2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

A visual bag of words method for interactive qualitative localization and mapping

David FILLIAT
ENSTA - 32 boulevard Victor - 75015 PARIS - France
Email: david.filliat@ensta.fr

Abstract— **Localization for low cost humanoid or animal-like personal robots has to rely on cheap sensors and has to be robust to user manipulations of the robot. We present a visual localization and map-learning system that relies on vision only and that is able to incrementally learn to recognize the different rooms of an apartment from any robot position. This system is inspired by visual categorization algorithms called bag of words methods that we modified to make fully incremental and to allow a user-interactive training. Our system is able to reliably recognize the room in which the robot is after a short training time and is stable for long term use. Empirical validation on a real robot and on an image database acquired in real environments are presented.**

I. Introduction

Localization is a basic requirement for many robotic applications. This capacity in complex environments usually relies on a map which can be either given to the robot, or learned while the robot discovers its surroundings. For many applications including personal and entertainment robotics, a system able to autonomously build a map while estimating the robot position is the best solution. This problem is usually referred to as Simultaneous Localization and Mapping (SLAM) [1].

We present in this paper a SLAM method specifically adapted for small personal robots with humanoid or animal shape, or at least equipped with a camera which direction can be controlled. This method relies on vision only and is *qualitative*, i.e. it is able to recognize the room the robot is in, but not give a precise metrical position. We have taken into account the fact that these robots are seldom autonomous and are very often manipulated by their user : our system is very robust to robot position and can recognize a room from a wide variety of point of view (figure 1). It is also very simple to train and relies on a simple interaction with the user that avoids any tedious separate map learning stage. It is directly inspired from bag of words methods used in object categories learning developed in the computer vision community.

In the next section, we explain the choices made in our localization system, then we present the bag of words method and show how we adapted these algorithms for incremental qualitative localization and mapping. We present extensive validation results in section V and discuss our method in section VI.

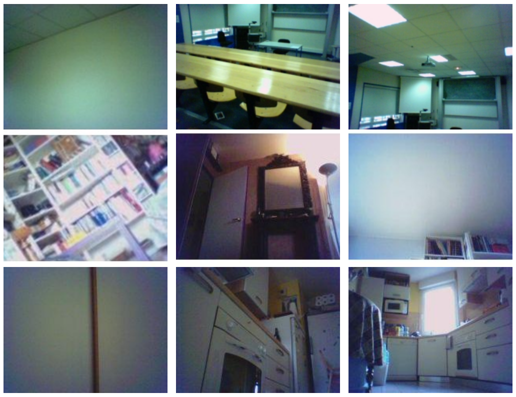

Fig. 1. Example of images used in this paper. The images in each row are taken in the same room. Note that some of the images contains almost no information about the position.

II. Localization and mapping for small humanoid robots

Navigation systems may use either topological or metrical maps [2]. In topological maps, only places such as rooms and their relations are learned and recognized [3], whereas in metrical maps, the precise metrical positions of environment features and of the robot are estimated [4], [5]. Of course several authors proposed hybrid approaches taking advantages of both representations [6].

In realistic scenarios for entertainment robotics, the robot can be moved directly by the user from one place to another, can fall or can be blocked where sensors will have difficulty to find useful information (e.g. under tables, in corners...). In these situations, a metrical approach, that usually requires a continuous tracking of features, will be very difficult to use. In a lot of situations, the localization algorithm will have to perform *global localization*, i.e. to estimate the current position without any reference to the previous position [2]. Global localization is usually more easily performed by topological methods, as they only give a broad estimation of the position.

The system we designed is *qualitative* : it estimates the position as a topological method, but without using the room relations. This estimation could then be used to initialize a metrical localization, to trigger location specific behaviors, such as looking for the charging station if the robot is in the correct room, or giving the context for a user-robot interaction (e.g. talking about cooking when in the kitchen).

1-4244-0602-1/07/$20.00 ©2007 IEEE.

The current most efficient navigation systems rely on laser range finders associated with metrical [4] or topological [6], [7] maps. However, these sensors are quite heavy and expensive and are not well suited for small personal robotics.

Navigation systems using a camera are more adapted and have been developed using both metrical [5] and topological [8], [9] approaches. In these later approaches the use of a panoramic camera is common and simplifies the problem by making all necessary information available at one time. In a humanoid or animal-like robot context, however, the use of a standard gaze-controlled camera is more natural, but imposes to actively search for information as a lot of images could be useless (e.g. when the robot is closely facing a wall or a furniture).

