract SIFT keys. In contrast, RFCH, though most reliable at medium ranges (as demonstrated by the standard deviations in Figure 9. a), retains the ability at long range to provide very rough approximations, generally adequate for the purpose of selecting a zoom level for the next step. For the final distance estimate, it should be pointed out that SIFT is used – but the magnification of the image will correspond to shifting the diagram in Figure 9. b) into the 0.5 m–1 m region where the method is most effective.

Figure 10 highlights the differences between RFCH and SIFT in distance estimation. Here, for each test image, the absolute error of the distance estimate is compared between the two methods and the percentage plotted of cases where RFCH gives the better estimate and vice versa. The graph shows that RFCH becomes more reliable at 2 m range or above.

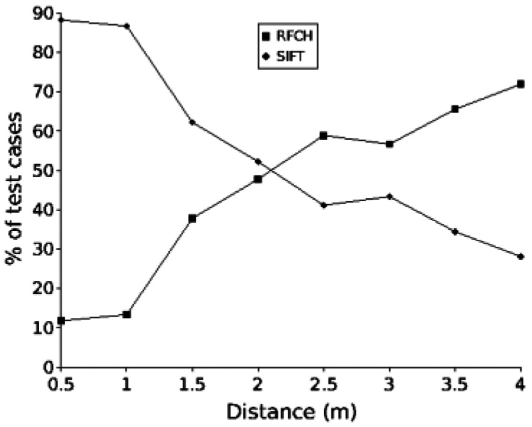


Figure 10. Proportion of instances in which RFCH and SIFT, respectively, provide the best estimate.

4.3. View Planning

The view planning algorithm was tested in three rooms with a different metric map in each experiment.

Some of the results of these experiments can be seen in Table 1, where object *B* stands for the book; *C*, for the cup; *D* for the robot; *M*, for the mouse pad, and *R* for the rice. The table shows how the number of objects involved in the exploration varies the amount of required searches. Note that the first case requires many more searches; this is because both the cup and the robot are regarded and their sizes do not allow them to be looked for in the same views. This effect is also visible when only one of them is included.

Objects	Area (m^2)	Nodes	Nodes used	Searches
BCDMR	31.6	8	7	18
BDMR	31.6	9	5	8
BCMR	31.6	8	4	8
	40.9	9	3	8
	17.2	4	2	6
BM	40.9	9	2	5
	17.2	4	2	3
B	31.6	7	2	5
	18.9	4	2	5

Table 1. View planning results.

4.4. Object Search

The book, the rice carton, the mouse pad and the robot were placed at different positions inside a room, as previously seen in Figure 1. Searching the room using estimations based on visual data produces the results shown in Figure 11. Note that all the objects are found and are accurately localized.

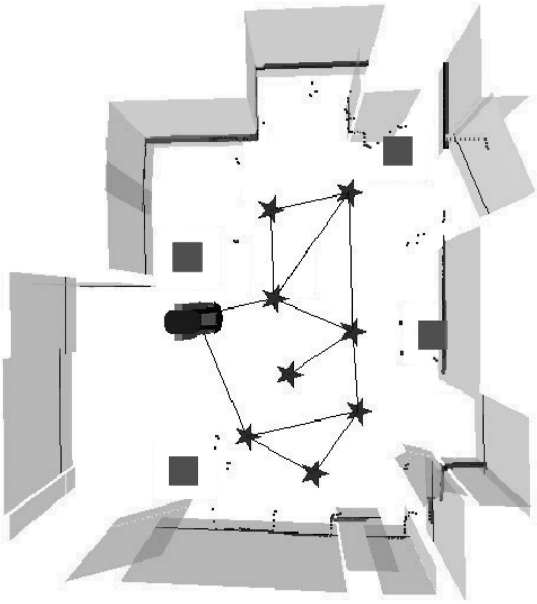


Figure 11. Object position estimation searching the room using image-based distance estimates.

4.5. Performance

Detailed timing of the performance of algorithms is not the aim of this paper, yet some qualitative evaluation has been performed. Of the various sub-tasks involved in performing the experiments described in Subsection 4.4, distance estimation takes the longest. This is because it constitutes a computationally complex task that is performed at each step of the visual search; once per acquired image per object.

Each such cycle typically takes a couple of seconds, a fact which makes the number of hypotheses (and thus searches) generated by the attentional mechanism very important. Figure 7 illustrates how the number of misleading hypotheses depends upon the distance (on average over all tests in Subsection 4.2). The false hypothesis count obviously also depends on the distinctiveness and size of the object sought.

