SIXTH FRAMEWORK PROGRAMME
INFORMATION SOCIETY TECHNOLOGIES

robots@home

An Open Platform for Home Robotics

Specific Targeted Research or Innovation Project

Deliverable 2.1

Methods representing space suitable to learning, semantic concept and the relation to human cognitive capabilities

Date: May 2, 2008

Organisation name of lead partner: ETH Zürich

Contributors: Stefan Gaechter, Cédric Pradalier, Davide Scaramuzza, Shrihari Vasudevan

Proposal/Contract no.: **FP6-2006-IST-6-045350**

Contents

1 Introduction

In [Vin06], ETHZ is leading the work-package 2, titled "Learning room layout". This deliverable reports on our effort towards Task 2.1 "Hierarchical representation of space". In order to show the relevance of our research to the international community, we chose to structure this deliverable as a collection of articles accepted in major international journals and conferences of the robotic community. In the following we will first show how these articles articulate together to form a consistent piece of work and then introduce the articles in their published forms.

1.1 Work-package description

For clarity, let us recall the objective of work package 2 and task 2.1 (from [Vin06]):

> The objective is to develop a hierarchical representations and cognitive map that provides the necessary capabilities to model space in several levels for combining and fusing the topological, metric and semantic information relevant for the home navigation task. It deals with building of such a representations of space from the perceptual input mainly attained in WP 3. Since perceptual data is never perfect, we need to study how to achieve consistency and scalability of the layer, how to effectively switch between the levels of the representation, and how to handle temporal dynamics and changes in environment.
>
> Finally, a main objective is to provide an easy yet intuitive user interface that portrays the information I the hierarchical representation to the human and selects useful information to obtain a better and consistent annotation of the things perceived. This human machine interaction is located here, because it is so tightly linked to the main function of showing the robot around: learning the room layout and annotating main items of furniture.
>
> **Task 2.1: Hierarchical representation of space**
>
> This Task focuses on the representation itself – i.e. the mapping process and its output. It addresses questions that deal with the content of the representation (e.g. objects and relationships between them), methods by which the representation is formed and managed (e.g. a combination of topological and metric layers – a hybrid map or a purely hierarchical representation) and how spatial and semantic concepts would be formed (e.g. association rules). Of particular importance is to enable the conceptualisation of space to learning spatial concepts as needed for the showing the robot around paradigm. It ends with the formation of semantic concepts within the space, which is then used for the annotations, see WP 3.
>
> Lastly, a comparison is also drawn on the similarities between the way we humans perceive and represent space and the corresponding representation in robots. This sheds a lot of light on what is a cognitive spatial representation and how robots could become more compatible with us. This also provides for a cognitive validation to the representation that would be finally stored. Thus, the sub package is the main focus of this work package. Contents at each level of the hierarchy and how they relate to each other.

1.2 Articulation of the deliverable

In the following, this report will include the following articles:

[SPS08]	D. Scaramuzza, C. Pradalier, R. Siegwart (2008), **Performance Evaluation of a Vertical Line Descriptor for Omnidirectional Images**, Proc. of *The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.
[GHS08]	S. Gachter, A. Harati, R. Siegwart (2008), **Incremental Object Part Detection toward Object Classification in a Sequence of Noisy Range Images**, Proc. of The *IEEE International Conference on Robotics and Automation (ICRA)*, February 2008.
[VS08]	S. Vasudevan, R. Siegwart, **Bayesian Space Conceptualisation and Place Classification for Semantic Maps in Mobile Robotics**, *Robotics and Autonomous Systems*, 2008, in press.

These works address, independently, three fundamental aspect of the representation of space. [SPS08] provides the basic functionalities that a metric mapping method (SLAM, [MNTS06]) would need in order to represent the environment. In fact, metric SLAM approaches rely on being able to detect and recognise features in the environment. It is important here to note the difference between the detection and the recognition. Detecting a feature is related to identifying that some perceptive stimuli is worth noticing. On the other hand, recognising a feature is being able to identify that a perceptive stimuli corresponds to something that has been observed previously, or, equally important, to something new. [SPS08] describes a very efficient way to detect and recognise vertical lines in images issued from an omnidirectional cameras. Vertical lines are an omnipresent part of environments build by and for humans, and as a results are very relevant to the development of Robots@Home.

One conceptual level higher, [GHS08] addresses the problem of recognising structured objects (chairs, table, etc..) in the context of mobile robotics. Here again, an important distinction must be done between recognising objects and identifying objects. Identifying objects is related to the identification of a specific instance of an object (model 123242 from a manufacturer's catalogue) whereas recognising object is related to the identification of an object concept (the chair concept in this instance). The work of [GHS08] relies on decomposing objects into parts (chair legs, chair seat, ...), identifying these parts from the data of a range imager and tracking their interrelations to recognise objects. As will be seen in the next paragraph, the hierarchical representation of objects is very similar, in principle to a form of hierarchical representation of space.

Finally, [VS08] adds another level of complexity in the notion of spacial representation: from the occurrence of objects (e.g. from [GHS08]) and from robots localisation (e.g. from a SLAM using [SPS08]) objects are grouped together based on their inter-relations and places are identified based on their possible use. For instance (see fig. 1), a place where three low-chairs are set around a coffee table is likely to be a lounge or a coffee room. On the other hand, a place where a computer screen is set on a table close to an office chair is likely to be an office. These concepts are learnt by example using probabilistic techniques such as clustering or Bayesian network classifiers.

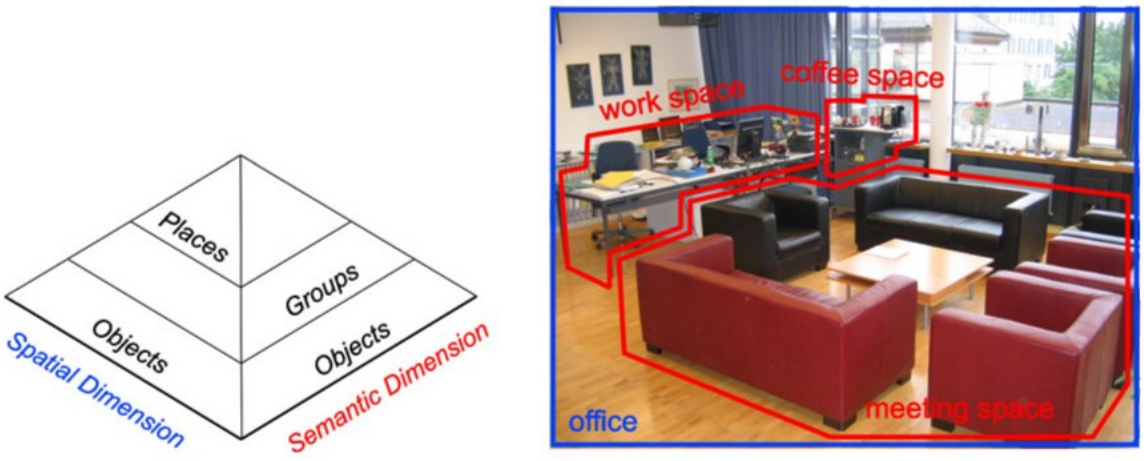


Figure 1: Place conceptualisation using hierarchical models

1.3 Perspectives

The three articles we present in this deliverable represent independent methods for the representation of space.They are suitable for learning, for semantic conceptualisation, and it has been shown[VGS07] that this kind of semantic is compatible with human representations.

However, our next tasks is to bridge the gaps between these approaches in order to create an integrated hierarchical representation of space that can be used to improve the autonomy and efficiency of service robots. A lot is still to be achieved to fulfil this goal, but we are confident the articles we present in this report will be important stepping stones toward it.

Figure 2: Igor, the butler robot. An inspiration for the future robots@home.

References

[GHS08] S Gachter, A Harati, and R Siegwart. Incremental object part detection toward object classification in a sequence of noisy range images. In *Proc. of The IEEE International Conference on Robotics and Automation (ICRA)*, February 2008.

[MNTS06] A Martinelli, V Nguyen, N Tomatis, and R Siegwart. A relative map approach to slam based on shift and rotation invariants. *Robotics and Autonomous Systems*, 2006.

[SPS08] D Scaramuzza, C Pradalier, and R Siegwart. Performance evaluation of a vertical line descriptor for omnidirectional images. In *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[VGS07] S Vasudevan, S Gachter, and R Siegwart. Cognitive maps for mobile robots - perspectives from a user study. In *Proc. of The IEEE International Conference on Robotics and Automation (ICRA), Workshop on Semantic Information in Robotics*, 2007.

[Vin06] Markus Vincze. *robots@home - Description of Work (Annex I)*. 2006.

[VS08] S Vasudevan and R Siegwart. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, 2008.

2 Robust Feature Extraction and Matching for Omnidirectional Images

Authors: D. Scaramuzza, C. Pradalier, R. Siegwart

Year: September 2008

Published in: IEEE International Conference on Intelligent Robots and Systems (IROS)

Performance Evaluation of a Vertical Line Descriptor for Omnidirectional Images

Davide Scaramuzza, Cédric Pradalier, and Roland Siegwart
Autonomous System Lab, ETH Zurich, Switzerland,

***Abstract*— In robotics, vertical lines have been always very useful for autonomous robot localization and navigation in structured environments. This paper presents a robust method for matching vertical lines in omnidirectional images. Matching robustness is achieved by creating a descriptor which is unique and very distinctive for each feature and is invariant to rotation and slight changes of illumination. We characterize the performance of the descriptor on a large image dataset by taking into account the sensitiveness to the different parameters of the descriptor. The robustness of the approach is also validated through a real navigation experiment with a mobile robot equipped with an omnidirectional camera.**

I. Introduction

A. Previous work

Omnidirectional cameras are cameras that provide a 360° field of view of the scene. Such cameras are often built by combining a perspective camera with a shaped mirror. Fixing the camera with the mirror axis perpendicular to the floor has the effect that all world vertical lines are mapped to radial lines on the camera image plane. In this paper, we deal with vertical lines because they are predominant in structured environments.

The use of vertical line tracking is not new in the robotics community. Since the beginning of machine vision, roboticians have been using vertical lines or other sorts of image measure for autonomous robot localization or place recognition.

Several works dealing with automatic line matching have been proposed for standard perspective cameras and can be divided into two categories: those that match individual line segments; and those that match groups of line segments. Individual line segments are generally matched on their geometric attributes (e.g. orientation, length, extent of overlap) [7]–[9]. Some such as [10]–[12] use a nearest line strategy which is better suited to image tracking where the images and extracted segments are similar. Matching groups of line segments has the advantage that more geometric information is available for disambiguation. A number of methods have been developed around the idea of graph-matching [13]–[16]. The graph captures relationships such as “left of”, “right of”, cycles, “collinear with” etc, as well as topological connectedness. Although such methods can cope with more significant camera motion, they often have a high complexity and again they are sensitive to error in the segmentation process.

This work was supported by European grant FP6-IST-1-045350 Robots@Home®

Besides these methods, other approaches to individual line matching exist, which use some similarity measure commonly used in template matching and image registration (e.g. Sum of Squared Differences (SSD), simple or Normalized Cross-Correlation (NCC), image histograms [4]). An interesting approach was proposed in [5]. Besides using the topological information of the line, the authors also used the photometric neighborhood of the line for disambiguation. Epipolar geometry was then used to provide a point to point correspondence on putatively matched line segments over two images and the similarity of the lines neighbourhoods was then assessed by cross-correlation at the corresponding points.

A novel approach, using the intensity profile along the line segment, was proposed in [6]. Although the application of the method was to wide baseline point matching, the authors used the intensity profile between two distinct points (i.e. a line segment) to build a distinctive descriptor. The descriptor is based on affine invariant Fourier coefficients that are directly computed from the intensity profile.

The methods cited above were defined for perspective images but the same concepts have been also used by roboticians in omnidirectional images under certain circumstances. The use of omnidirectional vision even facilitated the task because of the 360° field of view (see [1]–[3]). However, to match vertical lines among different frames only mutual and topological relations have been used (e.g. neighborhood or ordering constraints) sometimes along with some of the similarity measures cited above (e.g. SSD, NCC).

B. Outline

This paper extends our previous work [20], summarized in Sections II and III. In these sections, we describe how we built our robust descriptor for vertical lines, which is unique and very distinctive for each feature and is invariant to rotation and slight changes of illumination. The main contribution of this paper consists in characterizing the performance of the proposed descriptor on a large image dataset that takes into account the sensitiveness to image noise and to other different parameters of the descriptor. Furthermore, we also evaluate the robustness of the approach by tracking vertical lines in a real navigation experiment using a mobile robot equipped with an omnidirectional camera.

The present document is organized as follows. First, we describe our procedure to extract vertical lines (Section II) and build the descriptor (Section III). In Section IV, we provide our matching rules, while the analysis of the

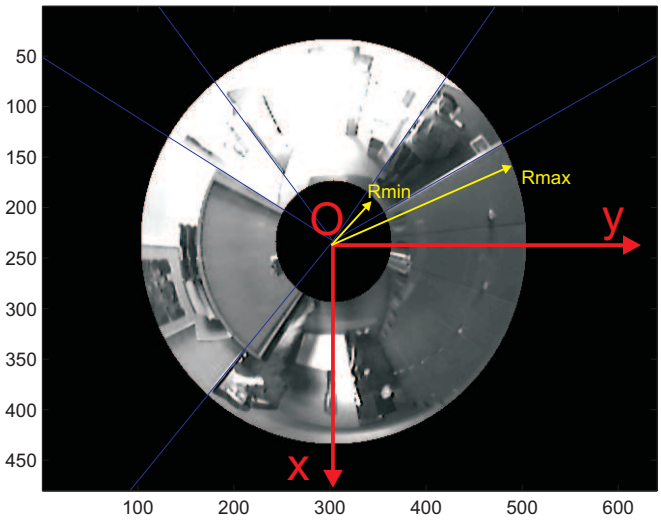


Fig. 1. Extraction of the most reliable vertical features from an omnidirectional image.

performance and the results of tracking are respectively presented in Sections V and VI.

II. Vertical Line Extraction

Our platform consists of a wheeled robot equipped with an omnidirectional camera looking upwards. In our arrangement, we set the camera-mirror system perpendicular to the floor where the robot moves. This setting guarantees that all vertical lines are approximately mapped to radial lines on the camera image plane (Fig. 1) In this section, we detail our procedure to extract prominent vertical lines. Our procedure consists of five steps.

The first step towards vertical line extraction is the detection of the image center (i.e. the point where all radial lines intersect in). As the circular external boundary of the mirror is visible in the image, we used a circle detector to determine the coordinates of the center. In the second step, we apply a Sobel edge detector.

The third step consists in detecting the most reliable vertical lines. To this end, we divide the omnidirectional image into 720 predefined uniform sectors, which give us an angular resolution of 0.5°. By summing up all binary pixels that vote for the same sector, we obtain the histogram shown in Fig. 3. Then, we apply non-maxima suppression to identify all local peaks.

The final step is histogram thresholding. As observed in Fig. 2, there are many potential vertical lines in structured environments. In order to keep the most reliable and stable lines, we put a threshold on the line length. As observed in Fig. 3), we set our threshold equal to 50% of the maximum allowed line length, i.e. $R_{max} - R_{min}$ (for a definition of these parameters see Fig. 1). Obviously, this choice is purely arbitrary and a different criterion could be used depending on the purpose (for instance, one can decide to have always a constant number of lines in each frame).

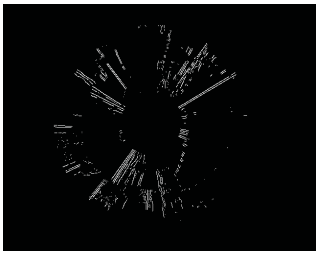


Fig. 2. Edge image of Fig. 1.

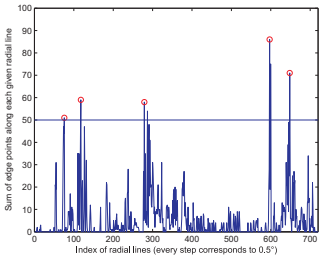


Fig. 3. Number of binary pixels voting for a given orientation angle.

III. Building the Descriptor

In Section IV, we will describe our method for matching vertical lines between consecutive frames while the robot is moving. To make the feature correspondence robust to false positives, each vertical line is given a descriptor which is unique and distinctive for each feature. Furthermore, this descriptor is invariant to rotation and slight changes of illumination. In this way, finding the correspondent of a vertical line can be done by looking for the line with the closest descriptor. In the next subsections, we describe how we built our descriptor.

A. Rotation Invariance

Given a radial line, we divide the space around it into three equal non-overlapping circular areas such that the radius r_a of each area is equal to $(R_{max} - R_{min})/6$ (see Fig. 4). Then, we smooth each area with a Gaussian window with $\sigma_G = r_a/3$ and compute the image gradients (magnitude $\mathbf{M}$ and phase $\mathbf{\Phi}$) within each of these areas.

Concerning rotation invariance, this is achieved by redefining the gradient phase $\mathbf{\Phi}$ of all points relatively to the radial line's angle θ (see Fig. 4).

B. Orientation Histograms

To make the descriptor robust to false matches, we split each circular area into two parts (the left and right across the line) and consider each one individually.

For each side of each circular area, we compute the gradient orientation histogram (Fig. 5). The whole orientation space (from -π to π) is divided into N_b equally spaced bins. In order to decide how much of a certain gradient magnitude m belongs to the adjacent inferior bin b and how much to the adjacent superior bin, each magnitude m is weighted by the factor $(1 - w)$, where

$$w = N_b \frac{\varphi - b}{2\pi}, \tag{1}$$

with φ being the observed gradient phase in radians. Thus, $m(1 - w)$ will vote for the adjacent inferior bin, while mw will vote for the adjacent superior bin.

According to what we mentioned so far, each bin contains the sum of the weighted gradient magnitudes which belong to the correspondent orientation interval. We observed that this weighted sum made the orientation histogram more robust to

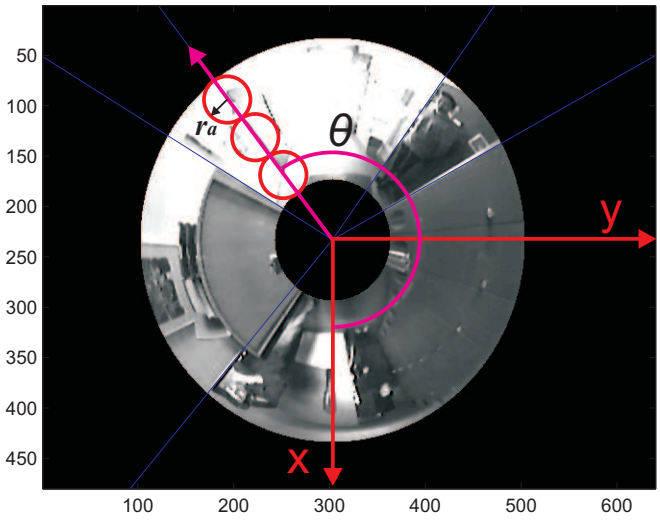


Fig. 4. Extraction of the circular areas. To achieve rotation invariance, the gradient phase $\mathbf{\Phi}$ of all points is redefined relatively to the radial line's angle θ.

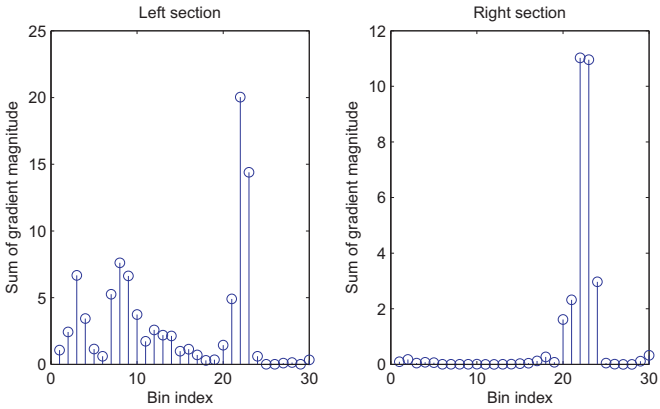


Fig. 5. An example of gradient orientation histograms for the left and right sides of a circular area.

image noise. Finally, observe that the orientation histogram is already rotation invariant because the gradient phase has been redefined relatively to the radial line's angle (Section III-A).

To resume, in the end we have three pairs of orientation histograms:

$$\mathbf{H_1} = [\mathbf{H_{1,L}}, \mathbf{H_{1,R}}], \ \mathbf{H_2} = [\mathbf{H_{2,L}}, \mathbf{H_{2,R}}] \\ \mathbf{H_3} = [\mathbf{H_{3,L}}, \mathbf{H_{3,R}}] \tag{2}$$

where subscripts L, R identify respectively the left and right section of each circular area.