Most localization system for autonomous robots use odometry or temporal coherency of the position to enhance the localization quality, often using a bayesian approach (e.g. [8], [10], [11]). However, in our context, this information is not available as soon as the robot is moved by the user, or shut down in a room and switched on in another. Moreover, our goal was to design a ”one shot” localization procedure that would give the position without robot movements (except the head). Therefore, we chose not to use this temporal information in this work.

In a personal robotic context, the training has to be very simple, so we chose to base the map learning on a continuous interaction with the user, and not on a separate learning phase. Our mapping system therefore rely on occasional user supervision, learning only when the user reports an error and gives the correct position. This also let the user impose the discretization of the environment and the name of the rooms.

Following these choices, our qualitative localization and mapping system has therefore to learn and to recognize the current position using only images. This problem is related to a categorization problem of the computer vision community : to infer the category of an object given images of this object. Our system is based on the state-of-the-art *bag of words* methods [12] used to solve this problem, adapted to our context.

III. Incremental Bag of Words method for visual localization

A. Overview of the method

The goal of object category learning is to build a classifier that will detect objects from given categories (e.g. car, faces...) in an image. A popular method [13] is to use a representation of the images as a set of unordered elementary features (the words) taken from a dictionary (or codebook). Using a given dictionary, the classifier is based on the frequencies of the words in an image. The term ”bag of words” refers to document classification techniques that inspired these approaches where documents are considered as unordered sets of words.

The words used in image processing are local image features. They may be constructed around interest points such as scale-space extrema (e.g. SIFT keypoints [14]), or simply on windows extracted from the image at regular positions and various scales. The features can be image patches, histograms of gradient orientations or color histograms [13]. As these features are sensitive to noise and are represented in high dimension spaces, they are not directly used as words, but are categorized using a vector quantization technique such as k-means. The output of this discretization is the dictionary.

Based on the words, a classifier is then trained to recognize the categories. Different techniques can be used such as Support Vector Machines (SVM), or Naive Bayes Classifiers [12]. Categorizing an image then simply entails extracting features, finding the corresponding words and applying the classifier to the set of words representing the image.

B. Adaptation to visual localization

In image classification, dictionary building and classifier training are performed off-line on image databases. We modified these processes to make them incremental.

Dictionary construction, which entails clustering the image features to create the words, if often performed using k-means, which requires the processing of a database of examples. In [13], the authors report limitations of this approach and show that better results are achieved using a fixed radius clusterer. The method we used, detailed in section IV, is similar but simpler and fully incremental.

The goal of the classifier is to infer the room from an image. This classifier should be trained incrementally, i.e. it should be able to process new examples and add new categories without the need to reprocess all the previous data. This feature is not common for the classical machine learning algorithms like SVM or boosting. For this reason, we used a voting method in which training simply entails estimating word statistics, and classifying simply entails reading these statistics, without any complex training process.

Moreover, in our context, two characteristics can be taken into account. First, some images could belong to several categories and bring no information about the position (for example, a picture of the ground if the ground material is the same in all rooms). Second, several images taken from a given position by moving the robot head are sure to belong to the same room and can be used to infer the category. These properties enable us to use an active search for information : it could be decided if an image should be used or not according to the information it carries, and new images could be taken into account if the quality of the current position estimation is not high enough. We estimate the localization quality from the vote results (see section IV).

On-line learning makes it possible to select relevant examples for the task at hand, thus making possible the use of an active learning method [15]. In our case, we learn only for the pictures which have been incorrectly classified after user feedback. This leads to a reduction in the need of training data and an improvement of the learning speed and accuracy in the spirit of popular algorithms like boosting [16]. In our context, it also enables a stabilization of the learning algorithm in the long term as learning is less and less performed as performance increase.

As for the features used for image characterization, the suitable set depends on the environment the robot is facing. In an environment where rooms have different colors, color histograms would be a good choice, while in mostly black and white environments, texture or image patches would be more appropriate. A unique optimal feature set is therefore difficult to select a priori. We have taken advantage of the voting scheme used for classification to integrate several feature spaces in a very natural way. Experiments reported in this paper make use of color, texture and local gradient orientation (see section IV).

IV. Visual localization and mapping algorithm

We will first describe the localization algorithm, assuming the map (i.e. the dictionary and associated statistics) has already been learned. We will then describe map learning, and finally give a description of the different feature used.