The movement of the robot and the camera, in comparison, take up relatively little time, and the view planning carries a negligible cost in our experiments as well. Nevertheless, this does not mean that it would be more efficient to replace the zooming procedure with moving up close to objects: doing so would require more movement and more initial images in order to achieve full coverage, as well as more navigation nodes. Also, perspective causes more deviation from training images at closer range.

5. Discussion

The results indicate that the combination of RFCH-based long range object detection strategy, distance estimate-based zooming, and SIFT-based close range recognition leads to a successful strategy for object acquisition. The addition of a visual view planning technique gives rise to a viable approach for object search and localization in indoor environments. This approach has several advantages: the ability to simultaneously search for multiple objects of different sizes, cover the scope of the environment for all objects with a limited number of views, and detect objects at long range. There are numerous conceivable improvements that could be worthwhile to explore.

5.1. Attention Mechanism

RFCH is a comparatively new method and might be improved in a number of ways. For example, the object-specific thresholds and the vote cell size are currently set manually. Finding a generally applicable way of determining these parameters would be an important improvement. Moreover, the sensitivity of the method to noise means it often generates false positives, reducing efficiency. Methods for alleviating noise effects by, for instance, averaging RFCH responses over time or space would be worth investigating. The use of other types of similar long range detection techniques could also be considered.

5.2. Scalability

The current visual search is not highly scalable in terms of the number of objects to be sought at the same time. It is adequate when searching for a specific object, but for more general tasks such as exploring, inventory or active knowledge maintenance more efficient modes of object detection may be needed. The obvious way of dealing with this would be a first stage indexing approach, which could produce hypotheses for the presence of whole classes of objects and subsequently refine these. This would also make for a more compact internal representation.

A related approach is that of abstraction, in which visual processes extract some form of semantic information which may then be used to guide classification and recognition. Similarly, the view planning could benefit from categorizing objects in terms of e.g. size categories, which would decrease complexity for cases where many similarly sized objects are in the set being searched for.

5.3. Map Complexity

Using a 2D map obtained from laser scans for view planning is somewhat problematic; without very strong assumptions of spatial layout, it does not entirely convey a reliable picture of occlusions, nor of the probability of the occurrence of objects. It is also very sensitive to flawed room subdivision: cells belonging to neighboring rooms, that may well be completely hidden, can still affect the plan, leading to futile image searches.

Some sort of 3D representation, whether obtained from vision or range scans, could help in this regard. Another path that could be investigated is improving the methods for subdividing the map into regions within which the assumptions hold true. In the end, however, a map built only from a 2D occupancy grid cannot fully capture all the relevant structures of a complex environment; data from other modalities must be included in order to do this.

5.4. Viewing Angles

Another issue is that the view planning algorithm in its current form does not take into account the fact that objects may be difficult or impossible to detect or identify when seen at some angles, even setting aside occlusion by other objects. Specular glare, lighting or perceptual aliasing may vary depending on direction. To ensure detection in the face of these complications, the planner must be made aware of them on an object-by-object basis.

5.5. Prior Knowledge

The planning strategy in this paper implicitly assumes that the prior probability distribution of each object over all the feasible locations is uniform. This is not always the case in reality. One natural extension of this work would be to weight possible locations of objects with probabilistic knowledge (learned, directly provided or deduced from semantics) of the likelihood of objects' presence. Such a weighting might increase efficiency tremendously when the quality of the agent's knowledge is high.

Other promising avenues of research also include simultaneous integrated object detection and mapping, online object learning and hierarchical approaches to detection.

6. Conclusion

This article presents a solution for the object search and localization problem in a realistic environment, incorporating both planning for efficient view selection – including robot motion – and visual search using a combination of receptive field cooccurrence histograms and SIFT features, and a method of visual distance estimation for the dual purpose of zoom level calculation and object positioning in the map. In a set of experiments, we have evaluated the reliability of RFCH-based object detection, the accuracy of the distance estimation methods, the operation of the view planning technique, and visual object search and localization as a whole. The results indicate that the system presents a viable approach for object search and localization in indoor environments.

Acknowledgment

This work was supported by the SSF through the Centre for Autonomous Systems (CAS), the EU FP7 IST Cognitive Systems Integrated Project "CogX" FP7-215181-IP, the STINT foundation and the Swedish Research Council, contract 621-2006-4520.