C. Building the Feature Descriptor

From the computed orientation histograms, we build the final feature descriptor by stacking all three histogram pairs as follows:

$$\mathbf{H} = [\mathbf{H_1}, \mathbf{H_2}, \mathbf{H_3}] \tag{3}$$

To have slight illumination invariance, we pre-normalize each histogram vector $\mathbf{H_i}$ to have unit length. This choice relies on the hypothesis that the image intensity changes linearly with illumination. However, non-linear illumination changes can also occur due to camera saturation or due to illumination changes that affect 3D surfaces with different orientations by different amounts. These effects can cause a large change in relative magnitude for some gradients, but are less likely to affect the gradient orientations. Therefore, we reduce the influence of large gradient magnitudes by thresholding the values in each unit histogram vector to each be no larger than 0.1, and then renormalizing to unit length. This means that matching the magnitudes for large gradients is no longer as important, and that the distribution of orientations has greater emphasis. The value of 0.1 was determine experimentally and will be justified in Section V.

Although this is not true in nature, this approximation proved to work properly and will be shown in Sections V and VI.

To resume, our descriptor is an N-element vector containing the gradient orientation histograms of the circular areas. In our setup, we extract 3 circular areas from each vertical feature and use 30 bins for each histogram; thus the length of the descriptor is

$$N = 3areas \cdot 2parts \cdot 30bins = 180 \tag{4}$$

Observe that all feature descriptors are the same length.

IV. Feature Matching

As every vertical feature has its own descriptor, its correspondent in consecutive images can be searched among the features with the closest descriptor. To this end, we need to define a dissimilarity measure (i.e. distance) between two descriptors.

In the literature, several measures have been proposed for the dissimilarity between two histograms $\mathbf{H} = \{h_i\}$ and $\mathbf{K} = \{k_i\}$. These measures can be divided into two categories. The *bin-by-bin* dissimilarity measures only compare contents of corresponding histogram bins, that is, they compare h_i and k_i for all i, but not h_i and k_j for $i \neq j$. The *cross-bin* measures also contain terms that compare non-corresponding bins. Among the *bin-by-bin* dissimilarity measures, fall the Minkoski-form distance, the Jeffrey divergence, the χ^2 statistics, and the Bhattacharya distance. Among the *cross-bin* measures, one of the most used is the Quadratic-form distance. An exhaustive review of all these methods can be found in [17]–[19].

In our work, we tried the dissimilarity measures mentioned above but the best results were obtained using the L_2 distance (i.e. Euclidean distance) that is a particular case of the Minkoski-form distance. Therefore, in our experiments we used the Euclidean distance as a measure of the dissimilarity between descriptors, which is defined as:

$$d(\mathbf{H}, \mathbf{K}) = \sqrt{\sum_{i=1}^{N} |h_i - k_i|^2} \tag{5}$$

By definition of distance, the correspondent of a feature, in the observed image, is expected to be the one, in the

TABLE I

THE PARAMETERS USED BY OUR ALGORITHM WITH THEIR EMPIRICAL VALUES

F_1 = 1.05	F_2 = 0.75	F_3 = 0.8

consecutive image, with the minimum distance. However, if a feature is no longer present in the next image, there will be a closest feature anyway. For this reason, we defined three tests to decide whether a feature correspondent exists and which one the correspondent is. Before describing these tests, let us introduce some definitions.

Let $\{\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_{N_A}}\}$ and $\{\mathbf{B_1}, \mathbf{B_2}, \ldots, \mathbf{B_{N_B}}\}$ be two sets of feature descriptors extracted at time t_A and t_B respectively, where N_A, N_B are the number of features in the first and second image. Then, let

$$D_i = \{d(\mathbf{A_i}, \mathbf{B_j}), j = 1, 2, \ldots, N_B)\} \tag{6}$$

be the set of all distances between a given $\mathbf{A_i}$ and all $\mathbf{B_j}$ $(j = 1, 2, \cdots, N_B)$.
Finally, let $minD_i = \min_i(D_i)$ be the minimum of the distances between given $\mathbf{A_i}$ and all $\mathbf{B_j}$.

A. First test

The first test checks that the distance from the closest descriptor is smaller than a given threshold, that is:

$$minD_i = F_1. \tag{7}$$

By this criterion, we actually set a bound on the maximum acceptable distance to the closest descriptor.

B. Second test

The second test checks that the distance from the closest descriptor is smaller enough than the mean of the distances from all other descriptors, that is:

$$minD_i = F_2 \cdot < D_i > \tag{8}$$

where $< D_i >$ is the mean value of D_i and F_2 clearly ranges from 0 to 1. This criterion comes out of experimental results.

C. Third test

Finally, the third test checks that the distance from the closest descriptor is smaller than the distance from the second closest descriptor:

$$minD_i = F_3 \cdot SecondSmallestDistance, \tag{9}$$

where F_3 clearly ranges from 0 to 1. As in the previous test, the third test raises from the observation that, if the correct correspondence exists, then there must be a big gap between the closest and the second closest descriptor.

Factors F_1, F_2, F_3 are to be determined experimentally. The empirical values used in our experiments are shown in Table I and will be justified in Section V.

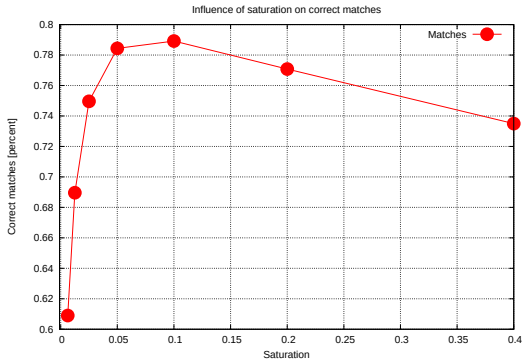


Fig. 6. Influence of saturation on correct matches.

V. PERFORMANCE EVALUATION

In this section, we characterize the performance of our descriptor on a large image dataset by taking into account the sensitiveness to different parameters, that are: image saturation, pixel noise, number of histogram bins, and use of overlapping circular areas. Furthermore, we also motivate the choice of the empirical values for F_1, F_2, and F_3, which are shown in Table I.

1) Ground truth: To generate the ground truth for testing our descriptor, we used a database of 850 omnidirectional pictures that is a subset of the whole video sequence used in Section VI. About 10 verticals were extracted in average from each image. then we matched each feature individually among the all database using the matching method of the previous section. To insure that matching was correct, we visually inspected every single correspondence individually. Correspondent features were labeled with the same ID. The images were taken from the hallway of our department. Figure 13 shows six sample images from our dataset. The images show that the illumination conditions vary strongly. Due to big windows, a mixture of natural and artificial lighting produces difficult lighting conditions like highlights and specularities. With regard to the viewpoint change, the maximum camera displacement between two views of the same vertical was about 5 meters, while the average displacement was around 2 meters.

2) Image saturation: As we mentioned in Section III-C, we threshold the values of the histogram vectors to reduce the influence of image saturation. The percent of correct matches for different threshold values is shown in Fig. 6. The results show the percent of verticals that find a correct match to the single closest neighbor among the all database. As the graph shows, the maximum percent of correct matches is reached when using a threshold value of 0.1. In the remainder of this paper, we will always use this value.

3) Image noise: The percent of correct matches for different amounts of gaussian image noise (from 0% to 10%) is shown in Fig. 7. Again, the results show the percent of correct matches found using the single nearest neighbor among the all database. As this graph shows, the descriptor is resistant to even large amount of pixel noise.

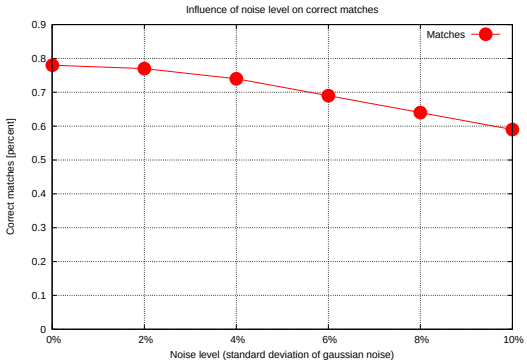


Fig. 7. Influence of noise level (%) on correct matches. The correct matches are found using only the nearest descriptor in the database.

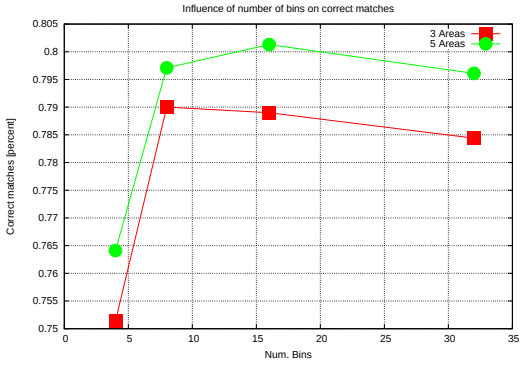


Fig. 8. Influence of number of bins on correct matches.

4) Histogram bins and circular areas: There are two parameters that can be used to vary the complexity of our descriptor: the number of orientations, N_b, in the histograms, and the number of circular areas. Although in the explanation of the descriptor we used 3 non overlapping circular areas, we evaluated the effect of using 5 overlapping areas with 50% overlap between two circles. The results are shown in Fig. 8. As the graph shows, there is a slight improvement in using 5 overlapping areas (the amelioration is only 1%). Also, the performance is quite similar using 8, 16, or 32 orientations in the histograms. Following this considerations, the best choice would seem to use 3 areas and 8 histograms bins in order to reduce the dimension of the descriptor. Conversely, as in this graph the percent of correct matches is found only using the nearest closest descriptor, the best matching results, when using the rules of Section IV, are obtained with 32 orientations. Thus, in our implementation we used 3 areas and 32 histogram bins.

5) Matching rules: Figure 9 shows the Probability Density Function (PDF) for correct and incorrect matches in terms of the distance to the closest neighbor of each keypoint. In our implementation of the first rule, we chose $F_1 = 1.05$. As observed in the graph, by this choice we reject all matches in which the distance to the closest neighbor is greater than 1.05, which eliminates 50% of the false matches while discarding less than 5% of correct matches.

Similarly, Fig. 10 shows the PDFs in the terms of the ratio

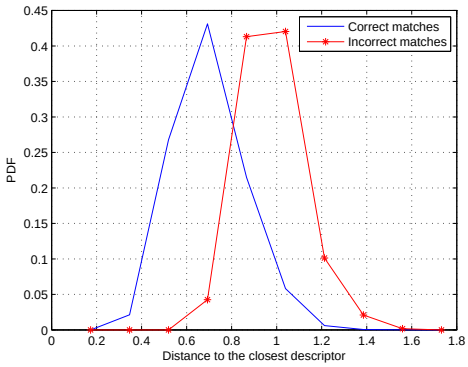


Fig. 9. The probability density function that a match is correct according to the first rule.

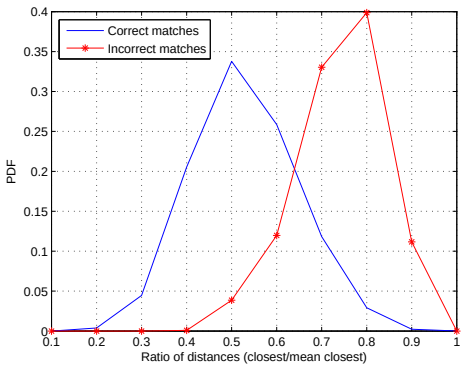


Fig. 10. The probability density function that a match is correct according to the second rule.

of closest to average-closest neighbor of each keypoint. In our implementation of the second rule, we chose $F_2 = 0.75$. As observed in the graph, by this choice we reject all matches where the ratio between the closest neighbor distance and the mean of all other distances is greater than 0.75, which eliminates 45% of the false matches while discarding less than 8% of correct matches.

Finally, Fig. 11 shows the PDFs in terms of the ratio of closest to second-closest neighbor of each keypoint. In our implementation of the third rule, we chose $F_3 = 0.8$; in this way we reject all matches in which the distance ratio is greater than 0.8, which eliminates 92% of the false matches while discarding less than 10% of correct matches.

VI. Experimental Results

In our experiments, we adopted a mobile robot with a differential drive system endowed of encoder sensors on the wheels. Furthermore, we equipped the robot with an omni-directional camera consisting of a KAIDAN 360 One VR hyperbolic mirror and a SONY CCD camera the resolution of 640×480 pixels. In this section, we show the performance of our feature extraction and matching method by capturing pictures from our robot in a real indoor environment.

The robot was moving at about $0.15\ m/s$ and was acquiring frames at $3\ Hz$, meaning that during straight paths the

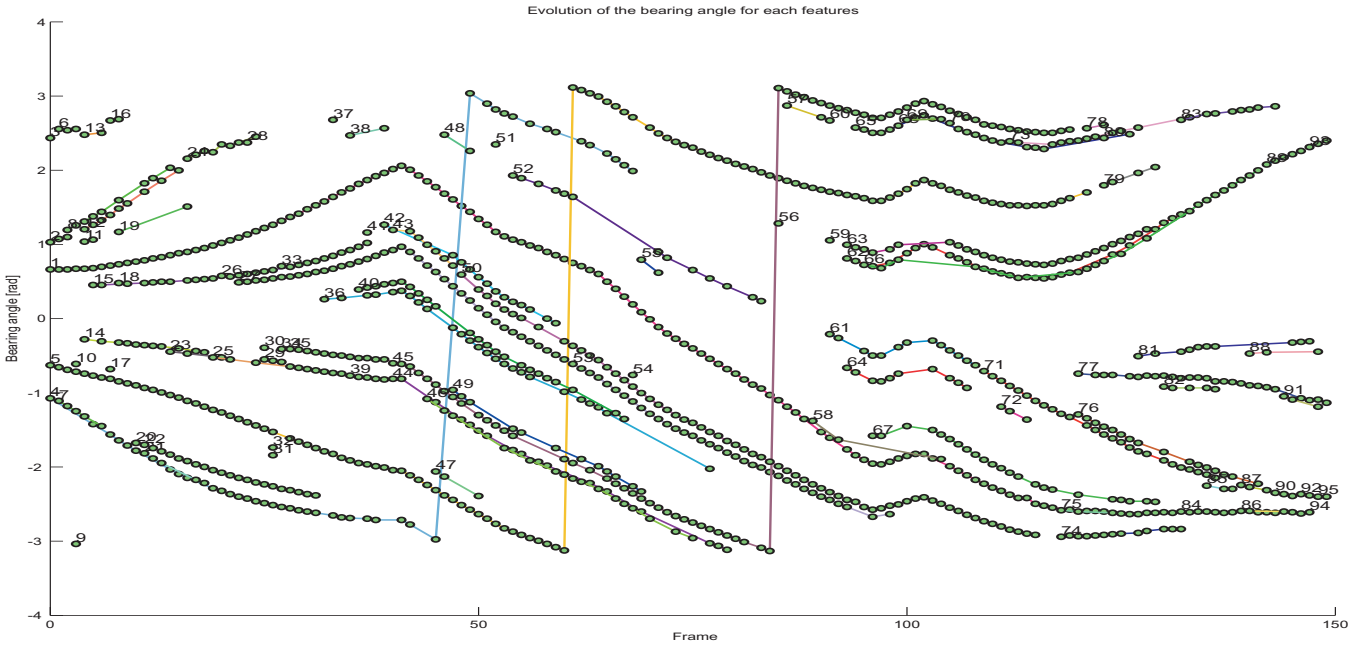


Fig. 12. Feature tracking during the motion of the robot. In y-axis is the angle of sight of each feature and in the x-axis the frame number. Each circle represents a feature detected in the observed frame. Lines represent tracked features. Numbers appear only when a new feature is detected.

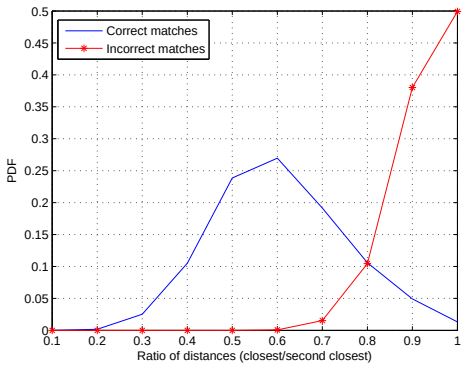


Fig. 11. The probability density function that a match is correct according to the third rule.

traveled distance between two consecutive frames was $5\ cm$. The robot was moved in the hallway of our institute and 1852 frames were extracted during the whole path. Figure 13 shows six sample images from the dataset.

The result of feature tracking is shown only for the first 150 frames in Fig. 12. The graph shown in Fig. 12 was obtained using only the three matching rules described in Sections IV-A, IV-B, IV-C. No other constraint, like mutual relations, has been used. This plot refers to a short path of the whole trajectory while the robot was moving straight (between frame no. 0 and 46), then doing a $180°$ rotation (between frame no. 46 and 106), and moving straight again. As observed, most of the features are correctly tracked over the time. Indeed, most of the lines appear smooth and homogeneous. The lines are used to connect features that belong to the same track. When a new feature is detected, this feature is given a label with progressive numbering and a new line (i.e. track) starts from it. In this graph, there are three false matches that occur at the points where two tracks intersect (e.g. at the intersection between tracks no. 1 and 58, between track no. 84 and 86, and between track no. 65 and 69). Observe that the three huge jumps in the graph are not false matches; they are only due to the angle transition from $-\pi$ to π.

Observe that our method was able to match features even when their correspondents were not found in the previous frames. This can be seen by observing that sometimes circles are missing on the tracks (look for instance at track no. 52). When a correspondence is not found in the previous frame, we start looking into all previous frames (actually up to twenty frames back) and stop when the correspondence is found.

If you examine the graph, you can see that some tracks are suddenly given different numbers. For instance, observe that feature no. 1 - that is the fist detected feature and starts at frame no. 0 - is correctly tracked until frame no. 120 and is then labeled as feature no. 75. This is because at this frame no correspondence was found and then the feature was labeled as a new entry (but in fact is a false new entry). Another example is feature no. 15 that is then labeled as no. 18 and no. 26. By a careful visual inspection, you can find only a few other examples of false new entries. Indeed, tracks that at a first glance seem to be given different numbers, belong in fact to other features that are very close to the observed one.

After visually inspecting every single frame of the whole video sequence (composed of 1852 frames), we found 37 false matches and 98 false new entries. Comparing these errors to the 7408 corresponding pairs detected by the

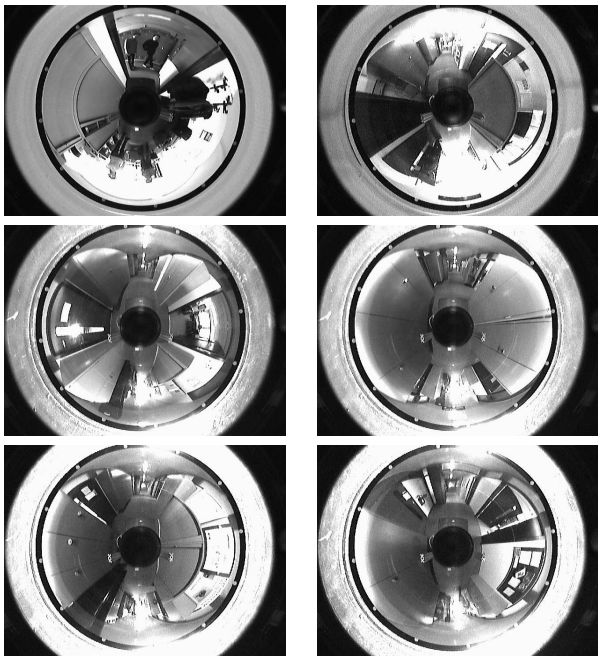


Fig. 13. Omnidirectional images taken at different locations.

algorithm over the whole video sequence, we had 1.8% of mismatches. Furthermore, we found that false matches occurred every time the camera was facing objects with repetitive texture. Thus, ambiguity was caused by the presence of vertical elements which repeat almost identical in the same image. On the other hand, a few false new entries occurred when the displacement of the robot between two successive images was too large. However, observe that when a feature matches with no other feature in previous frames, it is better to believe this feature to be new rather than commit a false matching.

As we already mentioned above, the results reported in this section were obtained using only the three matching rules described in Sections IV-A, IV-B, IV-C. Obviously, the performance of tracking could be further improved by adding other constraints like mutual relations among features.

VII. Conclusion

In this paper, we presented a robust method for matching vertical lines among omnidirectional images.The basic idea to achieve robust feature matching consists in creating a descriptor which is unique and distinctive for each feature. Furthermore, this descriptor is invariant to rotation and slight changes of illumination. We characterized the performance of the descriptor on a large image dataset by taking into account the sensitiveness to the different parameters of the descriptor. The robustness of the approach is also validated through a real navigation experiment with a mobile robot equipped with an omnidirectional camera. The performance of tracking was very good as many features were correctly detected and tracked over long time. Furthermore, because the results were obtained using only the three matching rules described in Section IV, we expect that the performance would be notably improved by adding other constraints like mutual relations among features.