A. Localization

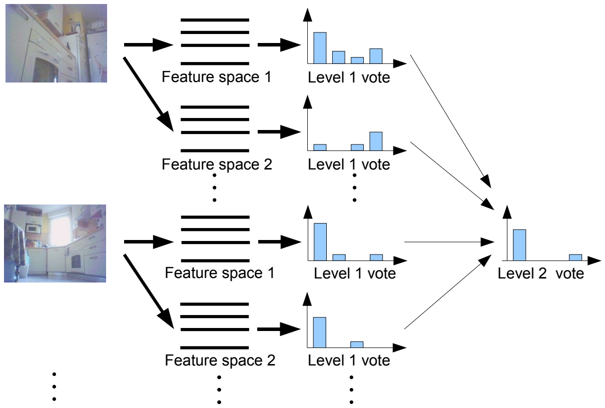

Fig. 2. Illustration of the two stage voting method used for localization.

A two stage voting method is used to estimate the robot position (figure 2). In a given robot position, a first picture is taken from a random head position. The features are extracted and the corresponding words are found in the dictionary. These words then vote at the first level for the rooms in which they have been perceived at least once. We only use features which correspond to a known word and we don't take into account the votes of words that have been seen in all rooms as they carry no information. The unknown features are stored for the learning phase (see next section).

A quality of the vote result is calculated as the percentage of vote represented by the difference between the maximum and the second maximum :

$$quality = \frac{n_{Winner} - n_{Second}}{\sum_i n_i}$$

where n_i is the number of votes for category i.

In order to take only informative images into account, the winning category votes at the second level only if the quality and the number of words are above some thresholds.

This process is repeated with the other feature spaces and with new images until the quality of the second level vote (estimated with the same method) reaches a given threshold (0.5 in all experiments) or a given number of images is reached (10 in all experiments). The room is then considered recognized if the quality threshold has been reached, or no recognition is made if the limit number of images has been reached. The new images taken for localization are taken with a new random head direction.

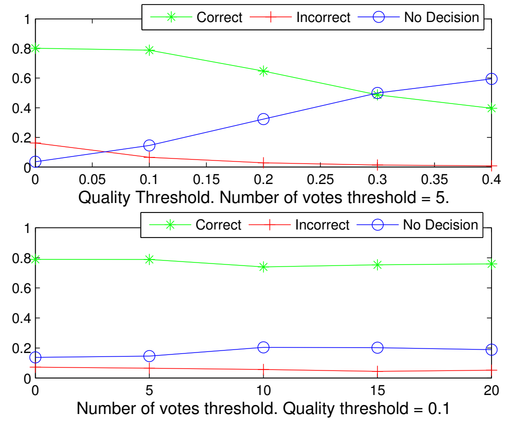

Fig. 3. Influence of the quality of vote and number of vote thresholds. The graph shows the percentage of correct, incorrect and undecided localization with a previously learned map. Results are a mean of 1000 localization experiments in 20 different random environments (see section V).

Experiments show that the quality threshold is very effective in reducing the false recognitions and has therefore to be chosen carefully to make a tradeoff between making mistakes and making no decision. While intuitively more votes should lead to a more reliable result, the threshold on the number of votes has in practice little influence (figure 3).

B. Mapping

The mapping procedure is interactive and processes images upon user feedback after the localization procedure is performed. If the user declares the localization incorrect, learning is performed using all the features that have been used for localization and the position label given by the user. Learning the map entails two processes : building the dictionary and gathering data for the classifier. These processes are incremental and require only a few computations.

The dictionary construction relies on an incremental nearest neighbor classifier. For a new feature, the closest word is found in the current dictionary. If the distance between the word and the feature is below a threshold, the word is recognized, else a new word initialized to the feature position is added. This method is clearly sensitive to noise in the feature extraction and to the order of feature processing, problems that are solved when using a batch method such as k-means. However our method is fully incremental and these limitations didn't appear to be a problem in our application.

As we use a voting method as classifier, map learning simply entails memorizing in which category a given word has been perceived, i.e. labeling all the words of the learned images with the label of the position.