References

[1] W.-P. Chin and S. C. Ntafos. Shortest watchman routes in simple polygons. *Discrete & Computational Geometry*, 6:9–31, 1991.

[2] S. Ekvall, D. Kragic, and P. Jensfelt. Object detection and mapping for service robot tasks. *Robotica: International Journal of Information, Education and Research in Robotics and Artificial Intelligence*, 2007.

[3] 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)*, pp. 30–35, April 2005.

[4] S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search.* PhD thesis, University of Bonn, July 2005.

[5] H.H. Gonzalez-Baños and J.C. Latombe. A randomized art-gallery algorithm for sensor placement. In *COMPGEOM: Annual ACM Sympo- sium on Computational Geometry*, 2001.

[6] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell*, 20(11):1254–1259, 1998.

[7] G.-J. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems*, 4(2), 2007.

[8] D. T. Lee and A. K. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] A. Oliva, A. B. Torralba, M. S. Castelhano, and J. M. Henderson. Top-down control of visual attention in object detection. In *ICIP* (1), pp. 253–256, 2003.

[11] T.C. Shermer. Recent results in art galleries [geometry]. In *Proceedings of the IEEE*, pp. 1384–1399, 1992.

[12] The semantic robot vision challenge. `http://www.semantic-robot-visionchallenge.org/`.

[13] P. Wang, R.h Krishnamurti, and K. Gupta. View planning problem with combined view and traveling cost. In *ICRA*, pp. 711–716. IEEE, 2007.

[14] C.-J. Westelius. *Focus of Attention and Gaze Control for Robot Vision.* PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 1995. Dissertation No 379, ISBN 91-7871-530-X.

[15] Y. Ye and J. K. Tsotsos. Where to look next in 3D object search. In *Symposium on Computer Vision*, pp. 539–544, 1995.

[16] Y. Ye and J. K. Tsotsos. Sensor planning in 3D object search. *Computer Vision and Image Understanding*, 73(2):145–168, 1999.

Received: December, 2007
Revised: April, 2008
Accepted: April, 2008

Contact addresses:
Kristoffer Sjö
Dorian Gálvez López
Chandana Paul
Patric Jensfelt
Danica Kragic
Centre for Autonomous Systems
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
e-mail: `dorian3d@gmail.com`,
`{krsj,chandana,patric,danik}@nada.kth.se`

Kristoffer Sjö is a robotics researcher at the Royal Institute of Technology (KTH), Stockholm, Sweden. He received his M.Sc. degree in applied physics and electrical engineering from the Linköping Institute of Technology in 2005, and has been working on his Ph.D. at KTH since 2007. His specialty is spatial representation and reasoning.

Dorian Gálvez López is a researcher at the Centre for Autonomous Systems at the Royal Institute of Technology (KTH), Stockholm where he also in 2007 pursued his Master thesis work. His research interests include robot localization and object recognition.

Chandana Paul received her Bachelors in brain and cognitive science (1996) and computer science (1998), and Masters in computer science (1998) from the Massachusetts Institute of Technology. She performed her Masters research at the MIT Artificial Intelligence Lab with Rodney Brooks, on the advancement of Mars rover technology. Following this, she moved to Zurich, Switzerland to perform doctoral research with Rolf Pfeifer at the Artificial Intelligence Lab, University of Zurich. Her research was in the area of biped robotics, and she received her PhD in computer science in 2004. She was most recently a Postdoctoral Researcher at the Mechanical and Aerospace Engineering Department at Cornell University, working with Hod Lipson and Ephrahim Garcia.

Patric Jensfelt received his M.Sc. in engineering physics in 1996 and Ph.D. in automatic control in 2001, from the Royal Institute of Technology, Stockholm, Sweden. Between 2002 and 2004, he worked as a project leader in two industrial projects. He is currently an assistant professor with the Centre for Autonomous System (CAS) and the principal investigator of the European project CogX at CAS. His research interests include mapping and localiation and systems integration.

Danica Kragic received her B.S. degree in mechanical engineering from the Technical University of Rijeka, Croatia, and her Ph.D. degree in computer science from the Royal Institute of Technology (KTH), Stockholm, Sweden in 1995 and 2001, respectively. She is currently a professor in computer science at KTH and chairs the IEEE RAS Committee on Computer and Robot Vision. She received the 2007 IEEE Robotics and Automation Society Early Academic Career Award. Her research interests include vision systems, object grasping and manipulation and action learning.