References

[1] Brassart, E., Delahoche, L., Cauchois, C., Drocourt, C., Pegard, C., Mouaddib, E.M., Experimental Results got with the Omnidirectional Vision Sensor: SYCLOP, International workshop on omnidirectional vision (OMNIVIS 2000), 2000.

[2] Yagi, Y., and Yachida, M., Real-Time Generation of Environmental Map and Obstacle Avoidance Using Omnidirectional Image Sensor with Conic Mirror, CVPR'91, pp. 160-165, 1991.

[3] D. Prasser, G. Wyeth, M. J. Milford (2004b), Experiments in Outdoor Operation of RatSLAM, Australian Conference on Robotics and Automation, Canberra Australia, 2004.

[4] R. Gonzalez, R. Woods, Digital Image Processing, Addison Wesley, Prentice Hall, ed. 2, ISBN: 0201180758, 2002.

[5] Baillard, C., Schmid, C., Zisserman, A., and Fitzgibbon, A., Automatic line matching and 3D reconstruction of buildings from multiple views, SPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery, IAPRS Vol.32, Part 3-2W5, pp. 69-80, 1999.

[6] D. Tell and S. Carlsson, Wide baseline point matching using affine invariants computed from intensity profiles, In Proceedings of the European Conference Computer Vision, pp. 814828, Dublin, Ireland, 2000.

[7] Medioni, G. and Nevatia, R., 1985. Segment-based stereo matching. Computer Vision, Graphics and Image Processing 31, pp. 218.

[8] Ayache, N., 1990. Stereovision and Sensor Fusion. MITPress.

[9] Zhang, Z., 1994. Token tracking in a cluttered scene. Image and Vision Computing 12(2), pp. 110120.

[10] Crowley, J. and Stelmazyk, P., 1990. Measurement and integration of 3d structures by tracking edge lines. In: Proc. ECCV, pp. 269280.

[11] Deriche, R. and Faugeras, O., 1990. Tracking line segments. In: Proc. ECCV, pp. 259267.

[12] Huttenlocher, D. P., Klanderman, G. A. and Rucklidge, W. J., 1993. Comparing images using the Hausdorff distance. IEEE T-PAMI.

[13] Ayache, N. and Faugeras, O., 1987. Building a consistent 3D representation of a mobile robot environment by combining multiple stereo views. In: Proc. IJCAI, pp. 808810.

[14] Horaud, R. and Skordas, T., 1989. Stereo correspondence through feature grouping and maximal cliques. IEEE TPAMI 11(11), pp. 11681180.

[15] Gros, P., 1995. Matching and clustering: Two steps towards object modelling in computer vision. Intl. J. of Robotics Research 14(6), pp. 633642.

[16] Venkateswar, V. and Chellappa, R., 1995. Hierarchical stereo and motion correspondence using feature groupings. IJCV pp. 245269.

[17] M.M. Rahman, P. Bhattacharya, and B.C. Desai, Similarity Searching in Image Retrieval with Statistical Distance Measures and Supervised Learning, in Pattern Recognition and Data Mining, pp. 315-324, 2005.

[18] Y. Rubner, C. Tomasi, and L. J. Guibas, .The earth mover's distance as a metric for image retrieval,. International Journal of Computer Vision, vol. 40, no. 2, pp. 99-121, 2000.

[19] Y. Rubner et al, .Empirical evaluation of dissimilarity measures for color and texture,. Computer Vision and Image Understanding, vol. 84, no. 1, pp. 25-43, 2001.

[20] Scaramuzza, D., Criblez, N., Martinelli, A. and Siegwart, R., Robust Feature Extraction and Matching for Omnidirectional Images, Proceedings at the 6th International Conference on Field and Service Robotics (FSR 2007), Chamonix, France, July 2007.

[21] Y. Bar-Shalom and T.E. Fortmann, Tracking and data association, mathematics in science and engineering, vol. 179, Academic Press, 1988.

3 Incremental Object Part Detection toward Object Classification in a Sequence of Noisy Range Images

Authors: S. Gachter, A. Harati, R. Siegwart

Year: May 2008

Published in: IEEE International Conference on Robotics and Automation (ICRA)

Incremental Object Part Detection toward Object Classification in a Sequence of Noisy Range Images

Stefan Gächter, Ahad Harati, and Roland Siegwart
Autonomous Systems Lab (ASL)
Swiss Institute of Technology, Zurich (ETHZ)
8092 Zurich, Switzerland
{gaechter, harati, siegwart}@mavt.ethz.ch

***Abstract*— This paper presents an incremental object part detection algorithm using a particle filter. The method infers object parts from 3D data acquired with a range camera. The range information is quantized and enhanced by local structure to partially cope with considerable measurement noise and distortion. The augmented voxel representation allows the adaptation of known track-before-detect algorithms to infer multiple object parts in a range image sequence even when each single observation does not contain enough information to do the detection. The appropriateness of the method is successfully demonstrated by two experiments for chair legs.**

I. Introduction

In recent years, a novel type of range camera to capture 3D scenes emerged on the market. One such camera is depicted in figure 1. The measurement principle is based on time-of-flight using modulated radiation of an infrared source. Compared with other range sensors [1], range cameras have the advantage to be compact and at the same time to have a measurement range of several meters, which makes them suitable for indoor robotic applications. Further, range cameras provide an instant single image of a scene at a high frame rate though with a lower image quality in general [2]. The 3D information acquired with a range camera is strongly affected by noise, outliers and distortions, because of its particular measurement principle using a CMOS/CCD imager [3], [4]. This makes it difficult to apply range image algorithms developed in the past. Hence, the goal of this paper is to present an object part detection method adapted to range cameras.

Object parts – components with simple geometry – are quite proper features for object classification based on geometric models [5], [6], [7]. This approach can account for different views of the same object and for variations in structure, material, or texture of the objects of the same kind. The reason is that more or less the decomposition of the objects into its parts remains unchanged. The majority of the currently available approaches in the field of object classification are appearance based, which makes them very sensitive to the mentioned variations.

In general, range image algorithms depend on the robust estimation of the differential properties of object surfaces [8].

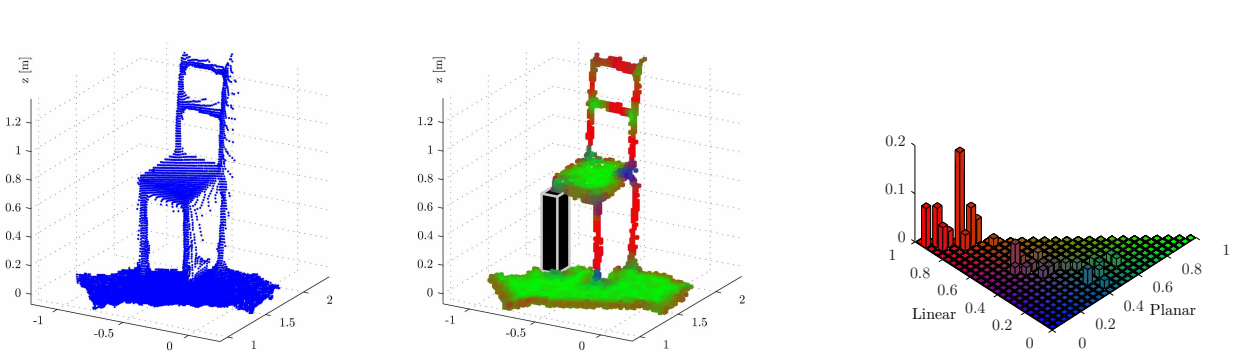
This work was partially supported by the EC under the FP6-IST-045350 robots@home project.

Fig. 1. SR-3000 Range Camera.

Given the noisy nature of images of range cameras, this can only be obtained with high computational cost. However, the detailed reconstruction of object surface geometry is not necessary for part based object classification as long as the parts are detected. On the other hand, object parts can be represented properly by bounding volumes [7], because the overall structure of an object part is more important and informative than the details of its shape or texture. For example, the concept of a chair leg is more related to its stick like structure than whether it is wooden or metallic, of light or dark color, round or square.

However, segmentation of range images into object parts remains the most challenging stage. Because of the low signal-to-noise ratio of the mentioned sensor, this is a particularly difficult problem. Using an incremental algorithm operating on several range images, can improve the performance. In fact, it is possible to skip segmentation and track hypothetical parts in the scene. This is a common approach in radar applications, where a target has to be jointly tracked and classified in highly noisy data [9], [10]. Hence, for each part category, a classifier is considered which incrementally collects the evidences from the sequence of range images and tracks the hypothetical parts. Therefore, the object part detection becomes the sequential state estimation process for multiple bounding-boxes at potential poses in the three-dimensional space. This is realized in the framework of a particle filter [10], which can cope with different sources of uncertainty, among them scene registration errors.

The contribution of this work lies in bringing well established algorithms from classification, tracking and state-estimation to the framework of object classification. In addition, to the best of our knowledge, this is a first work which addresses object part detection using a range camera. The presented work here paves the way toward incremental



(a) Point Cloud (b) Voxel Set (c) Shape Factor Histogram

Fig. 2. (a) Single point cloud and (b) a quantized version of a sequence of five registered range images at step $k = 25$ along with (c) the shape factor histogram of the right front leg. The bounding-box in (b) encloses all voxels that are considered to compute the histogram. The colors indicate the shape factors: red for *linear* like, green for *planar* like, and blue for *spherical* like local structures. Refer to the remaining part of the paper for the computational details.

part based object classification in the field of indoor mobile robotics. The approach presented here is quite general in handling different object parts with simple geometry. However, through out this paper, a chair leg is chosen as an example part to demonstrate the method.

II. Related Work

Part extraction from range images is a long standing issue in structure based object recognition and classification. Seminal work has been done by [11], where algorithms are presented that infer objects from surface information. Object parts are represented by surface patches. In the present work, bounding-boxes are adopted, which are more abstract volumetric representation than commonly used parametrical models based on surfaces [12], [13]. In addition, the quantization is achieved by the voxel representation which is related to occupancy grids, but less computationally intensive.

In [14], a method to capture local structure in range images is presented in order to classify natural terrain. In the present work, local structure is captured in the same way with shape factors. However, shape factors are calculated based on the voxel representation here.

The object part detection algorithm evolves from the work done in [15]. They developed a method for joint detection and tracking of multiple objects described by color histograms. Color-based tracking is a well researched topic in the vision community, see for example [16] and [17]. Here, these techniques are taken as inspiration to detect object parts in quantized point clouds using shape factor as color.

III. Range Image Quantization

One of the smallest range cameras in the market is the SR-3000 made by [18], see figure 1. For the work presented here, the SR-2 of the same manufacturer is used, which exhibits similar measurement performance for indoor applications. The SR-2 has a resolution of 124×160 pixels with maximum measurement range of 7.5 m. The intrinsic and extrinsic camera parameters are respectively calibrated based on the methods explained in [3] and [19].

Despite the calibration, the range image remains affected by noise, outliers and distortions. Main reasons include low emission power, scattering, and multiple reflections. A sample observation of a scene with a chair is shown in figure 2(a). Thus, a single observation does not contain enough information to detect object parts. On the other hand, registering different views over long runs accumulates alignment errors. Therefore, a sliding window containing the most recent five observations is considered. The corresponding point clouds are registered and quantized into a cubic voxel space with voxel size of 2 cm to reduce the computational burden. Voxels containing less than five points are neglected as outliers, see figure 2(b).

IV. Object Part Detection

The structural variability of objects is strongly related to the number and type of parts and their physical relationship with each other. Such relationships can be encoded within a probabilistic grammar in order to perform object classification [7]. Towards such an approach, object parts are modeled as probabilistic bounding-boxes to handle uncertain measurements of the range camera.

A bounding-box is a cuboid defined by the center point and the span length. The probabilistic extension assumes these parameters as random variables. Here, particle filter is used to estimate them. Each particle encodes hypothetical positions and extensions of some object parts, i.e. their bounding-boxes. The evolution of the particles over time enables the simultaneous detection and tracking of the object parts. Gradually, particles with realistic hypotheses survive, whereas the others die off. The fitness of each particle – the resampling weight – is obtained based on the shape factor histograms calculated in the image regions defined by the corresponding bounding-boxes.

A. Shape Factor

The shape factors characterize the local part structure by its linear, planar, or spherical likeliness. They are calculated for each voxel using its surrounding spatial voxel distribution

by the decomposition of the distribution into the principal components – a set of ordered eigenvalues and -vectors. Here, the standard principal component analysis is used.

In the literature, different methods are presented on how to compute the shape factors. In [20] and [21] a tensor representation is proposed to infer structure from sparse data. For the present work, the same scheme is used with a different normalization:

$$\begin{aligned} r_l &= \frac{\lambda_1 - \lambda_2}{\lambda_3 + \lambda_2 + \lambda_1}, \\ r_p &= \frac{2(\lambda_2 - \lambda_3)}{\lambda_3 + \lambda_2 + \lambda_1}, \\ r_s &= \frac{3\lambda_3}{\lambda_3 + \lambda_2 + \lambda_1}. \end{aligned} \tag{1}$$

where λ_i are the ordered eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ obtained by the decomposition of the spatial voxel distribution. r_l, r_p, and r_s express local similarity to linear, planar, and spherical shapes respectively. Here, the shape factors are normalized by the sum of the eigenvalues [22] so that each lies in the range of $[0, 1]$ and their sum is one: $r_l + r_p + r_s = 1$. Another normalization scheme is to use the maximum eigenvalue [21]. The shape factors can also be defined by reasoning on the volume spanned by the eigenvalues [23]. Which of the shape factor computation methods is used, depends largely on their ability to characterize voxels distinctively according to the object structure at hand.

Figure 2(b) depicts a shape factor colored voxel set of a chair, where for each voxel the shape factor was computed according to (1). The computation was done within a neighborhood window of size $11\times11\times11$ voxels defining the scale of the local structure. As it is visible, this method correctly classifies the local structure; legs appear as linear, seat and back as planar, and joints as spherical structures.

B. Histogram as Feature Vector

The shape factor distribution in the region of interest defined by the bounding-box is approximated by a histogram to obtain a unique feature vector that models an object part. This approach is inspired by the work done in [16], where color histograms are used to track objects. In the present application, histograms have the advantage to be robust against the structural variability of object parts: rotation, partial occlusion, and scale have little effect on the model. In addition, the computational cost of histograms is modest.

Since the three positive elements of the shape factor sum up to one, they are constrained to a triangle in 3D space. Thus, it is sufficient to consider only two elements to populate a 2D histogram with $N_t = \frac{1}{2}(N_b^2 + N_b)$ bins, where the histogram shape is approximated by a triangular matrix of size N_b. Figure 2(c) depicts the shape factor histogram of the bounding-box volume enclosing the leg in figure 2(b). It is clearly visible that linear shape factors dominate indicating the general stick like structure of the object part. Because the number of bins N_t already becomes large for a small N_b, dimensionality reduction is applied on the 0 as feature vector. The dimensionality reduction is done by standard principal component analysis of the training set retaining the dimensions covering 95 % of the feature distribution mass.

Finally, six simple geometric features are added to the feature vector to account for the occupancy and eccentricity of the voxel distribution in the bounding-box.

C. Support Vector Classifier

In order to judge, if an object part in question is likely to belong to a certain class it is necessary to evaluate a quality measure. This can be done by computing a distance between a template and the generated feature vector. This is commonly done in color based tracking [16]. However, template matching might not be discriminative enough to cover an entire class of an object part. Using a classifier learned on a large amount of training data often results in a better detection performance. A suitable training method is the support vector machine (SVM), because it is less prone to overfitting, is applicable on high dimensional features, and the resulting classifier allows the estimation of meaningful posterior probabilities. In the present work, a support vector classifier with a polynomial kernel is trained using the framework provided by [24]. A training set of 2340 samples is generated. An equal number of positive and negative samples are used to avoid any bias in the learning. The 1170 positive samples of chair leg are manually extracted from voxel images from different views of twelve different chairs. The 1170 bad samples are randomly selected from a stream of voxel images containing background clutter or non-leg parts.

D. Incremental State Estimation

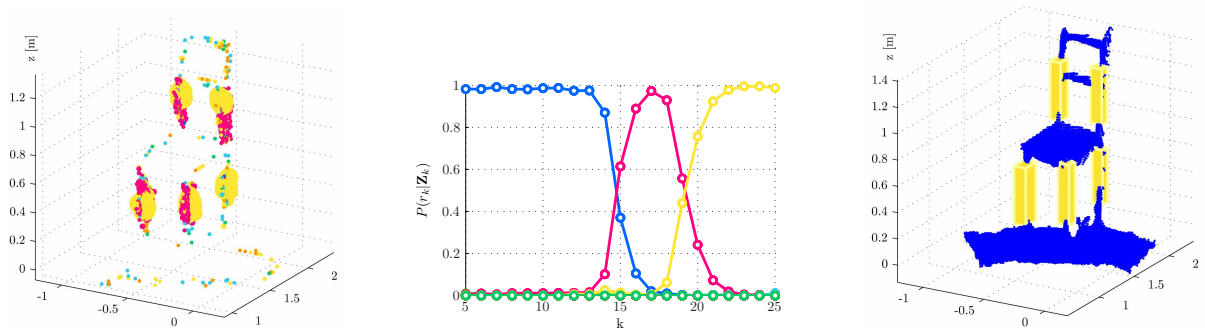
The aim is to incrementally detect object parts modeled by a bounding-box in a sequence of voxel images. The detection algorithm typically has to handle multiple object parts of the same type. Thus, the problem can be stated formally as follows:

$$p(\mathbf{y}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{y}_k|\mathbf{y}_{k-1}) p(\mathbf{y}_{k-1}|\mathbf{Z}_{k-1}) d\mathbf{y}_{k-1} \tag{2}$$

$$p(\mathbf{y}_k|\mathbf{Z}_k) \propto p(\mathbf{z}_k|\mathbf{y}_k) p(\mathbf{y}_k|\mathbf{Z}_{k-1}), \tag{3}$$

where $\mathbf{y}_k = [R_k, \mathbf{x}_{1,k}^\mathsf{T} \ldots \mathbf{x}_{r_k,k}^\mathsf{T}]^\mathsf{T}$. $\mathbf{y}_k$ is the augmented state, which contains the current estimate of number of object parts present in the view R_k and their bounding-box parameters $\mathbf{x}_{i,k}$ at step k. This incremental state estimation can be implemented by a particle filter. Here, the algorithm presented in [15] is used; an extension of the traditional particle filter [25], capable of tracking multiple targets. However, the transition and observation models have been adapted where necessary.

1) Transition Model: The object part number R_k is modeled by a Markov chain with a predefined transition matrix, where the state value at step k is a discrete number $r_k = \{0, \ldots, M\}$ with M being the maximum number of parts expected in each view, set to 8 here. The Markov chain defines three possible cases on how the number of parts can evolve over time: the number remains *unaltered*, *increases*, or *decreases* from step $k-1$ to k.



(a) Observation Density (b) Part Presence Probability (c) Estimated Parts

Fig. 3. Results for the first experiment. (a) Particle distribution at step k = 20 during the update. Particle size indicates its weight. (b) Evolution of the part presence probability over time. (c) Estimated object parts at step k = 25. The color indicates the number of hypothetical parts encoded by a particle: blue for 3, magenta for 4, and yellow for 5 states. Other colors indicate higher or lower number of states, where the maximum number of states is eight.

When the number of parts remains unaltered, their states are assumed to be affected by a process noise, which takes into account the measurement deficiencies and registration errors. Therefore, the proposal distribution for the bounding-box parameters is given by $p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1}) = \mathcal{N}(\mathbf{x}_{i,k-1}, \mathbf{C}_u)$, where $\mathbf{C}_u$ is the covariance matrix assumed to be diagonal. In the experiments of this paper, for a chair leg, the diagonal entries for the bounding-box position are set to 1, 1, and 9 cm^2 and for the extension to 49, 49, and 64 mm^2, considering more uncertainty along the vertical direction.

When the number of parts decreases, r_k hypothetical parts are selected at random from the possible r_{k-1} with equal probability. The selected parts parameters are then affected by the process noise.

The crucial case is when the number of parts increases. Then, the current state of the particle has to be augmented by additional elements. For the r_{k-1} parts that continue to exist, again the process noise is considered. For the $r_k - r_{k-1}$ new hypothetical parts, the bounding-box position is uniformly sampled from occupied voxels, which have proper shape factors. In addition, to preserve consistency of different instances of a part, an intersection test is performed [26]. Hence, the initialized bounding-boxes for each particle keep a certain distance, here 10 cm.