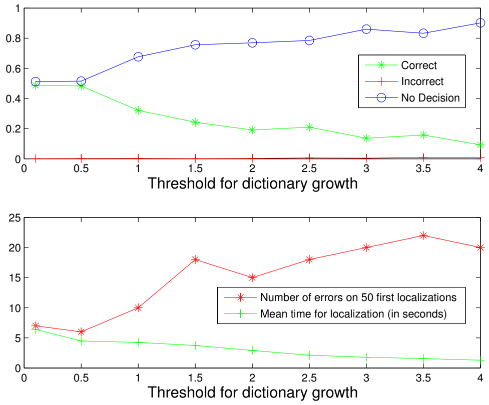

Fig. 4. Influence of the dictionary growth threshold in the color histograms feature space. Results are a mean of 20 mapping sessions in different random environments (see section V). The top graph show the performance obtained on 50 localization experiments after 100 localization and mapping steps.

The threshold used for dictionary construction influences several factors. Increasing the threshold decreases the number of words, thus speeding the search for the words corresponding to the features, but degrades localization performance. In particular an increase of the threshold value makes it longer for the statistics on the words to become meaningful, thus increasing the number of localization errors at the beginning of the map learning process (figure 4).

The threshold also has to be set independently for each feature space, but, once chosen, dictionary construction performs similarly on all the environments we tested. The thresholds used in this paper were empirically tuned on our database (see section V) and perform well in real situations.

C. Image features

For all the experiments described in this paper, we used three different feature spaces :

- SIFT keypoints (Scale Invariant Feature Transform) [14]: interest points are detected as the maximum over scale and space of the convolution by differences of gaussians. Keypoints are histograms of gradient orientations around the detected point. These keypoints are invariant in scale and rotation and are among the most efficient for recognition [17]. The descriptor used are of dimension 128.
- Local color histograms : The image is decomposed in a set of adjacent windows of constant size. The histograms of the H value in the HSV color space for each window are used as features. The windows used are of size 40x40 pixels and the descriptors of dimension 16.
- Local normalized grey level histogram: The image is decomposed in a set of adjacent windows of constant size. The histograms of the grey-level value normalized between 0 and 1 in each window are taken as features. These features are very simple descriptors of the texture in the window. The windows used are of size 40x40 pixels and the descriptors of dimension 16.

As all these features are histograms, we used a χ^2 distance for histogram comparison [18]:

$$\|H_1 - H_2\|^2 = \sum_i \frac{(H_{1,i} - H_{2,i})^2}{H_{1,i} + H_{2,i}}$$

where H_i is the value of the i^{th} histogram bin.

V. Experimental results

Our system has been validated using a Sony Aibo robot. All the processing was performed off-board using MATLAB on a remote computer through the URBI interface language [19]. The images used are of relatively low quality with 208x160 pixels resolution (figure 1). Unless otherwise specified, the threshold for quality used was 0.1 and the threshold for the number of feature was 5. In the following, a localization experiment entails putting the robot in a random room and running the localization and mapping procedure.

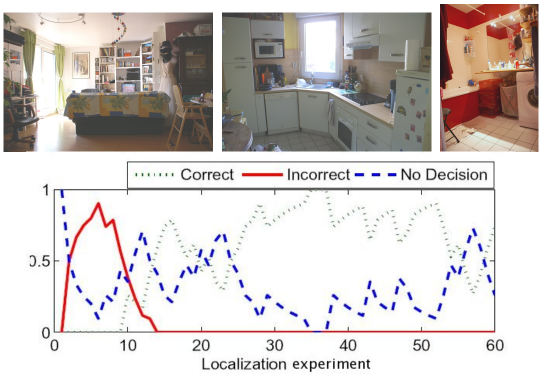

Fig. 5. Example of the evolution of recognition rates on a real robot in a 3 room apartment. Values are the average on 5 successive localization experiments. Top row show global views of the 3 rooms used.

Our system consistently shows good on-line performances for environments containing up to 4 different rooms in non prepared environments with moving people present in the rooms (figure 5). In particular, the number of erroneous localization quickly drops to zero, while the numbers of correct recognition oscillate above 50 % depending on the specific room difficulties. The number of labels given by the user needed to correctly learn a map is usually between 30 and 50, depending on the environment difficulty and on the information contained in the images. A localization experiment is performed in about 10 seconds including robot movements and processing.

	kitchen	living	bathroom	no decision
kitchen	18	0	0	2
living	0	14	0	6
bathroom	0	0	18	2

TABLE I

Confusion matrix in a 3 room apartment.

We also tested the quality of the resulting maps. Table I shows the confusion matrix obtained for 60 localization experiments using the map learned in the 3 rooms apartment (figure 5) and without learning new images. Each line shows the localization results when the robot is in a given room. There is no errors, with a global correct localization rate of 83 %, while the "no decision" rate depends on the room. The fact that "living" is less recognized shows that it has less unique features than the other rooms.