2) Observation Model: The observation likelihood function generates the importance weights used to incorporate the measurement information $\mathbf{z}_k$ in the particle set. Since the parts have to be detected from various view angles out of sparse and noisy data, the observation model is a non-linear function of the part state and measurement noise. As in [27], instead of using a generative observation model, which is common in a Bayesian estimation framework, a discriminative one is selected, namely the learned support vector classifier presented previously.

In the detection framework, the observation likelihood is usually defined as a ratio of the probability that an object part is present to the probability of its absence. This is equivalent to the ratio of the classification probabilities computed with the learned classifier. Assuming that the classification can be done independently for each hypothetical object part, the observation likelihood for each particle is given by

$$L(\mathbf{y}_k) = \prod_{i=1}^{r_k} \frac{p(\mathbf{z}_k|\mathbf{x}_{i,k})}{1 - p(\mathbf{z}_k|\mathbf{x}_{i,k})}, \tag{4}$$

where $p(\mathbf{z}_k|\mathbf{x}_{i,k})$ is the classification probability for part i. Considering this probability as a distance $a_{i,k}$ in the range of $[0, 1]$ and an exponential function to compute a similarity measure, the unnormalized importance weight $\tilde{\pi}_k$ for each particle is computed as:

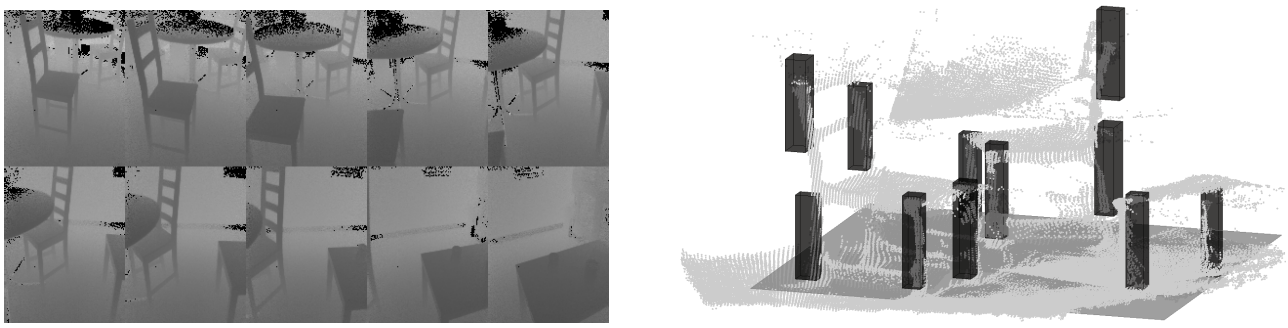
$$\tilde{\pi}_k = \begin{cases} 1, & R_k = 0 \\ \exp\left(-\frac{1}{b}\sum_{i=1}^{r_k}(1 - 2a_{i,k}) + r \cdot c\right), & R_k > 0 \end{cases} \tag{5}$$

where b is a parameter to adjust the observation sensitivity and c accounts for the a priori knowledge. Here, b = 0.21 and c = 0.21 are used, both determined experimentally.

When no a priori knowledge is considered, the weighting scheme defined above has a pivoting point for a classification probability equal to 0.5; meaning that particles with large number of hypothetical object parts and a probability only slightly greater than 0.5 are favored over particles with small number of parts but a high probability. Hence, the object part detection algorithm has an inherent tendency for exploration.

V. Experiments

The above discussed incremental object part detection method is exemplified by the detection of chair legs. Chair legs in reality are designed with various shapes and tilt angles. Here, they are modeled by a vertical bounding-box defined by its center point position $\mathbf{s} = [s_x, s_y, s_z]^\mathsf{T}$ and its extension $\mathbf{t} = [t_x, t_y, t_z]^\mathsf{T}$ to cover the overall shape for the class of chair legs. With the assumption of upright chairs, the rotations are neglected since they are not properly captured by the range camera. However, the effect of such rotations are generally small and pose minor variations with respect to the overall structure. For other parts of a chair, such as seat and back, the rotation around the z-axis has to be considered in the state.



(a) (b)

Fig. 4. Second experiment at cafeteria. (a) Samples of range image sequence at an interval of 50 steps starting at step 0 and ending at 450. (b) Detected stick like parts in the scene with a round dining table, two chairs and a coffee table. The black bounding-boxes indicate the estimated positions and extensions of stick like parts. Only two point clouds are depicted.

Two experiments are performed with the range camera mounted on a robot at height of about 1.1 m facing downward with a tilt angle of about 15°. In the first experiment, only one chair is in the scene while in the second experiment the robot is observing a round dining table, two chairs and a coffee table in the cafeteria of our lab. In both experiments, the robot slowly approaches the objects in the scene recording range images and odometry at about 2 Hz. Totally 200 and 450 range images are captured in the first and second experiment respectively. Because of occlusions and the narrow field of view of the camera, the number of hypothetical chair legs in the view varies considerably, see 4(a). Hence, the algorithm should dynamically adapt to what is present in the view.

Considering the complexity of the scenes, the number of particles is set to 750, which is rather low because of the intelligent initialization scheme. The outcome of the first experiment is summarized in figure 3. Figure 3(a) shows the observation density of the particle filter at step k = 20. The weights are represented by the size of the depicted particles. It can be seen that the particles at the chair legs and back columns are bigger than the ones at the seat. Therefore, the weighting based on the support vector classifier is successful. On the other hand in this figure, particles with different colors represent different number of hypothetical legs. Therefore, a competition between red and yellow particles corresponding to four and five legs is taking place at this step. This is a result of the explorative behavior of the transition matrix and necessary for discovering new parts, which may enter the scene. The same fact is depicted more properly in figure 3(b) versus time, where the probability of the number of object parts present in the view is approximated by the ratio of the number of particles sharing the same r value to the total number of particles. At step k = 25, three legs and two columns of the back support are successfully detected as can be seen in figure 3(c).

In the second experiment with a more realistic scenario, the robot is faced with the challenge of object part detection in the cafeteria. In figure 4(b), the estimated object parts are depicted overlaid with two original point clouds. Depicted are the hypothetical legs with the probability larger than 0.5. The observed deviation between the estimated bounding-boxes and the real parts are mainly because only two point clouds are depicted. If the whole 450 point clouds are considered together, the errors of range camera and odometry result in a messy accumulation of points where no geometry is detectable. As mentioned, this fact is one of the main motivations in using an incremental estimation method. In figure 5, upper graph, depicts the probability of the number of object parts present in the view. As can be seen, the

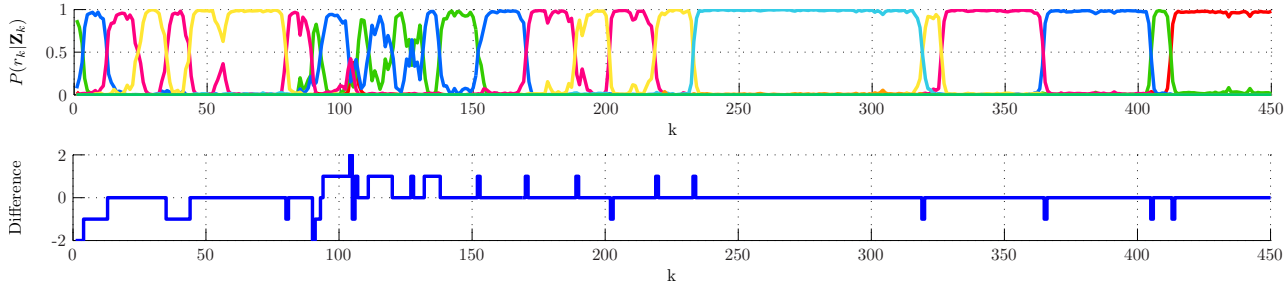


Fig. 5. Second experiment at cafeteria. In the upper figure is depicted the evolution of the part presence probability over time. The color indicates the number of hypothetical parts encoded by a particle: red for 1, green for 2, blue for 3, magenta for 4, yellow for 5, and cyan for 6 stick like parts. Other colors indicate higher or lower number of parts with the maximum of eight. In the lower figure is depicted the difference between the detected and the actual number of hypothetical legs in the view.

probabilities oscillate where the scene changes considerably – between step 80 and 140 – and the algorithm has to deal with many appearing and disappearing parts. This is also evident in figure 5, lower graph, where the difference between the detected and the actual number of hypothetical legs in the view is depicted. The actual number of parts is determined by visual inspection of the voxel images. In the end, all leg like parts are detected and no false positive remains.

VI. Conclusion

This paper presented an algorithm for object part detection using an extended particle filter as an estimation engine with a support vector classifier based observation function. The algorithm can handle multiple parts of the same class and deal with different sources of uncertainties. The provided experimental results show that using a limited number of particles it is possible to successfully estimate the position and extension of multiple chair legs – an exemplary object part – in an incremental process.This proves the accomplishment of the primary goal: accumulation of information in a sequence of noisy and sparse observations.

However, the method needs further testing and improvements for its robust application in robotics. First, the detection has to be extended to multiple classes of object structures by training the corresponding support vector classifiers. In addition, further investigation can be done on recently introduced support vector classifiers [27].

Finally, more informative constraints can be utilized in the particle filter by considering plausible object configurations. The presented algorithm is currently being integrated into a part based object classification system.

Acknowledgement

Thanks to Jacek Czyz for providing further insights into his particle filter implementation and to Luciano Spinello for the fruitful discussions on support vector classifiers.

References

[1] F. Blais, "Review of 20 years of range sensor development," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 231–243, January 2004.

[2] S. May, K. Pervoelz, and H. Surmann, *Vision Systems - Applications*. I-Tech, 2007, ch. 3D Cameras: 3D Computer Vision of wide Scope, pp. 181–202.

[3] T. Kahlmann, "Range imaging metrology: Investigation, calibration and development," Ph.D. dissertation, Eidgenössische Technische Hochschule Zürich, ETHZ, Diss ETH No 17392, 2007.

[4] S. A. Gudmundsson, H. Aanaes, and R. Larsen, "Environmental effects on measurement uncertainties of time-of-flight cameras," in *International Symposium on Signals, Circuits and Systems (ISSCS 2007)*, vol. 1, 2007, pp. 1–4.

[5] R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artificial Intelligence*, vol. 17, pp. 285–348, 1981.

[6] L. Stark and K. W. Bowyer, *Generic Object Recognition using Form and Function*, ser. Series in Machine Perception and Artificial Intelligence, H. W. P. Bunke, Ed. World Scientific, 1996.

[7] M. A. Aycinena, "Probabilistic geometric grammars for object recognition," Master's thesis, Massachusetts Institute of Technology - Department of Electrical Engineering and Computer Science, 2005.

[8] P. J. Besl, *Surfaces In Range Image Understanding*. Springer-Verlag Inc., New York, 1988.

[9] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, 1993.

[10] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter - Particle Filters for Tracking Applications*. Artech House, 2004.

[11] R. B. Fisher, *From Surfaces to Objects - Computer Vision and Three Dimensional Scene Analysis*. John Wiley & Sons Ltd., Chichester, Great Britain, 1989, http://homepages.inf.ed.ac.uk/rbf/BOOKS/FSTO/ (14.9.2007).

[12] H. Rom and G. Medioni, "Part decomposition and description of 3D shapes," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, vol. 1, 1994, pp. 629–632.

[13] W. H. Field, D. L. Borges, and R. B. Fisher, "Class-based recognition of 3D objects represented by volumetric primitives," *Image and Vision Computing*, vol. 15, no. 8, pp. 655–664, August 1997.

[14] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert, "Natural terrain classification using 3-d ladar data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 5, 2004, pp. 5117–5122.

[15] J. Czyz, B. Ristic, and B. Macq, "A color-based particle filter for joint detection and tracking of multiple objects," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 2, 2005, pp. 217–220.

[16] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proceedings of the European Conference Computer Vision (ECCV)*, ser. LNCS 2350, A. H. et al., Ed. Springer-Verlag, 2002.

[17] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive object tracking based on an effective appearance filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1661–1667, 2007.

[18] MESA Imaging AG, Switzerland, http://www.swissranger.ch/ (13.9.2007).

[19] S. May, B. Werner, H. Surmann, and K. Pervölz, "3D time-of-flight cameras for mobile robotics," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[20] C.-F. Westin, "A tensor framework for multidimensional signal processing," Ph.D. dissertation, Department of Electrical Engineering - Linköping University, Sweden, 1994.

[21] G. Medioni, M.-S. Lee, and C.-K. Tang, *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

[22] C.-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz, "Geometrical diffusion measures for MRI from tensor basis analysis," in *Proceedings of the 5th Annual Meeting of the International Society for Magnetic Resonance Medicine (ISMRM)*, 1997, p. 1742.

[23] S. Gächter, "Incremental object part detection with a range camera," Autonomous Systems Lab, Swiss Federal Institute of Technology, Zurich (ETHZ), Switzerland, Tech. Rep. ETHZ-ASL-2006-12, 2006.

[24] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Department of Computer Science - National Taiwan University, Tech. Rep. July 18, 2007, 2007.

[25] M. Isard and A. Blake, "CONDENSATION - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[26] T. Akenine-Möller and E. Haines, *Real-Time Rendering*, second edition ed. A K Peters, 2002.

[27] C. Shen, H. Li, and M. J. Brooks, "Classification-based likelihood functions for bayesian tracking," in *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance (AVSS'06)*, 2006, p. 33.

4 Bayesian Space Conceptualisation and Place Classification for Semantic Maps in Mobile Robotics

Authors: S. Vasudevan, R. Siegwart,

Year: 2008, in press

Published in: International Journal of Robotics and Autonomous Systems

Bayesian Space Conceptualization and Place Classification for Semantic Maps in Mobile Robotics

Shrihari Vasudevan *, Roland Siegwart

Autonomous Systems Laboratory, Swiss Federal Institute of Technology Zürich, 8092 Zürich, Switzerland.

Abstract

The future of robots, as our companions is dependent on their ability to understand, interpret and represent the environment in a human compatible manner. Towards this aim, this work attempts to create a hierarchical probabilistic concept-oriented representation of space, based on objects. Specifically, it details efforts taken towards learning and generating concepts and attempts to classify places using the concepts gleaned. Several algorithms, from naive ones using only object category presence to more sophisticated ones using both objects and relationships, are proposed. Both learning and inference use the information encoded in the underlying representation - objects and relative spatial information between them. The approaches are based on learning from exemplars, clustering and the use of Bayesian network classifiers. The approaches are generative. Further, even though they are based on learning from exemplars, they are not ontology specific; i.e. they do not assume the use of any particular ontology. The presented algorithms rely on a robots inherent high-level feature extraction (object recognition, structural element extraction) capability to actually form concept models and infer them. Thus, this report presents methods that could enable a robot to to link sensory information to increasingly abstract concepts (spatial constructs). Such a conceptualization and the representation that results thereof would enable robots to be more cognizant of their surroundings and yet, compatible to us. Experiments on conceptualization and place classification are reported. Thus, the theme of this work is - conceptualization and classification for representation and spatial cognition.

Key words: Conceptualization of Space, Place Classification, Bayesian Inference, Spatial Cognition, Robot Mapping, Semantic Mapping

1. Introduction

Robot mapping is a well researched problem, however, with many very interesting challenges yet to be solved. An excellent and fairly comprehensive survey of robot mapping has been presented in [1]. Robot maps can be generally classified into three categories - metric ([2],[3]), topological ([4], [5]) and hybrid ([6], [7]). The one similarity between these representations is that all of them are navigation-oriented. Thus, while these maps are certainly useful in getting robots to move around, they fail to encode much of the spatial semantics in the environment. This results in robots having a very modest level of spatial awareness (an understanding of space). The focus of this work is to address these deficiencies. Further, a robot may use such representations to perform spatial cognition to different extents. While (metric) localization and topological place recognition (is this *my* office ?) have been well explored ([2], [5] & [8]) in the research community, place classification (is this *an* office ?) is a more general problem and warrants the formation of a conceptual model of the place. The work reported here addresses this issue in the overall context of improving the semantic content of state-of-the-art robot representations.

Typically, humans perceive space in terms of objects, states and descriptions, relationships etc. This is both intuitive and is also validated through user studies that were conducted in [9]. Thus, a *cognitive* spatial representation, for a mobile robot, could be expected to encode similar information. The work reported in [10] attempted to create such a representation by encoding typical household objects and doors within a hierarchical probabilistic framework. It used a SIFT [11] based object recognition system and a door detection system based on lines extracted from range scans. It also proposed a first conceptualization of different places, based on the objects that were observed. Spatial cognition was demonstrated in two ways - place classification using

* Corresponding author.

Email address: shrihari.vasudevan@ieee.org (Shrihari Vasudevan).

the models learnt and place recognition using the probabilistic relative object graph representation (a graph encoding objects and 3D relative spatial information between them). The conceptualization and place-classification that was performed were preliminary steps in the direction. This work attempts to build on that representation by endowing a robot with the capability to form functional concepts of places based on the objects and inter-object relationships that it perceives. For instance, consider a kitchen that is composed of a storage-space, a cooking-space and a dining-space, each of which are in turn composed of several objects pertinent to it. This work enables a robot exploring the kitchen to actually understand (and internally represent) that there is an area to dine, to cook and to store things in the place, and that the place is a kitchen because of this.

2. Related Work

Many works either inspire or are closely related to the work presented here. In the Artificial Intelligence (AI) community, the problem of *generalization* has been well addressed. The work [12] provides a good overview of different generalization strategies that exist and how they relate to each other. The approach presented in this work can be likened to a data driven approach which requires a set of positive / negative exemplars (or a "teacher") to learn from. The problem of *conceptual clustering* is another closely related and well established research area. Perhaps, the best known example of this, is the COBWEB system [13]. This system attempted to perform unsupervised incremental probabilistic conceptual clustering. The problem, approach and the methodology of generating and using probabilities is different from that presented here. Among more recent works, the aspects dealt with in this report, bear similarities with [14], it presented a generative probabilistic model for classification and clustering of relational data. The model is based on previous work by the authors on Probabilistic Relational Models. The model incorporates a large set of dependencies between the latent variables representing the entities of the data; it used an approximate Expectation-Maximization algorithm to learn the parameters of the underlying model and inference was based on Belief Propagation. Another closely related work, to that presented here, is reported in [15]. It provides a Bayesian approach to learning concepts from a few positive exemplars. The specific example demonstrated is that of learning axis-parallel rectangles in multi-dimensional space.

Recent works in robotics that are relevant to the work presented here include [16] and [17]. The former used an AI based reasoning engine that specified rules for each concept based on an ontology. The latter used the objects to differentiate between similar structured rooms - this was done by integrating the object cues within an AdaBoost framework. The state-of-the-art in robot place classification typically relies on object occurrence cues, used in a logic or rule based framework, possibly with a predefined ontology. A recent contribution that works along these lines is detailed in [18]. The objective of the work reported here, is to formulate a principled Bayesian approach in order to incorporate semantic concepts in robot spatial representations and enable robots to be able to reason about their surroundings. The scenario envisioned is that of a robot being taught different spatial concepts by its human user.

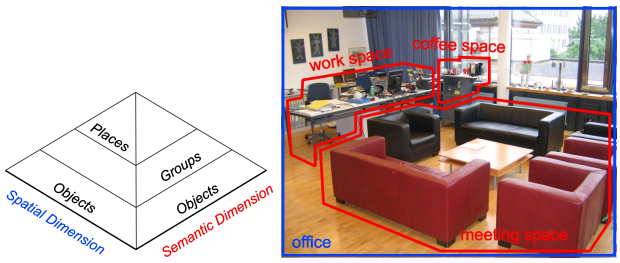
A concept that provides for the basis of the approach presented here is that of the Bayesian network classifiers-in particular, the Naive Bayesian Classifier (NBC). It is well known that NBC's (generative classifiers) although being unarguably simplistic models that make strong assumptions, are able to successfully compete with any of the other state-of-the-art (discriminative) classifiers [19]. The work [20] gives a nice overview on the different kinds of Bayesian network classifiers that exist and also elicits on ways to learn them. The approach presented in this report also draws on the vast amount of work done in the area of *clustering*, a good survey of which is presented in [21]. Additionally, this work attempts to be fully probabilistic and is grounded on a Bayesian Programming methodology, as described in [22].

The approaches presented in this paper bear a strong resemblance to the state-of-the-art techniques applied in the Computer Vision - Image / Object Class Recognition community. The purely object based models that will be presented in this paper bear a resemblance to the "bag-of-words" approach where no explicit geometric information is used to classify objects. Relevant examples of this technique include [23] and [24]. Both works categorize natural scenes based on a set of "code-words" (basically, image patches that characterize local information). The idea was to model different scene categories in terms of these code-words. While the former uses a supervised local-labeling scheme during training, the latter is unsupervised in that it simply extracts local features from each image and looks for a similarity in feature space. The approach reported later in this work (and the proposed future extensions) incorporates inter-object relationships in the reasoning process and thus, bears a strong resemblance to the "constellation" or "part-based-approaches" used in object classification. Perhaps, the most relevant examples amongst this class of approaches are [25], [26] and [27]. Fergus et al. [26] model an object as a number of parts. Each of these parts is characterized by its appearance, scale etc. Further, the shape of the object is represented as a joint Gaussian density of the locations of the parts. All of this information is combined in a Generative model to classify objects. Bouchard et al. [27] propose a method to group image features into local feature classes; these are in turn inferred as being parts of an object; the object parts are then used in an inference process to identify the object class. Sudderth et al. [25] also propose a hierarchical probabilistic model to classify objects in terms of its parts. These parts describe the expected appearance and position, in an object centric reference frame, of features detected by a low level interest point detector like SIFT. Each object category has its own

distribution over these parts.

The contribution of this work is the formulation of a sound Bayesian methodology to enable a robot to conceptualize and classify its environment as it explores it. The inference process is generative, uses information encoded in the representation (objects and relative spatial information between them), is not ontology specific and only relies on a robots high-level feature extraction (object recognition and structural element extraction) capability. Such an inference would enable a robot to establish a consistent link from sensory information it obtains to increasingly abstract spatial concepts. The representation that is formed as a result of the conceptualization, encodes a greater level of semantic information (concepts) than before and enables a robot to be more spatially aware of its surroundings. Also, the representation would be totally compatible with humans (demonstrated in [9]).

3. Approach



(a) (b)

Fig. 1. **(a)** General approach - A robot uses the sensory information it perceives to identify high level features such as objects, doors etc. These objects are grouped into abstractions along two dimensions - spatial and semantic. Along the semantic dimension, objects are clustered into groups so as to capture the spatial semantics. Along the spatial dimension, places are formed as a collection of groups of objects. Spatial abstractions are primarily perceptual formations (occurrence of walls, doors etc.) whereas semantic or functional abstractions are primarily conceptual formations (similarity of purpose / functionality ; spatial arrangement). The representation is a single hierarchy composed of sensory information being mapped to increasingly abstract concepts. **(b)** An example scenario - The figure depicts a typical office setting. The proposed approach would enable a robot to recognize various objects, cluster the respective objects into meaningful semantic entities such as a meeting-space and a work-space and even understand that the place is an office because of the presence of a place to work and one to conduct meetings.

Figure 1 illustrates the overall approach that is being pursued. In [10], a key idea was to enhance robots spatial representation by changing the feature set from the now common lines, corners etc. to higher level features such as objects and doors. It established the link between the robots sensors, the objects and the places. This work attempts to build on that idea by asking the question - given a set of objects, how can a robot be made to gain a deeper understanding of its surroundings ? It attempts to form groups in accordance with the hierarchy shown. The objective is a greater incorporation and usage of spatial semantics, the resulting outcome is a concept oriented (thus, more semantic) representation of space. In this report specifically, two questions are addressed - (1) How can a robot build a conceptual model of the functional entities that constitute a place ? and (2) How can a robot understand that it is in a particular type of place. The former refers to the problem of conceptualization and the latter, the problem of place classification.

In accordance with figure 1(a), objects are incrementally grouped into clusters which are conceptualized as functional groupings (concepts or groups in this report). These groups provide for meaningful semantics that the robot can glean as it explores a place. The robot can then use the groups to infer about or classify the place. Inference is based on the Naive Bayes Classifier (NBC). The key improvement lies in the creation of an intermediate level of semantic understanding, which certainly increases semantic content in the representation but may also improve understanding at higher levels of abstraction.

Figure 2 depicts the process that occurs during the learning and cognition stages. This paper is about the modules that have been encircled in the figures. It must be emphasized that perception is not the thrust of this work even if it is addressed in the context of real experiments; rather it is proposition of algorithms that can enable the creation of a semantic map for a mobile robot. Note that all of the information that is used in the cognition process is directly obtained from the representation (map) that a robot would form when it explores its surroundings - in this way the presented work remains grounded and builds on a robot mapping basis. It assumes that a robot can detect objects (currently demonstrated using a SIFT based object recognition system applied on real sensory data in [10]) and that a human user would show the robot around in a "home tour" scenario annotating semantic entities suited to them. The robot would use its feature extraction (object detection) capabilities in conjunction with the semantic annotations provided to learn models of these concepts, based on the approach reported here.

3.1. *Overview of attempted approaches*

Figure 3 gives a quick overview of this report, depicting the various approaches that have been attempted in order to conceptualize and cognize space. The first preliminary steps (M1 in figure 3) towards object based conceptualization were made in the work [10]. The classification was based on a very simplistic Naive Bayesian Classifier (NBC) [28] that did not learn from negative exemplars. The likelihood formulation in the conceptualization was not useful for handling multiple occurrences of objects. It also did not use any explicit relationship between the number of occurrences of an object and the concept. Classification was done only on the basis of the evidence that was present and did not consider that which was absent, the latter is very signif-

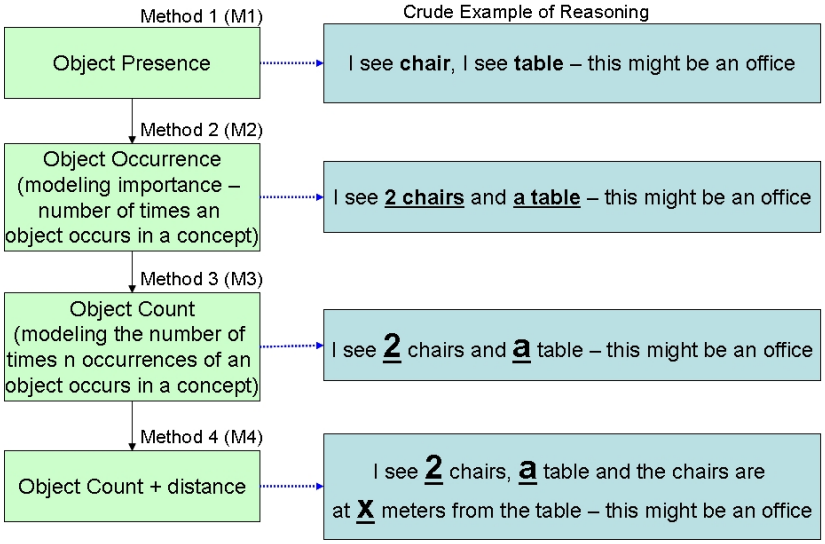


Fig. 3. An overview of different approaches to conceptualization and place classification. The first three approaches use objects only whereas the last one incorporates relationships as well. This report briefly describes M1 and M2; it highlights their issues. Further, it provides an elaborate evaluation of M3 and M4 to estimate their prediction accuracies; it also compares them to understand the precise effect of incorporating relationships.

icant information. Further, [10] represented spatial semantics through only the *presence* of objects. In order to get around these problems, the work [29] modeled the *importance* of different objects towards the formation of different concepts (M2 in figure 3). While this work could overcome the previously mentioned problems, it had the drawback of having a low classification rate, even if the correctness of the classified cases was quite high. Also, conceptually, the *importance* model basically gave each occurrence of every object the same contribution towards a particular concept. This probably explained the low classification rate as for the same evidence, multiple concepts could be inferred and no clear outcome was observed. In order to get over this problem and propose a more explicit model for inferring concepts, [30] was performed. In this work, the learnt knowledge explicitly modeled the contribution of a certain *number of occurrences* of each object towards a particular concept (M3 in figure 3). This work produced very encouraging results in terms of both the classification rate as well as the correctness of the classified cases. In an attempt to build on the representation proposed in [10] and the promising results obtained in [30], the method M4 (figure 3) was proposed in order to incorporate characteristic relationships in the process of identifying concepts. The following sections provide detailed descriptions, experimental evaluations and comparison of the approaches denoted M3 and M4.

3.2. *On the clustering methodology*

3.2.1. *Approach*

The conceptualization process to actually infer the concepts works on clusters of objects. Different clustering approaches inspired by [21] were attempted. Most were based on nearest neighbor approach as distance between objects is a reasonable clustering metric. The objective, however, was to also make use of the semantic information captured in the concept models learnt by the robot. Thus, a nearest neighbor approach in conjunction with a Maximum-a-posteriori (MAP) estimate of the best case concept for the incoming (perceived) object, was the basis of the clustering method that has finally been used in this work. The former used the distance to the center of the cluster as the metric whereas the latter was the concept that had the maximum posterior belief given the occurrence of the single object. It is computed by learning, from the training data, the likelihood of observing the object, given the occurrence of the concept. This is the information encoded in the learning process of the algorithm M2 presented in [29] - thus, the clustering process can be understood as the application of M2 for each object. The behavior of the algorithm can be briefly summarized in three steps in the same order of precedence - (1) choose the nearest cluster that has the same concept as the best case concept suggested for the incoming object, (2) choose the nearest cluster that is conceptually dissimilar but "acceptably likely" with respect to the best case concept and (3) create a new cluster with the incoming object of type suggested by the best case concept. In the absence of any concept models (i.e. no prior training), the clustering process would boil down to a nearest neighbor approach.

3.2.2. *Experiments & Discussion*

Table 1 summarizes the evaluation of the clustering process. *Correct* cases correspond to objects which belonged to the respective clusters, in comparison with the training data. A significant number of clusters were either fused or broken with others. In most cases, this resulted in for in-

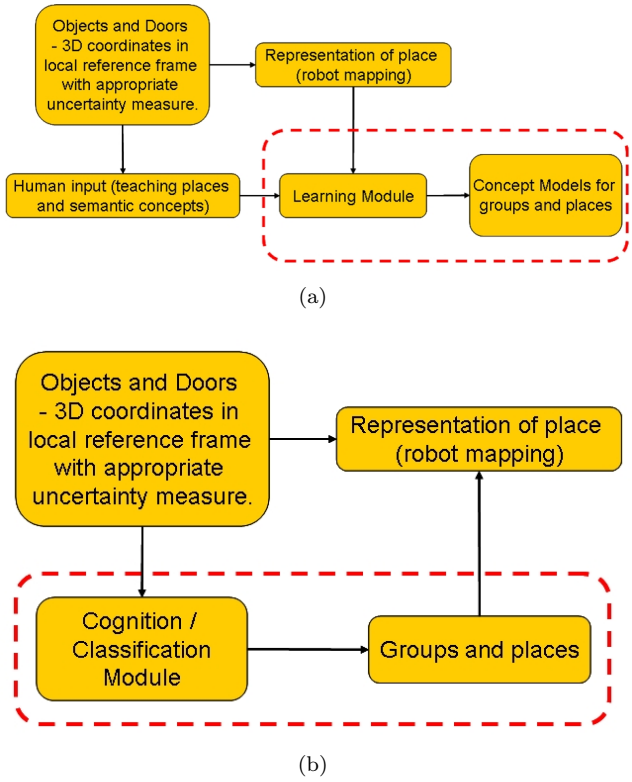


Fig. 2. **(a)** Information flow in the training / learning process: The robot is assumed to have a high-level feature (objects, doors etc.) extraction capability. As it explores it surroundings, it creates a map of the environment. During a "home-tour" like scenario, the robot would be shown different instances of various concepts - both spatial and semantic. The robot with its feature extraction capability, together with the algorithms presented in this report will learn concept models for the exemplars it has been taught. **(b)** Information flow in the testing / cognition process: As before, an exploring robot with high level feature extraction capability will attempt to create an object based representation of space. The objects and relationships that are mapped are also interpreted as instances of various semantic concepts (groups) and places. These higher level concepts and places form the higher levels of the representation that would realize the vision depicted in figure 1

Table 1
Evaluation of the clustering algorithm (with concept models)

Outcome	Cases	Percentage (%)
Singleton	11	1.1100
Fused or Broken	296	29.8688
Correct	684	69.0212

stance, the fusion of two adjacent work-spaces or the inclusion of one or more objects of one cluster in another one. Cases did occur, where objects characteristic of one concept were clustered with those of another. A clear conceptualization could be unlikely in these cases. A few objects were separated from the rest and formed clusters by themselves - these were regarded as being inaccurate with respect to the training input (where only large objects such as cupboards were treated as singleton clusters). The number of such cases however, was quite low.

The advantage of a semantic clustering process over a

Input: Concept models for each concept (prior; likelihood of observing different objects given concept occurrence) and the current clusters and/or objects being perceived.

For each new object observed -
If there exists at least one cluster

(i) Get all nearest neighbor clusters (metric = distance to center of cluster) within a predefined range. Arrange this in ascending order of distance (decreasing order of nearness). Also have their concepts in a separate list.
(ii) Get the best case concept - the concept whose occurrence is most indicated by the occurrence of the object. It is a maximum a posteriori (MAP) estimate of the concept.
(iii) Cases - do the first applicable case
  (a) If there is no best case concept (case 0)
    (i) If there is at least one nearest neighbor - choose the nearest neighbor.
    (ii) Else add object to a new cluster
  (b) Choose the nearest cluster from list that has the same concept as the best case concept suggested for the incoming object. (case 1).
  (c) Choose the nearest cluster from list, that is conceptually dissimilar but "acceptably likely" (defined by a threshold that is experimentally found) with respect to the best case concept (case 2).
  (d) Create a new cluster with the incoming object of type suggested by the best case concept. (case 3)

Else create first cluster.

Fig. 4. Informal description of the clustering process. The algorithm assumes the presence of an object recognition system and a set of concept models for the known concepts. The general behavior of the algorithm can be summarized in terms of steps b, c and d. In the absence of any prior training, the algorithm would function as a nearest neighbor clustering algorithm.

Table 2
Evaluation of a nearest neighbor clustering algorithm (no concept model)

Outcome	Cases	Percentage (%)
Singleton	5	0.5045
Fused or Broken	425	42.8860
Correct	561	56.6095

simple nearest neighbor like process is shown in figures 14, 15 and tables 1, 2 respectively. In the former, a human user's perception of different clusters is sought - thus, a cupboard would be a separate entity with respect to for instance a cooking-range. The nearest neighbor algorithm does not consider any object level semantics and simply uses the distance to the center of the cluster as a metric - thus even though clusters are formed, they may not be "semantically meaningful". Note that bad clusters will result in either an inability to conceptualize or a completely incorrect conceptualization. In the tables above, clearly, using the nearest neighbor algorithm alone would result in an ≈12% increase in objects that have been fused or broken with other clusters (with a corresponding decrease in correct outcomes) with respect to the training data. Although not explicitly depicted in the tables, the conceptualization and place clas-

sification accuracies dramatically drop to around the ≈50% range, in the case where the nearest neighbor algorithm is used alone.

3.3. *Method 3 (M3): The Object Count model*

3.3.1. *Approach*

$$P(\,c\,,\,X_0\,,\,X_1\,,\,\ldots\,,\,X_n\,) = P(\,c\,) \,*\, \prod_{i=0}^{n} P(X_i \,|\, c) \tag{1}$$

Equation 1 shows the joint probability distribution (JPD) of the model. Given a set of objects (o_i), the equation computes the belief in a concept given number of occurrences (m_i) respectively, of each of the objects. Every X_i in the equation denotes an $o_i = m_i$ for the corresponding object; c denotes the concept that is to be inferred. The inference is principled on the Bayes rule that interprets this in terms of the prior belief in the concept and the likelihoods of observing the specific number of occurrences of the respective objects, given the concept. Given that a NBC is the underlying formalism, the object occurrences are assumed to be independent of each other, given the concept. The same method is also used to infer about the place given the occurrence of one or more concepts.

The concept model that is used for the inference, encodes the likelihood of the occurrence of a specific number (over a range) of a certain object towards the formation of a particular concept. It is worth noting that encoding and using the number of occurrences of various objects rather than just their individual occurrences is a more informative method of distinguishing between concepts. For instance, chairs and tables are common to both a work-space and a dining-space, however, the number of occurrences of each of them is one distinguishing factor. For a set of concepts c_i, a set of objects o_i and a range of possible number of occurrences m_i, the training process uses a collection of positive and negative concept exemplars to compute the likelihoods as shown in equation 2.

$$P(o_i \,=\, m_i \,|\, c_i) = \frac{N_{o_i = m_i} \,+\, \delta}{N_{exemplars} \,+\, (2 * \delta)} \tag{2}$$

where the numerator encodes the number of occurrences of the particular case $o_i = m_i$, for every object over a range of occurrences, and the denominator encodes the number of positive or negative exemplars of the concept. The terms δ and $2{*}\delta$ ensure that an event that has not been encountered during prior training, is only something that the robot has no prior information about (belief = 0.50) and not something that may never occur. The value of δ decides the reliance on the training data. In the experiments reported in this work, δ takes a very low value of 0.001 so as to reflect the training data accurately. The likelihoods were also limited to taking values between upper and lower bounds in order to avoid 'un-interesting' inferences that could be produced in the limiting cases.

The concept model in its present form would generalize on exactly the set of exemplars presented to it, assuming that the exemplars were themselves void of any uncertainty. This aspect is very significant as, for instance, if in training, four chairs were always observed in a dining-space, the occurrence of three chairs (a very plausible scenario) would probably render the algorithm being unable to comprehend the group. In such a scenario, the algorithm should infer the possible existence of a dining-space, albeit with a greater uncertainty (lower belief). Thus, the algorithm should be able to generalize in a manner such that it is able to handle at least conceptually "adjacent" cases to what it has observed before. Also, given that a user is expected to teach the robot in an on-line learning scenario (and not use some predefined data-base of models), it would only be appropriate to consider the training input as being uncertain. To this effect, a Gaussian uncertainty was incorporated in the training input - so that every training input affects not only $P(o_i = m_i|c)$ but also its neighbors. The choice of the Gaussian noise to be used would depend on the local circumstances and the aspects that need to be modeled. In the experiments presented here, $N(0.0, 0.4472)$ was used in order to consider only the number of occurrences $o_i = m_i$, $o_i = m_i - 1$ and $o_i = m_i + 1$ respectively (i.e. only the immediate neighbors).

A Bayesian program is a systematic formulation for the creation and usage of Bayesian models such as the one used in this work. Elaborate details on the concept, its structure and its semantics are available in [22]. The Bayesian program used to do the learning and inference process is summarized as shown in figure 5. The complete probabilistic model for the system, including the parameters, likelihoods and the question to be answered, are depicted in it.

3.3.2. *Experiments & Discussion*

Experiments were conducted on a dataset (more details of which are given in the appendix) that included physically measured object and coordinate information from 11 offices and 8 kitchens. The office data was represented in terms of three concepts (apart from some free-standing objects). These were work-space, storage-space and meeting-space. The kitchen data was described in terms of ten concepts, namely cooking-space, garbage-space, dining-space, bottle-group, glass-group, box-group, mug-group, bag-group, poster-group and book-group. Concepts used in this work represent the manner in which the places were understood by the authors; they are similar to those observed in [9]. The approach however is not ontology-specific. Developing a standardized ontology that could perhaps enable high-level communication between robots is beyond the scope of this work.

Two instances each, of office and kitchen data were used only for testing and the others for both training and testing. Training was performed to learn the unknown parameters shown in figure 5, for each concept. Each concept was trained with its set of positive exemplars and against all

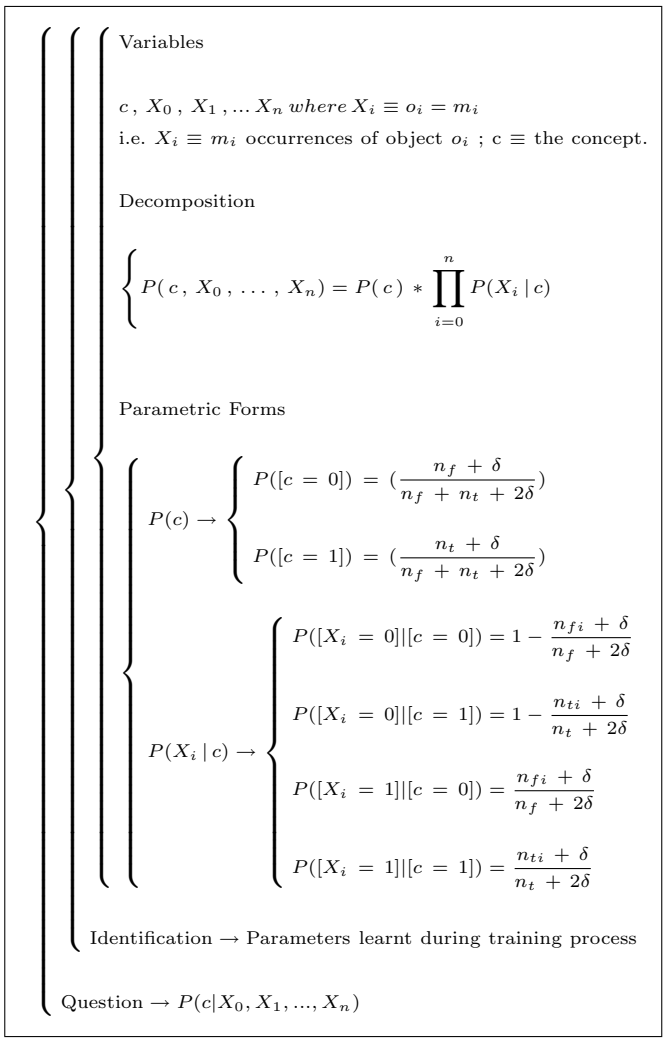


Fig. 5. The Bayesian program that summarizes the learning and inference processes. '0' denotes a 'false' and '1' denotes a 'true'. $\delta = 0.001$ to accurately reflect on the training data, it ensures that a previously unseen event is an unknown event (belief = 0.5) and not one that never occurs. n_{fi} and n_{ti} are the number of occurrences of the case $o_i = m_i$ in negative and positive exemplars respectively. n_f and n_t are the number of negative and positive exemplars respectively. The same process can be applied to infer about places given the concepts observed.

other exemplars as negative ones. Testing and evaluation involved the comparison of each of the 991 objects (total number in 19 places) with the corresponding objects in the training input. Conceptualization resulted in four outcomes. An object may have been conceptualized correctly, i.e. it belongs to the correct conceptual group with respect to the training data; it may belong to a group that has not been classified (due to insufficient evidence or multiple competing hypotheses inhibiting a clear inference); it could be a free-standing object in training, that has been assigned a label; finally, the object may belong to a group that has been incorrectly classified. Figures 13 and 14 respectively depict the outcome of clustering, conceptualization and place classification of an office and a kitchen.