To provide more representative results, our system has been tested on a database acquired in real, non modified and populated environments : our lab, and several apartments[1]. A total of 10 different rooms have been used and images taken from ten different positions, in different robot configurations (standing, on its back, hold by the user) have been recorded in each room at different times of the day. For each position, the robot has taken 50 images by sampling head directions so that the localization algorithm can choose random head directions from the database. The database is composed of a total of 5000 images.

We evaluate our algorithm on 20 different environments built by randomly selecting 3 to 7 rooms from the database. A localization experiment of the algorithm is performed by randomly selecting a room from the environment and a position in that room. Randomly selected images from that position are then used according to our localization and mapping algorithm.

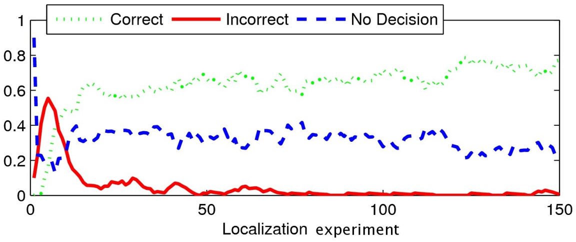

Fig. 6. Evolution of recognition rates on our database experiments. Values are the average on 5 successive localization experiments.

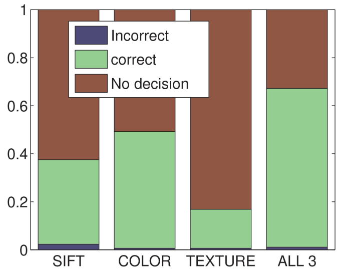

Fig. 7. Performances obtained using the different feature spaces and all the feature spaces simultaneously. Results are a mean of 20 mapping sessions in different random environments . The graph show the performance mean obtained on 50 localization experiments after 100 localization and mapping steps.

Figure 6 shows the mean performances obtained on this database. These result confirm the capacity to provide almost no erroneous localization, while giving the correct position around 70 % of the time. It also shows that the active learning scheme used allows for a stable long term simultaneous localization and map learning without the need to stop the learning stage at a given time. Figure 7 shows that the 3 feature spaces implemented in our system are effectively useful in our environments and that the combination of the 3 leads to more correct decisions.

VI. Discussion

Contrary to a lot of visual topological methods (e.g. [6], [8], [9], [20]) our method use a standard camera which is more adapted to humanoid or animal-like robotics. It is also different from other methods that use a standard camera ([11], [21]) in its capacity to deal with uninformative images that are frequent in an entertainment robotics context. In fact, by using different local characteristics of the images and several images for localization, our method does not try to recognize *an image*, but accumulate local cues about *the location*. This make our method robust to local environment modifications such as people passing by, as was shown in our evaluations.

The global performance of the system could appear quite poor when other authors [8] report up to 97% of location recognition. Note that the experimental setup is particularly difficult in our case, as the robot could really be placed anywhere in the environment, without any assumption on the robot position (standing, on its back, in the users arms) and with a camera of relatively small resolution and low quality. Moreover, as explained in section II, we chose not to use odometry or temporal coherency in localization which is usually a key factor in the result quality. In a different context, using our localization system and accumulating evidences while the robot moves would obviously improve the performances. Finally, the threshold on the vote quality were chosen to avoid erroneous localizations, thus favoring *No decision* and giving less correct localizations. Removing the threshold would lead to more correct recognitions, but also to more errors.

Our active localization strategy is currently very simple as new images needed to enhance localization quality are taken with random head positions. Several authors describe efficient active localization methods, usually based on entropy in the framework of metrical localization [22], [23]. Adapting such strategies in our case is difficult because our model is not rich enough to predict what perceptions should be when the robot moves the head (as the bag of words retain no structure). Using such informed active localization therefore entails a modification of the underlying representation which is a subject of future work.

Our method is currently limited in the size of the environment to about 7 rooms by the procedure that searches for the word that corresponds to a feature. The reason is that we use a simple linear search algorithm. This problem is well known

[1] We found it very difficult to compare our system with existing methods because of the intensive use of active information acquisition which prevents using most publicly available databases, as recording lots of images from a given position is not usual.

and a search function using a tree structure [24] should allow to increase the size of the manageable environments.

Voting method are used by others for visual localization systems (e.g. [11]). However, if we use more rooms, the method will probably be limited. More advanced classifiers such as Support Vector Machine (used in [12]) or boosting (used in [7]) will be needed. However, in their most common formulation, these algorithms do not perform incremental learning, even if solutions exist [25].