Detailed description of this model, the experimental results and the effect of incorporation of the Gaussian uncertainty in the model are provided in [30]. The result of the tests conducted with the Gaussian uncertainty incorporated is given in table 3 as a benchmark for comparison with subsequent approaches presented in this report. The results were very encouraging. Even in the context of place classification, it produced perfect results, identifying all 11 offices and 8 kitchens correctly.

Table 3
Evaluation of the conceptualization algorithm using model M3 (object count)

Outcome	Cases	% (of classified)	% (of total)
Incorrect	175	18.5381	17.6589
Not classified	47	-.	4.7427
Free Object	9	0.9534	0.9082
Correct	760	80.5085	76.6902

The following conclusions were drawn on method M3 over the previous approaches (M1 and M2)

(i) Superior results were obtained for both conceptualization as well as place classification, when compared with previous approaches - M1 and M2.
(ii) The method addressed all conceptual deficiencies highlighted in the previous approaches.
(iii) Object count was proved to be a better classification feature than object significance/importance.
(iv) The Gaussian uncertainty incorporated in the training input helps the approach achieve superior generalization capability in that it can better handle conceptually adjacent cases.

In an attempt to build on these results and use more spatial semantics towards conceptualization and classification, model M4 was proposed. It was primarily meant to extend M3 by incorporating spatial relationships into the framework. Detailed information on this approach follows.

3.4. *Method 4 (M4): The Object Count + Relationship model*

3.4.1. *Approach*

$$P(\,c\,,\ X_{1\ldots n1}\,,\ R_{1\ldots n2}) = P(\,c\,) * \prod_{i=1}^{n1} P(X_i \mid c) * \prod_{j=1}^{n2} P(R_j \mid c) \quad (3)$$

Equation 3 shows the joint probability distribution (JPD) of the model used in this work. Given a set of objects (o_i), the equation computes the belief in a concept given number of occurrences (m_i) of each of the objects and the relationships R_j, observed between the object instances. Every X_i in the equation denotes an $o_i = m_i$ for the corresponding object; c denotes the concept that is to be inferred and a R_j denotes a relationship between two

objects. In the current model, only the distances between objects in 3D space are used towards inferring concepts. These spatial relationships are in-line with the underlying choice of representation and are available to the robot as it constructs the map of the space in accordance with [10]. Given that a NBC is the underlying model, the numbers of object occurrences and the individual relationships are assumed to be independent of each other, given the concept. The inference is principled on the Bayes rule that interprets it in terms of the prior belief in the concept, the likelihoods of observing the specific number of occurrences of the various objects given the concept and the likelihood of observing the specific relationships, again, given the concept. The same method is also used to infer about the place given the occurrence of one or more concepts.

The concept model for objects is exactly the same as that presented earlier in approach M3 (incorporating the Gaussian uncertainty in training). The concept model for relationships is learnt as a Gaussian mixture model (GMM) from the relationship values. While object occurrence models are based on discrete variables, the relationship models are formed from continuous random variables. The relationship models are learnt by taking all previous occurrences of the relationship and using the Expectation Maximization (EM) algorithm [31] to learn the GMM parameters for that relationship. Of the many relationships that actually exist, those that are to be modeled are selected using the number of occurrences as a measure of their significance. The reasoning behind the use of the GMM to model spatial relationships between objects is that - the method attempts to capture characteristic relationships that are typically maintained between objects (for instance a book is always placed above a table). Each of these relationships may possibly occur in a few frequently occurring values. The objective of this modeling is to identify which spatial relationships are distinctive enough and which values of these relationships occur most often.

It must be noted that the relationship data that are used for the learning step include its value and its covariance matrix (in accordance with the belief representation in [10]). The EM algorithm is adapted to include the uncertainty in the input data as done in [32]. It uses a KL-divergence approach to deriving the update equations, the same may also be derived using the maximizing the log-likelihood approach as both processes are equivalent. The final update equations (similar to the standard ones) are given by equations 4 through 7 using the convention followed by [31]. N data items (x_n) are assumed to be modeled using K Gaussian mixture components of the model, defined by model parameters π_k (prior), μ_k (mean) and Σ_k (covariance matrix). $\gamma(z_{nk})$ denotes the responsibilities of the k^{th} mixture component explaining the n^{th} data item. Finally, C_n denotes the uncertainty of each of the data items to be clustered. The parameters of the GMM are obtained through an iterative expectation-maximization process that ends when the model parameters converge.

$$\gamma(z_{nk}) = \frac{\pi_k \,.\, p(x_n|z_{nk}=1;\theta)}{\sum_{j=1}^{K} \pi_j \,.\, p(x_n|z_{nj}=1;\theta)} \tag{4}$$

where

$$p(x_n|z_{nk}=1;\theta) = 2\pi^{-d/2} \,.\, \Sigma_k^{-1} \,. \\ exp\{-0.5 * (x_n-\mu_k)^T.\Sigma_k^{-1}.(x_n-\mu_k) \\ -0.5 * Tr(\Sigma_k^{-1}.C_n)\}$$

$$\mu_k{}^{new} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})\, x_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{5}$$

$$\Sigma_k{}^{new} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})\, [(x_n-\mu_k{}^{new})'.(x_n-\mu_k{}^{new}) + C_n]}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{6}$$

$$\pi_k{}^{new} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N} \tag{7}$$

Visual inspection of some of the relationship data suggested that relationships that had little diversity (typically also the infrequent ones) could be modeled using a single Gaussian mixture whereas some of the more frequently occurring ones having diverse values, with two or more Gaussian mixture components. The exact number of Gaussian mixture components required to model the relationship is dependent on the dataset. Thus, the Bayesian information criterion (BIC) [33] was used to decide the number of mixture components used to model a particular relationship.

A relationship is selected for modeling based on its significance (directly decided by the number of occurrences of it). Thereafter it is subject to an EM based GMM modeling procedure. Finally, the relationship is also subjected to a post-modeling test which decides if it is actually used or not. In order to ensure that the modeling is appropriate, a parametric bootstrap approach was adopted to check the modeling. Samples were generated using the GMM parameters obtained from the EM procedure. These samples together with the original data are together again subject to the EM based GMM modeling and the resulting GMM is compared with that obtained previously. A students t-test is performed to compare the two GMM's. Only if the parameters are equal at a certain level of significance, is the modeling taken to be appropriate and the relationship model used thereafter for inference. The null hypothesis in this case was that the GMM models were identical. In this report, a 1% level of significance is used in a two tailed testing of the GMM models. Cases where the null hypothesis is rejected at the given level of significance result in the correponding relationship being discarded from the set of relationship models that are used thereafter for the inference process.

The Bayesian Program describing the complete learning and inference processes using the presented model is shown in figure 6.

3.4.2. *Overview of Experiments*

A first set of experiments were conducted in exactly the same manner as described in section 3.3.2. The objective

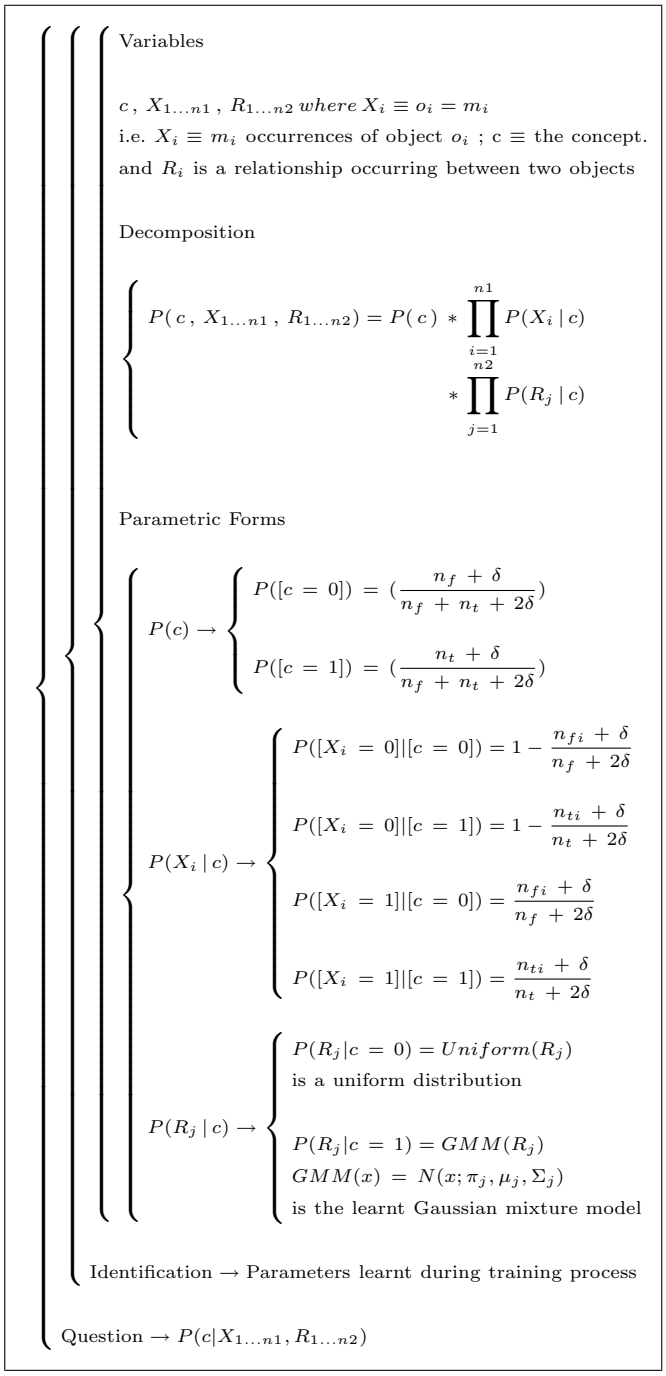


Fig. 6. The Bayesian program that summarizes the learning and inference processes. '0' denotes a 'false' and '1' denotes a 'true'. $\delta = 0.001$ to accurately reflect on the training data, it ensures that a previously unseen event is an unknown event (belief = 0.5) and not one that never occurs. n_{fi} and n_{ti} are the number of occurrences of the case $o_i = m_i$ in negative and positive exemplars respectively. n_f and n_t are the number of negative and positive exemplars respectively. The same process can be applied to infer about places given the concepts observed. π, μ and Σ are respectively the prior, mean and covariance of the GMM representation of the j^{th} relationship R_j.

was to compare the two models through similar tests in order to understand the effect of incorporating relationships in the framework. Thus, the clustering model is kept identical to that used before. The evaluation of the clustering was presented in table 1 and that of the conceptualization algorithm follows. Figures 13 and 14 respectively depict the conceptualization and place classification outcomes for an office and a kitchen respectively.

3.4.3. *Evaluating the conceptualization algorithm*

The experiments attempt to understand two aspects of the approach - (1) the effect of incorporating spatial relationships (specifically, the distance) on the conceptualization outcome (2) the effect of varying the minimum required number of samples (hereafter denoted as NOCC) for a relationship to be considered significant enough to be included in the model. As a benchmark, table 3 depicts the outcome of the conceptualization when only the objects are used - the object count model (with Gaussian improvement) detailed in [30].

The approach was evaluated for different cases of NOCC. The results are tabulated in the tables 4 through 7. Place classification (of the 19 places) was also compared between these models and these results are quantified in table 8.

Table 4
Evaluation of the conceptualization algorithm
(model A : NOCC = 5)

Outcome	Cases	% (of classified)	% (of total)
Incorrect	118	16.2311	11.9072
Not classified	264	-	26.6398
Free Object	9	1.2380	0.9082
Correct	600	82.5309	60.5449

Table 5
Evaluation of the conceptualization algorithm
(model B : NOCC = 10)

Outcome	Cases	% (of classified)	% (of total)
Incorrect	122	15.1365	12.3108
Not classified	185	-	18.6680
Free Object	9	1.1166	0.9082
Correct	675	83.7469	68.1130

Table 6
Evaluation of the conceptualization algorithm
(model C : NOCC = 20)

Outcome	Cases	% (of classified)	% (of total)
Incorrect	126	14.4495	12.7144
Not classified	119	-	12.0081
Free Object	9	1.0321	0.9082
Correct	737	84.5183	74.3693

The addition of relationships adds extra metrics for conceptualization/classification to the system. This has the ef-

Table 7
Evaluation of the conceptualization algorithm
(model D : NOCC = 30)

Outcome	Cases	% (of classified)	% (of total)
Incorrect	139	15.4102	14.0262
Not classified	89	-	8.9808
Free Object	9	0.9978	0.9082
Correct	754	83.5920	76.0848

Table 8
Evaluation of the place classification algorithm

Model	Correct cases (of 19)	% accuracy
A	19	100.00
B	19	100.00
C	19	100.00
D	19	100.00

fect of reducing the number of false positives and increasing the number of true-positives (column 3 of the tables) when compared with table 3. This is a good outcome and occurs because the extra information (relationships) helps to better discern "uncertain" clusters of objects. The number of unclassified cases increases (column 4 of the tables). This can be explained as: uncertain cases which might have qualified with the object models alone, may fail with the relationship models now included. However, the total number of incorrect cases is also significantly lower than in table 3. Inability to conceptualize is a better outcome than incorrect conceptualization.

The higher the value of NOCC, the fewer the number of relationships that are included in the relationship model. But the ones that are included, are frequently observed, probably worth modeling and probably better modeled (more data). These could be distinctive "features" that would be useful for conceptualization and place classification. However, the side effect of this (only for the dataset used here) would be that some concepts such as meeting-space and dining-space would have little distinction (in terms of relationships) as they are quite infrequently observed (few relationships) in comparison to a concept such as work-space. Clearly from tables 4, 5, 6 and 8, higher values of NOCC produce superior results both in terms of false-positives / true-positives (column 3) and the classification rates (column 4). In terms of classification rate, the higher the value of NOCC, the closer the system behaves to the work presented in [30] (table 3). Lower values of NOCC (NOCC = 5) were specifically attempted to observe the behavior of the system by including as many relationships as possible. From table 7, we see a slight decrease in performance. The reason is that fewer relationships are now included in the concept model. As NOCC increases beyond this point, the performance will gradually tend to that when no relationships are used at all - this is the benchmark table 3.

The following conclusions could be drawn at this point -

– Incorporation of relationships had two major outcomes - one major and one minor.
  · Major effect - Incorporation of the distance relationship reduces the number of incorrect outcomes by being unable to classify them. Since a no-classification is better than a wrong-classification, this is a definite improvement.
  · Minor effect - Incorporation of the distance relationship reduces the number of incorrect outcomes by correctly classifying them. One example is shown in 14.
– Using fewer relationships (higher NOCC above) resulted in using the most distinctive and characteristic ones which up-to a certain point improves the performance. Beyond a certain threshold, performance would drop and in the limiting case (no relationships qualified for use) the model would simplify to an M3 one and the performance would also be that of M3.

4. Further Experiments

4.1. *Overview*

Further experiments have been conducted with three aims - (1) to validate the conclusion arrived at earlier, on the effect of incorporating relationships in the inference process (2) to obtain a reasonable estimate of the generalization capability of the algorithms presented earlier (object count only, object count + distance) and (3) to demonstrate the applicability of the algorithm in a real robot setting.

A lot of prior literature exists on appropriate methods to estimate generalization capabilities of learning/classification algorithms. One particularly influential work is [34]. The authors recommend 10-fold stratified cross validation (KFSCV, with K = 10) as the best methodology towards estimating the generalization accuracy of a classifier. Leave-one-out cross validation (LOOCV) has been widely understood to be a low bias - high variance method. In this report, we take this lead and do an 8 fold approximately stratified cross validation (8 fold in order to keep the folds as stratified as possible given the highly unbalanced dataset used in this work). Further, as the folds constitute only one specific combination of kitchens and offices, a leave-k-out cross validation (LKOCV) test was further performed (with K=2; 1 office and 1 kitchen). LKOCV is computationally very expensive and normally the computational process may well "explode" at or beyond K=3. However, in order to be able to comprehensively test for every combination theoretically possible in the dataset (for the chosen value of k), this test was performed. The results have been provided below.

Experiments were conducted on the aforementioned dataset that included physically measured object and coordinate information from 11 offices and 8 kitchens. A detailed description of the dataset that was used for these tests is provided in the appendix. In LKOCV, 2 places (1 office and 1 kitchen) of the total of 19 were used as test cases in each cycle of testing with the remaining 17

places being used for training. Thus, in LKOCV, a total of ${C^{11}}_1 * {C^8}_1 = 88$ training-testing cycles were performed and the results were averaged over all cycles to produce the results shown below. In KFSCV, 8 folds were created with each having one kitchen instance and one or more office instances, in a manner so as to maximize stratification across folds. Each fold was then taken as a test case and the accuracy was measured by a classifier trained on the remaining folds. Testing and evaluation, for the the different kinds of clustering and conceptualization outcomes, was performed as mentioned earlier in section 3.3.2. Two kinds of results are looked at - the general range of outcomes (Range) and the expected value of accuracy (Expected value).

4.2. *K(=8) fold stratified cross validation (KFSCV)*

Table 9
KFSCV results - clustering

	Range (%)		Expected value (%)
Outcome	Mean	Std. Dev.	-
Singleton / Incorrect	1.2147	1.3671	1.2109
Fused / Broken	31.7588	9.9388	33.1988
Correct	67.0265	10.2366	65.5903

Table 9 depicts the evaluation of the clustering outcome. Both from the points-of-view of the mean of the range of outcomes and the expected value, ≈66-67% of the objects were correctly clustered with respectively ≈33-32% of the objects being fused with other clusters / broken from their respective clusters in the ground truth data set. If an object is fused with a cluster of a similar concept as the case in the ground truth - the conceptualization outcome would not be affected much. Objects of one concept type when clustered with those of another - can cause a no-classification or even an incorrect classification.

Table 10
KFSCV M3 accuracy - Expected Outcome

Outcome	Cases	% of Classified	% of Overall
Incorrect	327	33.7810	32.9970
Not classified	23	-	2.3209
Free Object	6	0.6198	0.6054
Correct	635	65.5992	64.0767

Table 11
KFSCV M4 accuracy - Expected Outcome

Outcome	Cases	% of Classified	% of Overall
Incorrect	248	27.7405	25.0252
Not classified	97	-	9.7881
Free Object	9	1.0067	0.9082
Correct	637	71.2528	64.2785

Tables 10 and 11 depict the evaluation of the conceptualization outcomes for the methods M3 and M4, with respect to the expected accuracies. Place classification accuracy = 100 % in every cycle of testing for both M3 and M4. Clearly there is a slight increase (albeit very small) in number of correct outcomes for a significant decrease in the number of incorrect outcomes, with a corresponding increase in numbre of unclassified cases. Inability to classify is a better outcome than incorrect classification. Thus, the major effect of the incorporation of relationships to the model is the conversion of a lot of incorrectly classified cases to unclassified ones - in this sense, the distance acts as a binding or constraining element to an otherwise scattered group of features / nodes in a graph. There is a minor/ marginal effect of conversion of incorrect cases to correct ones.

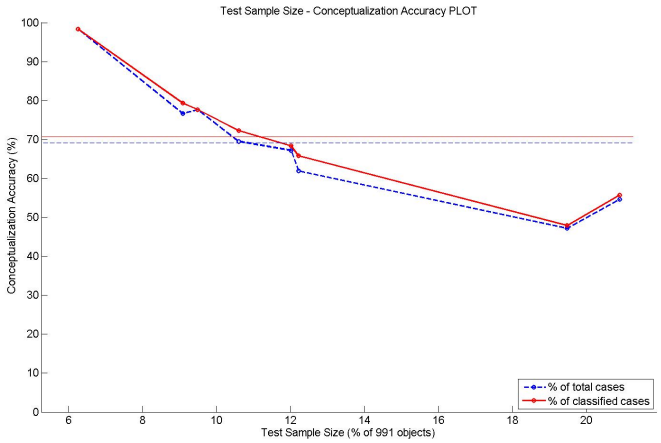


Fig. 7. KFSCV M3 Range of accuracies - Plot of accuracies obtained for various test sets considered (1 for each fold). Note that the general behavior observed is that larger the test set used, the smaller the training set that the algorithm gets to learn from and hence, smaller the accuracy. The plot indicates that up-to about 10-12% of the dataset used for testing could yield above-average results. The mean over all accuracies obtained for all tests is about 69%

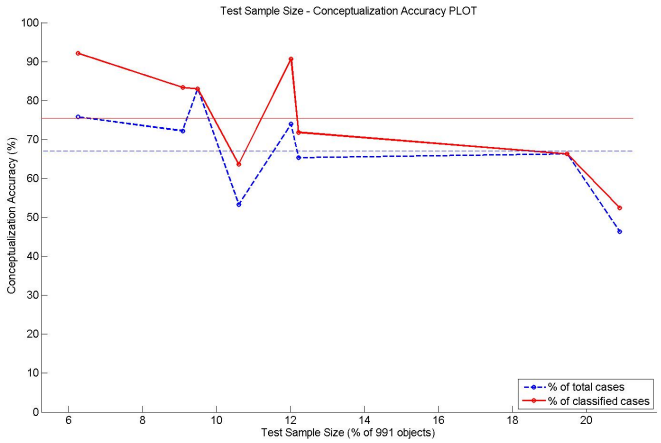


Fig. 8. KFSCV M4 Range of accuracies - Plot of accuracies obtained for various test sets considered (1 for each fold). Note that the general behavior observed is that larger the test set used, the smaller the training set that the algorithm gets to learn from and hence smaller the accuracy. The mean over all accuracies obtained for all tests is about 67%, which is very much comparable to that obtained in 7. Note that relationships are also used for this test. The corresponding tables indicate a significant drop in incorrect outcomes.

Table 12
KFSCV M3 accuracy - Range

	% of Classified		% of Overall	
Outcome	Mean	Std. Dev.	Mean	Std. Dev.
Incorrect	28.5501	15.9473	27.8348	15.6233
Not classified	0.0	0.0	2.2619	1.9689
Free Object	0.7800	2.0144	0.7548	1.9471
Correct	70.6699	15.4331	69.1484	15.7431

Table 13
KFSCV M4 accuracy - Range

	% of Classified		% of Overall	
Outcome	Mean	Std. Dev.	Mean	Std. Dev.
Incorrect	23.5079	14.5333	21.2114	13.1550
Not classified	0.0	0.0	10.8048	7.3639
Free Object	1.0639	2.2281	0.9491	1.9386
Correct	75.4282	14.1211	67.0347	12.0943

Tables 12, 13 and figures 7, 8, depict the evaluation of the conceptualization outcomes for the methods M3 and M4, with respect to the range of outcomes obtained. Place classification accuracy = 100% in every cycle. Here too, the results show that on an average, the range of results obtained using M4 were superior to that obtained from M3. The following additional points were noted:

- The standard deviation / variance of the readings is entirely dependent on the case. Some places or place combinations may be accidentally hard and some others easy. The absence or presence of these places compounded with the problem of using an unbalanced datasets can give rise to a larger than a desired standard deviation. This should not be considered as being too indicative of the quality of the algorithm itself, rather, it may be used to understand that the dataset is clearly unbalanced. Tests have confirmed that not using places 13 and 19 alone (for testing; i.e. the classifier is always trained on them) reduces the standard deviation by 3-4 %.
- Purely from the mean values of the outcome obtained, it is clear that for a similar (or slightly lesser) number of correct outcomes, there is a significant drop in incorrect outcomes with a corresponding increase in average numbers of unclassified cases obtained. This is exactly the same interpretation as obtained before.
- The graphs further show another expected trend - smaller training sets (larger test sets) result in lower estimates of generalization accuracy. Up to about 10% of the dataset, when used for testing, can yield good results.

4.3. *Leave-k-out cross validation (LKOCV)*

KFSCV splits the dataset into K folds and estimates K classification accuracies, one corresponding to each fold.Further, in stratified KFSCV, an attempt was made to keep an approximately similar number of concept instances in each fold - this amounted to having 1 kitchen and 1 (or more, as required and based on size) office. In an attempt to estimate the generalization accuracy over the set of all theoretically possible cases, an elaborate and computationally expensive LKOCV test was performed. The results obtained are summarized below.

Table 14
LKOCV results - clustering

	Range (%)		Expected value (%)
Outcome	Mean	Std. Dev.	-
Singleton / Incorrect	1.0023	1.1735	1.0184
Fused / Broken	34.2462	9.2719	34.7085
Correct	64.7516	9.1477	64.2731

Table 14 depicts the evaluation of the clustering outcome. Both from the points-of-view of the mean of the range of outcomes and the expected value, ≈64-65% of the objects were correctly clustered with respectively ≈35-34% of the objects being fused with other clusters / broken from their respective clusters in the ground truth data set.

Table 15
LKOCV M3 accuracy - Expected Outcome (across 88 tests)

Outcome	Cases	% of Classified	% of Overall
Incorrect	3487	37.1788	36.2361
Not classified	244	-	2.5356
Free Object	57	0.6077	0.5923
Correct	5835	62.2135	60.6360

Table 16
LKOCV M4 accuracy - Expected Outcome (across 88 tests)

Outcome	Cases	% of Classified	% of Overall
Incorrect	2588	30.0441	26.8939
Not classified	1009	-	10.4853
Free Object	72	0.8358	0.7482
Correct	5954	69.1200	61.8726

Tables 15 and 16 depict the evaluation of the conceptualization outcomes for the methods M3 and M4, with respect to the expected accuracies. Place classification accuracy = 100 % in every cycle of testing for both M3 and M4. These results provide the clearest indication yet of the improvement obtained due to incorporating relationships. A 1.2% increase in correct outcomes is obtained for a simultaneous drop in incorrect outcomes by nearly 10%. The drop of course is primarily compensated for with an increase in unclassified cases. These results clearly show the major and minor effects of adding relationships that were mentioned before.

Tables 17 and 18 depict the evaluation of the conceptualization outcomes for the methods M3 and M4, with respect to the range of outcomes obtained. Place classification accuracy = 100% in every cycle of testing, for both M3 and M4. Here too, the results show that on an average, the range of results obtained using M4 was superior to that

Table 17
LKOCV M3 accuracy - Range

	Classified cases		Overall cases	
Outcome	Mean	Std. Dev.	Mean	Std. Dev.
Incorrect	32.3037	17.3204	31.5406	17.1243
Not classified	0.0	0.0	2.6165	2.4527
Free Object	0.7496	1.3286	0.7363	1.3068
Correct	66.9467	17.2223	65.1065	16.6033

Table 18
LKOCV M4 accuracy - Range

	Classified cases		Overall cases	
Outcome	Mean	Std. Dev.	Mean	Std. Dev.
Incorrect	25.6077	16.8771	23.0740	15.3331
Not classified	0.0	0.0	11.6876	8.3612
Free Object	1.0638	1.6885	0.9167	1.4267
Correct	73.3285	16.7693	64.3216	13.9219

obtained from M3. A similar interpretation can be drawn about the results as in the KFSCV test. For the classified cases, M4 performs much better than M3.

4.4. *Experiments on a real Robot*

Experiments have been conducted on a real robot platform with real sensor data [1]. The experiments are meant to demonstrate the integration of the cognition algorithm (M4) as described in this report together with prior work on object based robot mapping, described in [10]. Figures 9 and 10 depict the output obtained and ground truth for an office. The output demonstrates that the robot has been able to construct an object(feature) based relative metric map of the place ; it has been able to use the objects and the metric information stored in the map to form higher level semantic constructs; and finally, it has been able to use these higher level semantic constructs to infer that the place is an office. In the figure, objects of a particular cluster are denoted by a single color and a number in parenthesis. The legend of the figure displays the result of the conceptualization process, the concept inferred and the belief in that concept are depicted.

A salient aspect of the approach is the endowment of the robot with a greater level of spatial awareness. The office mapped in the figure 10 had a completely different arrangement (with some different objects), when the dataset used for learning was first collected (time difference $\approx$ 1 year). With time, even though the office would be a different one in a topological sense - the robot, if it found itself in this place, would at-least be aware that it is in an office. This would provide at-least a semantic localization capability that could be used to filter out place hypotheses that cannot occur.

A second experiment was conducted in the kitchen (refreshment room) of our laboratory premises [2]. Figure 11 displays the output of the approach when applied on this dataset. The particular room was a challenging one for two reasons - noisy sensor data and not a picture-perfect description of a kitchen. The following points are worth observing:

- The larger-font black text in the figure is meant to clarify objects which may be too close to each-other. The text below the legend of the figure lists the objects in the right part of the image/room, again for clarity sake. The image is a 2D top-down depiction of a 3D map. Objects to the right of the image (with particular reference to the objects around the cooking-space) are relatively on the inside / above those on the left.
- There is only one coffee machine in the ground truth. Two occur in the image due to "noisy" stereo data for one particular observation (upon recognizing a partial view of the coffee machine). This can be dealt with using appropriate estimation techniques, these are not the focus of this work. Note that a separate test was performed to confirm that cluster 6 would indeed be inferred as a cooking-space even if the coffee-machine was not present in it. The water-heater in cluster 10 occurs because of the occurrence of a new cluster (cluster 10) that is closer to it than the cooking-space (cluster 6). Note that data association during mapping, from one sensory observation to another, is done using a nearest neighbor filter between objects observed in each sensory observation and those already present in the map (at the level of individual clusters).
- The "chairs" in cluster 1 and 2 correspond to the sofas in the figure 12. This is due to the lack of a sofa object in the robot's learnt models. (a chair and a sofa would perform the same function in this context).
- The dataset is not an easy one as it doesn't contain some very typical objects of a kitchen such as a cooking range and an oven. This particular place has never been learnt/tested in prior datasets. However, even in the presence of incomplete and uncertain (some noisy stereo data) sensory information, the final place classification works out appropriately to that of a kitchen. In this sense, this data-set demonstrates a level of robustness in the approach.
- Single object clusters such as 12 and 13 are inferred as storage spaces due to two reasons - (1) unbalanced learning dataset which has a greater number of storage-space instances (refer appendix) and (2) the absence of most

[1] A movie of the experiment is available at http://www.asl.ethz.ch/research/asl/cogniron. The movie *imagesOffice.avi* shows a mobile robot moving around in an office and recognizing objects whereas the movie *objectmapOffice.avi* shows the resulting object based relative metric map formation, conceptualization and place classification processes.

[2] A movie of the experiment is available at http://www.asl.ethz.ch/research/asl/cogniron. The movie *imagesKitchen.avi* shows a mobile robot moving around in a kitchen and recognizing objects whereas the movie *objectmapKitchen.avi* shows the resulting object based relative metric map formation, conceptualization and place classification processes.

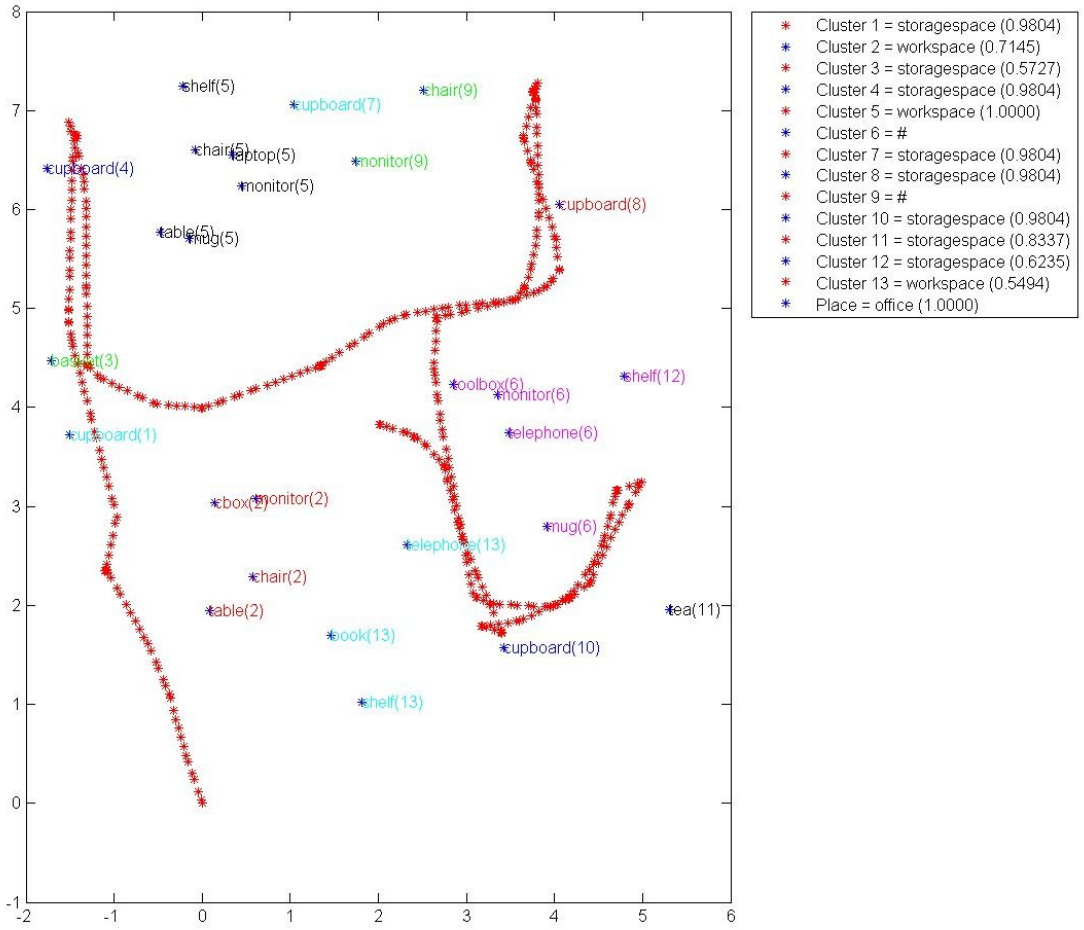


Fig. 9. Bayesian Space Conceptualization and Place Classification by a Mobile Robot. The figure depicts a robot exploring a room, recognizing objects, creating a probabilistic object graph based map of the room, using the objects and inter-object relationships perceived to probabilistically form higher level concepts and finally using these concepts to classify/infer on the place. The representation thus obtained is a semantically enriched one; the robot can thus demonstrate a greater degree of spatial awareness. Stereo vision together with a SIFT based object recognition system are used in conjunction with odometry to perform this experiment. Note that the deviation in the robots path is expected as only odometry is used. This can be corrected using scan-matching techniques, for instance. The end product is a hierarchical probabilistic concept oriented representation of the office. Objects are incrementally clustered (as perceived) and each cluster is inferred to be an instance of a particular concept. In the figure, objects of a particular cluster are denoted by a single color and a number in parenthesis. The legend of the figure displays the result of the conceptualization process with the concept inferred and the belief in that concept being shown.

Fig. 10. An approximate ground truth for office mapped in figure 9 (taken at a later time than actual experiment)

other objects / relationships in the cluster (inference is based on both the evidence present and that which is absent). This can be dealt with by constraining the system manually to infer only after a certain amount of evidence is accumulated (usually 2 objects is enough to get more meaningful concepts) but for little evidence and a system that has learnt from unbalanced data, the system is bound to favor the concept best represented amongst the training instances.

5. Discussion

5.1. *On the presented approaches*

(i) In all, four approaches to Bayesian Space Conceptualization and Place Classification for a mobile robot have been presented in this report. While the first two were presented in brief (with the first one being a naive approach) and the findings of experiments conducted on them have been mentioned, the later two approaches have been studied in detail to understand their generalization performance and the effect of incorporating relationships in the cognition process.

(ii) M1 - This was a naive approach presented in an earlier attempt at the larger goal of this work. It had the following issues
- Semantics was represented only by the presence of objects. This was the main limitation.
- There was no learning from negative exemplars.
- Inference was only based on the evidence at hand.
- The approach did not handle multiple object occurrences.

(iii) M2 - It addressed all previously mentioned limitations by using an appropriately chosen likelihood function and grounding the approach on a systematic Bayesian Programming formalism. The behavior of this algorithm was characterized by very high classification accuracy but low classification rate. This was primarily because every occurrence of each object was contributing to each concept to an equal extent. This problem was addressed in the subsequent approaches.

(iv) Clustering - In all the tests conducted (preliminary tests, LKOCV & KFSCV), the clustering process produced similar results with about 1% of the objects being incorrectly/singly clustered, between about 30 and 34 % of the objects being fused or broken into non-native clusters and about 69 and 65 % respectively, of the objects being correctly clustered with respect to the training input.

(v) M3 - Object count proved to be a better feature than object presence (M1) or object significance/importance (M2). Both in terms of classification rate and accuracy, this model performed better than M1 and M2. Further, incorporating a Gaussian uncertainty in the training input improved the generalization capability in that the algorithm could handle conceptually adjacent cases better.

(vi) M4 - The incorporation of relationships had two positive effects on the conceptualization process - one major and the other minor. The major effect was that a significant proportion of previously incorrect outcomes were now not-classified pending further evidence. Since inability to classify is a better outcome than misclassification, this effect is interpreted as a positive outcome and is vastly responsible for the reduction in the number of incorrect outcomes. The minor effect was the correct interpretation of a small number of objects that were previously incorrectly conceptualized.

(vii) In the experiments presented, the incorporation of relationships resulted in the following outcome - for a similar or slightly better number of correct outcomes, the relationships enable a very significant drop in incorrect outcomes with a corresponding increase in number of unclassified cases. Both LKOCV and KFSCV exhibited this behavior with approximately similar numbers. Of the classified cases, incorporating relationships clearly increases the ratio of the correct outcomes to the incorrect outcomes. Place classification was perfect for both models (not so for prior models like M2). Thus, the incorporation of relationships (specifically, distance in this work) leads to an improvement in the classifier capability, it adds additional constraints to a purely object based model in order to more clearly define the concept under consideration.

(viii) The approaches presented clearly lead to an increase in semantic content in mobile robot representations. The approach also enables a robot to gain an intermediate level of understanding before making an inference about the place as a whole. This ability will also function to some extent as a filter (incorporating some robustness) for higher level inference. The approaches presented clearly have the ability to map sensory information to increasingly abstract concepts.

5.2. *Extending the approach*

5.2.1. *Incorporating directional relationships*

While distances between objects go some way in constraining a system of objects to a particular configuration (which is learnt as the model of a particular concept), the directional relationships between objects may play a significant role in this context. This information is metrically represented in terms of the angular relationships between the objects, encoded in the map of the developed. While the framework would be exactly the same as that presented in M4 earlier, the precise modeling of the relationships needs further research. This is ongoing work.