As shown in section V, the parameters for the construction of the vocabulary are quite sensitive in our approach. However, we found that if the parameters have to be tuned for each feature space, the same parameters can be used for all the environments we used for the tests. More generally, the parameters of the voting scheme influence the ratio between making errors and giving no answers. It is therefore a decision to be made depending on the final application : should we take the risk of always giving the most likely location and make some occasional errors, or should we be more conservative ?

Finally, our system is a qualitative localization system, but does not support navigation, as no structure of the environment is recorded. This follows from the choice of not using odometry, nor a metrical method for map learning. To support navigation, we plan to build an hybrid localization system [6] by integrating purely visual local metrical SLAM method such as those of Davison [5] in each room. This metrical localization would enable us to locate doors or interesting navigation points in each room and to guide the robot between rooms. Moreover, this technique would be an answer to the limitation in map size observed for standard metric SLAM algorithms [5].

VII. Conclusion and future work

The qualitative visual localization system we have designed based on image category learning achieves good localization results in realistic situations. The localization and mapping algorithm is completely incremental and only requires a very simple user interaction for map learning. Based on an active localization scheme, it is efficient in any robot position and after any user manipulation of the robot.

In future work, a more efficient informed active localization method will be developed, implying a modification of the underlying map. We also plan to integrate local metrical SLAM methods to add a navigation capacity to our system.

Acknowledgment

The author would like to thank three anonymous reviewers for useful comments that help greatly improving the paper quality.

References

[1] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Uncertainty in Artificial Intelligence*. Elsevier, 1988, pp. 435–461.

[2] D. Filliat and J. A. Meyer, "Map-based navigation in mobile robots - I. a review of localisation strategies," *Journal of Cognitive Systems Research*, vol. 4, no. 4, pp. 243–282, 2003.

[3] B. J. Kuipers and Y. T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robotics and Autonomous Systems*, vol. 8, pp. 47–63, 1991.

[4] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, "A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[5] A. Davison, Y. G. Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision," in *Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*, July 2004.

[6] A. Tapus and R. Siegwart, "Incremental robot mapping with fingerprints of places," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2005)*, 2005.

[7] O. M. Mozos, C. Stachniss, and W. Burgard, "Supervised learning of places from range data using adaboost," in *Proceedings of the International Conference on Robotics and Automation ICRA'05*, 2005.

[8] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2000)*, vol. 2. IEEE Press, 2000, pp. 1023–1029.

[9] P. E. Rybski, F. Zacharias, J.-F. Lett, O. Masoud, M. Gini, and N. Papanikolopoulos, "Using visual features to build topological maps of indoor environments," in *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, 2003, pp. 850–855.

[10] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, 1999.

[11] J. Kosecka and X. Yang, "Global localization and relative positioning based on scale-invariant features," in *Proceedings of the International Conference on Pattern Recognition*, 2004, pp. 319–322.

[12] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV04 workshop on Statistical Learning in Computer Vision*, 2004, pp. 59–74.

[13] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *International Conference on Computer Vision*, 2005.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, November 2001.

[16] A. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[17] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *International Conference on Computer Vision & Pattern Recognition*, vol. 2, June 2003, pp. 257–263.

[18] J.-J. Gonzalez-Barbosa and S. Lacroix, "Rover localization in natural environments by indexing panoramic images." in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002*, 2002, pp. 1365–1370.

[19] J. Baillie, "Urbi: A universal language for robotic control," *International journal of Humanoid Robotics*, 2004.

[20] M. Jogan and A. Leonardis, "Robust localization using an omnidirectional appearance-based subspace model of environment," *Robotics and Autonomous Systems*, vol. 45, no. 1, pp. 51–72, 2003.

[21] J. Wolf, W. Burgard, and H. Burkhardt, "Using an image retrieval system for vision-based mobile robot localization," in *In Proc. of the International Conference on Image and Video Retrieval (CIVR)*, 2002.

[22] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, 1998.

[23] J. M. Porta, B. Terwijn, and B. Krse, "Efficient entropy-based action selection for appearance-based robot localization," in *Proceedings of the International Conference on Robotics and Automation ICRA'03*, 2003, pp. 2842–2847.

[24] J. S. Beis and D. G. Lowe, "Indexing without invariants in 3d object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1000–1015, 1999.

[25] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *NIPS*, 2000, pp. 409–415.