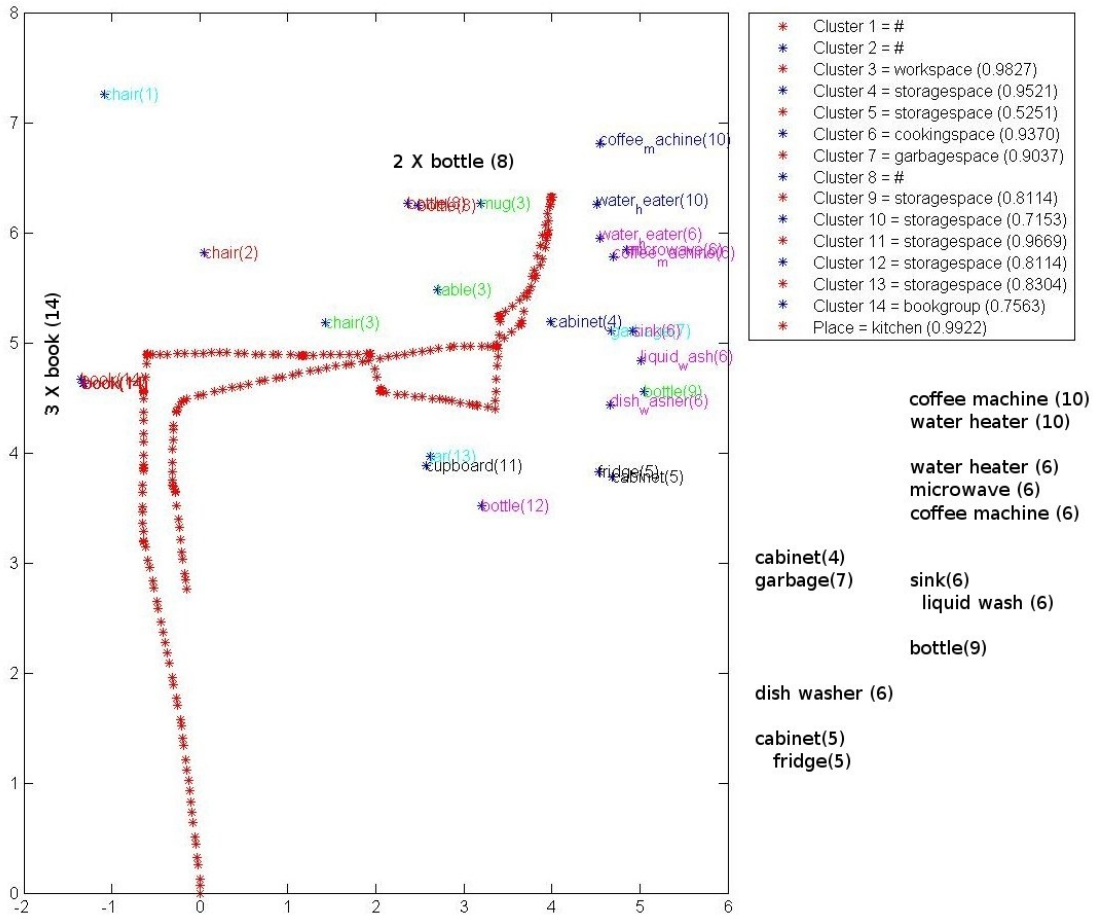


Fig. 11. Bayesian Space Conceptualization and Place Classification by a Mobile Robot. The figure depicts a robot exploring a room, recognizing objects, creating a probabilistic object graph based map of the room, using the objects and inter-object relationships perceived to probabilistically form higher level concepts and finally using these concepts to classify/infer on the place. The representation thus obtained is a semantically enriched one; the robot can thus demonstrate a greater degree of spatial awareness. Stereo vision together with a SIFT based object recognition system are used in conjunction with odometry to perform this experiment. Note that the deviation in the robots path is expected as only odometry is used. This can be corrected using scan-matching techniques, for instance. The end product is a hierarchical probabilistic concept oriented representation of the kitchen. Objects are incrementally clustered (as perceived) and each cluster is inferred to be an instance of a particular concept. In the figure, objects of a particular cluster are denoted by a single color and a number in parenthesis. The legend of the figure displays the result of the conceptualization process with the concept inferred and the belief in that concept being shown. Note that *'n X object'* is used to denote n occurrences of an object.

(a) (b)

Fig. 12. An approximate ground truth for kitchen (refreshment room) mapped in figure 11 (taken at a later time than experiment)

5.2.2. *Improving object representations*

This work could be extended by using 3D bounding boxes of objects that can describe a richer set of semantic relationships between objects - such as the concepts of "touching" and "facing". An appropriate object perception system that can give accurate viewpoint results together with appropriate mathematical formalisms describing the relationships would be required to realize this. This is the projected goal of this work.

5.2.3. *Improved clustering*

The work can also be improved from the clustering point of view. Bad clusters lead to bad concepts i.e. incorrect semantics. The clustering model has been intentionally kept the same in order to facilitate comparison and understand the effectiveness of each feature in the context of conceptualization. However, the clustering algorithm is order dependent and the ability to make/break certain links between objects and clusters on the fly, during the clustering-conceptualization process, will make a useful contribution in improving the performance of the algorithm presented.

5.2.4. *Towards Hierarchical (Semantic - Topological - Metric) SLAM*

While the presented work was not aimed at addressing the SLAM problem directly, it proposes a representation of space that is firmly grounded in the state-of-the-art in mobile robotics and yet extends it to address the lack-of-semantics problem in it. The representation developed in [10] as well as its extension in this work is basically a metric map which also implicitly encodes the topology of space (connectivity between doors, [10]) as well as semantics (concepts and places categories) within it. Higher level abstractions are generated from the underlying metric representation of space. Thus, representation can also be viewed as a global topological map (with local metric maps) and as a collection of concept models (semantic maps). Further, [10] and the work presented here address the issues of place classification (an area of contribution) and place recognition; localization in a relative metric map has been studied by the SLAM community ([35] for instance). While place recognition is a form of topological localization, place classification can be thought of as a form of *Semantic Localization*. Thus, this work significantly addresses and forms a basis for a new kind of Hierarchical SLAM, one that is Semantic - Topological - Metric in nature. To bring these elements together within a SLAM context would be a direct and useful extension of this work.

6. Conclusion

A Bayesian approach to conceptualization and classification of space for mobile robots was presented. The suggested algorithm was based on the Naive Bayes Classifier (NBC) and was implemented using a clustering mechanism and a sound Bayesian Programming methodology. The concept models included an object model that encoded the likelihood of observing a specific number of instances of a certain object, in an instance of the concept and a relationship model that encoded the most characteristic relationships using a Gaussian mixture model. The entire model was probabilistic and all stages of the work (training / testing) worked on uncertain data. The incorporation of relationships over and above the object count model resulted in an improved performance of the classifier, in that it made much less errors for a comparable or slightly better rate of accuracy. The algorithm incrementally formed conceptual groups of objects - these represented semantic (functional) groupings that were aimed at capturing spatial semantics; further, they were used for classifying places. The generated concepts increase the amount of semantic information contained in a robot's spatial representation. They also endow the robot with the capability of being more spatially aware machines, capable of reasoning about spatial semantics.

7. Appendix

Experiments were conducted on a dataset that included physically measured object and coordinate information from 11 offices and 8 kitchens. The office data was represented in terms of three concepts (apart from some free-standing objects). These were work-space, storage-space and meeting-space. The kitchen data was described in terms of ten concepts, namely cooking-space, garbage-space, dining-space, bottle-group, glass-group, box-group, mug-group, bag-group, poster-group and book-group. Concepts used in this work represent the manner in which the places were understood by the authors; they are similar to those observed in [9]. The approach however is not ontology-specific.

The dataset is a highly unbalanced one, in that some concepts are extensively represented, while some others are not. This aspect, while not desirable and contributive to the high variance obtained in the results presented in this report, is quite realistic and expected to occur in real world scenarios. Hence, this dataset was used.

Some details regarding the dataset:

- Number of offices = 11
- Number of kitchens = 8
- Number of concept types in dataset = 13
- Number of concept instances = 172
- Number of object types in dataset = 77
- Number of objects in dataset = 991
- Number of free objects in dataset = 9
- Number of instances of individual concepts:
- Number of objects in individual places:

  Places 1-11 are offices whereas places 12-19 are kitchens.

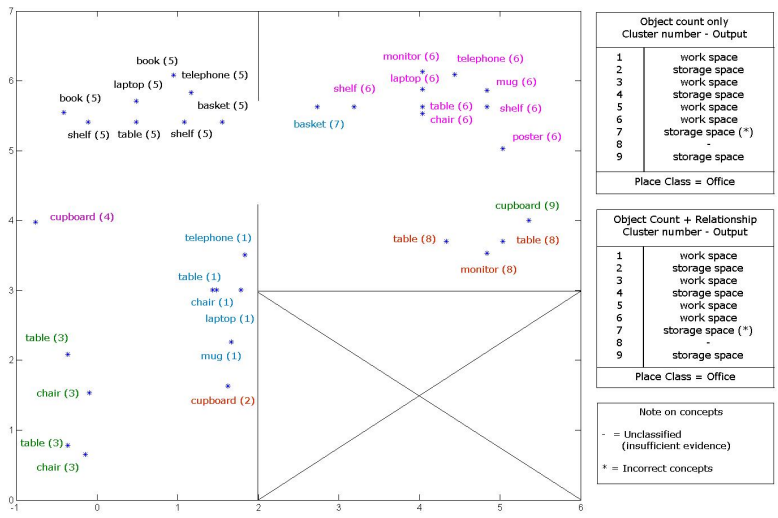


Fig. 13. Outcome of the clustering, conceptualization and place classification processes for an office. The depiction is a top-down view. Each cluster of objects is identified by a color and a number in parenthesis. On the right are the outcomes as obtained using the two models - M3 (object count only) and M4 (object count + relationship). Each cluster is classified as being one of 13 concepts used in this work. Note that the basket in cluster 7 is classified as a storage space. This is primarily due to the non-occurrence of all other known objects in that cluster and the prior probability of the occurrence of the storage-space concept in relation to that of other concepts. Both models (M3/M4) yield identical results in this particular case.

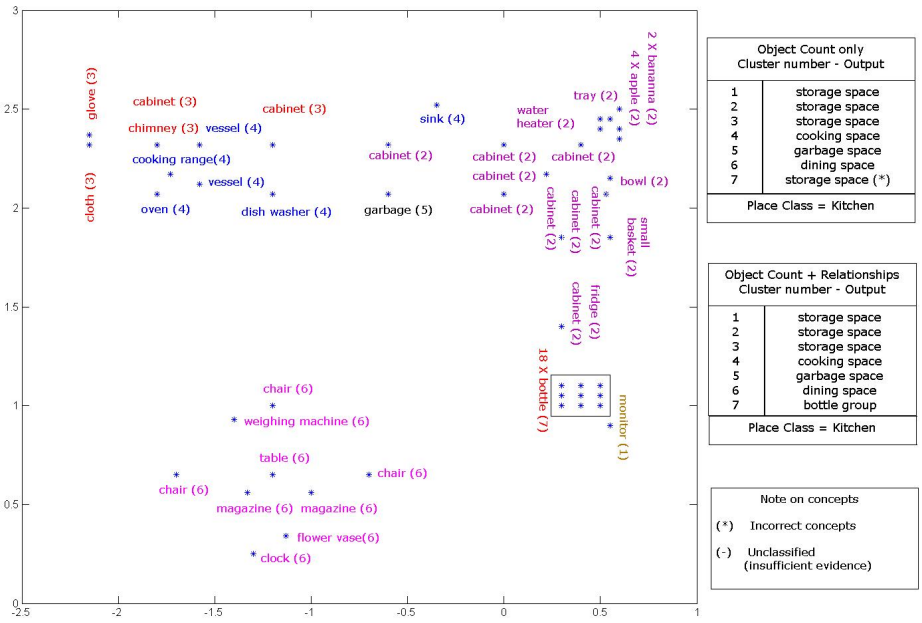


Fig. 14. Outcome of the conceptualization and classification processes for a kitchen. The depiction is a top-down view. Each cluster of objects is identified by a color and a number in parenthesis. The rectangles depict the cabinets containing various objects within it in different rows. Each cluster of objects is identified by a color and a number in parenthesis. Note that *'n X object'* is used to denote n occurrences of an object. On the right are the outcomes as obtained using the two models M3 (object count only) and M4 (object count + relationship). Each cluster is classified as being one of 13 concepts used in this work. Note that cluster 7 is an example of a case where incorporating relationships actually helps convert an incorrect outcome (obtained using object count only) to a correct one.

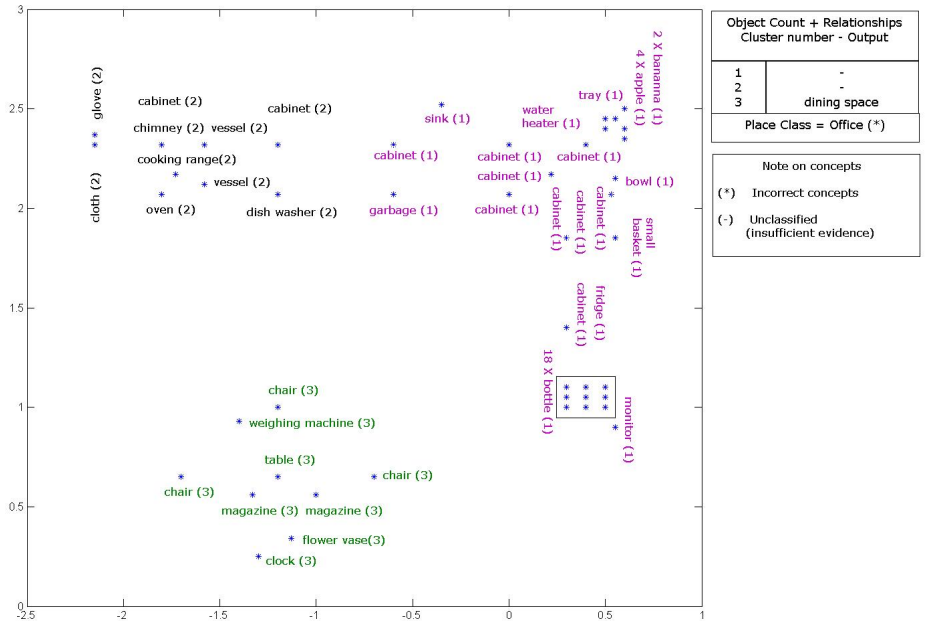


Fig. 15. Outcome of the clustering, conceptualization and place classification processes, as applied to a kitchen, using only a nearest neighbor clustering algorithm (no concept models used in clustering process). The depiction is a top-down view. Each cluster of objects is identified by a color and a number in parenthesis. The rectangles depict the cabinets containing various objects within it in different rows. Each cluster of objects is identified by a color and a number in parenthesis. Note that *'n X object'* is used to denote n occurrences of an object. On the right are the conceptualization and place classification outcome as obtained using M4 (object count + relationship) model. When compared with figure 14, clearly the clustering in this case does not take into account object level semantics in order to form clusters. This results in grouping objects together which could otherwise not occur together. Using the concept models in the clustering process would result in a more semantically appropriate and human like grouping of objects.

Table 19
Concept distribution in dataset

Concept	N(instances)	Concept	N(instances)
Work-space	45	Storage-space	63
Cooking-space	14	Garbage-space	13
Bottle-group	13	Box-group	9
Dining-space	4	Bag-group	3
Glass-group	2	Mug-group	2
Poster-group	2	Book-group	1
Meeting-space	1	–	–

Table 20
Object distribution in places

Place	N(objects)	Place	N(objects)	Place	N(objects)	Place	N(objects)
1	38	6	51	11	50	16	44
2	31	7	48	12	31	17	88
3	33	8	18	13	120	18	60
4	28	9	35	14	54	19	122
5	35	10	59	15	46	–	–

Acknowledgments

The authors thank all people who allowed the use of their offices and kitchens for data collection. This work has been supported by the EC under FP6-IST-002020-COGNIRON, FP6-IST-027140-BACS and FP6-2006-IST-6-045350-Robots-At-Home. The authors thank Ahad Harati for his support and suggestions on aspects of the work, Arnau Ramisa for his support in improving the object recognition module for the real robot experiments, Cedric Pradalier and Rudolph Triebel for their suggestions on the report.

References

[1] S. Thrun, Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann, 2002, Ch. Robotic mapping: A survey.

[2] K. O. Arras, Feature-Based Robot Navigation in Known and Unknown Environments, Ph.D. thesis, Swiss Federal Institute of Technology Lausanne (EPFL), Thesis number 2765 (2003).

[3] R. Chatila, J. P. Laumond, Position referencing and consistent world modeling for mobile robots, in: IEEE International Conference on Robotics and Automation, 1985.

[4] H. Choset, K. Nagatani, Topological Simultaneous Localization And Mapping (SLAM): toward exact localization without explicit localization, IEEE Transactions on Robotics and Automation 17 (2001) 125–137.

[5] A. Tapus, Topological SLAM - Simultaneous Localization And Mapping with fingerprints of places, Ph.D. thesis, Swiss Federal Institute of Technology Lausanne (EPFL), Thesis Number 3357 (2005).

[6] S. Thrun, Learning Metric-Topological Maps for Indoor Mobile Robot Navigation, Artificial Intelligence 99 (Issue-1) (1998) 21–71.

[7] N. Tomatis, I. Nourbakhsh, R. Siegwart, Hybrid Simultaneous Localization And Map building: A natural integration of

topological and metric, Robotics and Autonomous Systems 44 (2003) 3–14.
[8] I. Ulrich, I. Nourbakhsh, Appearance-Based Place Recognition for Topological Localization, in: IEEE International Conference on Robotics and Automation (ICRA), San Francisco, CA, USA, 2000, pp. 1023–1029.
[9] S. Vasudevan, S. Gächter, R. Siegwart, Cognitive Spatial Representations for Mobile Robots - Perspectives from a user study, in: IEEE Int. Conf. on Robotics and Automation (ICRA) Workshop on Semantic Information in Robotics, Rome, Italy, 2007.
[10] S. Vasudevan, S. Gächter, V. T. Nguyen, R. Siegwart, Cognitive Maps for Mobile Robots - An object based approach, Robotics and Autonomous Systems 55 (5) (2007) 359–371.
[11] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[12] B. W. Wah, Generalization and Generalizability Measures, IEEE Transactions on Knowledge and Data Engineering 11 (1) (1999) 175 – 186.
[13] D. H. Fisher, Knowledge Acquisition Via Incremental Conceptual Clustering, Machine Learning 2 (1987) 139–172.
[14] B. Taskar, E. Segal, D. Koller, Probabilistic Classification and Clustering in relational data, in: Seventeenth International Joint Conference on Artificial Intelligence (IJCAI), Seattle, Washington, USA, 2001.
[15] J. B. Tenenbaum, Bayesian Modeling of human concept learning, in: S. S. M.S. Kearns, D. Cohn (Eds.), Advances in Neural Information Processing Systems (NIPS) 11, MIT Press, Canbridge, MA, USA, 1999.
[16] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.-A. Fernandez-Madrigal, J. Gonzalez, Multi-Hierarchical Semantic Maps for Mobile Robotics, in: IEEE/RSJ Interrnational Conference on Intelligent Robots and Systems (IROS), Edmonton, Canada, 2005, pp. 3492–3497.
[17] Ó. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, W. Burgard, Supervised semantic labeling of places using information extracted from sensor data, Robotics and Autonomous Systems 55 (5) (2007) 391–402.
[18] Ó. M. Mozos, P. Jensfelt, H. Zender, G.-J. M. Kruijff, W. Burgard., From Labels to Semantics: An Integrated System for Conceptual Spatial Representations of Indoor Environments for Mobile Robots., in: IEEE Int. Conf. on Robotics and Automation (ICRA) Workshop on Semantic Information in Robotics, 2007.
[19] A. Y. Ng, M. I. Jordan, On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes, in: Advances in Neural Information Processing Systems (NIPS) 14, MIT Press, 2002.
[20] J. Cheng, R. Greiner, Learning Bayesian Belief Network Classifiers: Algorithms and System, in: Canadian Conference on Artificial Intelligence (CSCSI), Ottawa, Canada, 2001.
[21] A. K. Jain, M. N. Murty, P. J. Flynn, Data Clustering: A Review, ACM Computing Surveys 31 (3) (1999) 264–323.
[22] O. Lebeltel, P. Bessière, J. Diard, E. Mazer, Bayesian Robots Programming, Autonomous Robots 16 (2004) 49–79.
[23] J. Vogel, B. Schiele, Natural scene retrieval based on a semantic modeling step, in: International Conference on Image and Video Retrieval, Dublin, Ireland, 2004.
[24] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
[25] E. Sudderth, A. Torralba, W. Freeman, A. Willsky, Learning hierarchical models of scenes, objects, and parts, in: IEEE International Conference on Computer Vision (ICCV), 2005.
[26] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2003.
[27] G. Bouchard, B. Triggs, Hierarchical part-based visual object categorization, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
[28] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley Interscience, 2000.
[29] S. Vasudevan, R. Siegwart, A Bayesian Conceptualization of Space for Mobile Robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, USA, 2007, (more detailed Technical Report available on author webpage).
[30] S. Vasudevan, A. Harati, R. Siegwart, A Bayesian Conceptualization of Space for Mobile Robots : Using the Number of Occurrences of Objects to Infer Concepts, in: 3rd European Conference on Mobile Robotics (ECMR), 2007.
[31] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[32] N. Vlassis, J. J. Verbeek, Gaussian mixture learning from noisy data, Tech. Rep. IAS-UVA-04-01, Informatics Institute, University of Amsterdam, The Netherlands (September 2004).
[33] G. Schwarz, Estimating the dimension of a model, Annals of Statistics 6(2) (1978) 461–464.
[34] R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, in: the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI), 1995, pp. 1137–1143.
[35] A. Martinelli, A. Svensson, N. Tomatis, R. Siegwart, SLAM based on quantities invariant of the robot's configuration, in: IFAC Symposium on Intelligent Autonomous Vehicles, 2004.