Interactive learning of visual topological navigation

David Filliat

Interactive learning of visual topological navigation

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

***Abstract*— We present a topological navigation system that is able to visually recognize the different rooms of an apartment and guide a robot between them. Specifically tailored for small entertainment robots, the system relies on vision only and learns its navigation capabilities incrementally by interacting with a user. This continuous learning strategy makes the system particularly adaptable to environmental lighting and structure modifications. From the computer vision point of view, the system uses a purely appearance-based image representation called bag of visual words, without any metric information. This representation was adapted to the incremental context of robotics and supplemented by active perception to enhance performances. Empirical validation on real robots and on the publicly available INDECS image database are presented.**

I. Introduction

Navigation is a fundamental capacity for mobile robots and numerous solutions have been proposed, adapted to different kind of robots. In this paper, we are specifically interested in small entertainment robots of humanoid or animal shape. Vision is the best suited sensor for these platforms due to its low cost, wide availability, low power consumption and highly informative output.

Vision-based navigation systems may use either topological or metrical maps [1]. In topological maps, only places such as rooms and their relations are learned and recognized [2], whereas in metrical maps, the precise positions of environment features and of the robot are estimated (e.g. [3]). In realistic scenarios for entertainment robotics, the robot is often moved directly by the user from one place to another, can fall or be blocked in places 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 probably fail, whereas a topological approach, able to recognize the rooms and guide the robot between them is more adapted. Moreover, topological approaches may be purely appearance based, thus avoiding the need for camera calibration.

In vision-based topological approaches, the use of a panoramic camera is common (e.g. [4], [5], [6]): this kind of sensor provides 360 information about the surroundings of the robot at one time, thereby making place recognition easier for example. In a humanoid or animal-like robot context, however, the use of a standard gaze-controlled camera is more natural, even if potentially more difficult to use. The introduction of active perception strategies [7], which are a key difference between computer vision and vision applied to robotics, is a natural way to compensate these difficulties.

The learning process should also be adapted to the context, where the user is usually eager to interact with his robot and is waiting for biologically plausible behaviours of the robot. It is therefore possible to take advantage of discontinuous user supervision to incrementally and progressively learn the navigation capabilities needed by the platform, instead of relying on a separate learning phase. As an interesting consequence, the space representation used by the robot will correspond to the concepts used by humans for navigation, thereby facilitating human-robot interactions. Such incremental training is also important to adapt the robot's spatial knowledge to the evolution of the environment such as varying lighting conditions and minor structure modifications (e.g. objects that are moved).

To provide a complete topological navigation system without using metric information, the system presented here integrates two components: a qualitative localization and mapping system and a visual homing method. The localization system (previously presented in [8]) is able to incrementally learn to recognize different rooms, while the visual homing method learns to guide the robot between rooms. Visual homing is a closed-loop strategy that iterates local goal direction prediction from an image and fixed length movement in the predicted direction. In this paper, we present a new evaluation of our qualitative localization method on the publicly available INDECS image database [9], and empirical evaluation of the visual homing method on real robots.

II. Related work

Using a standard camera, the authors of [9] perform qualitative localization by training a Support Vector Machine to predict the current room. Images are characterized by global histograms and the approach is shown to be robust over time to lighting and environment evolution. The approach has been adapted to incremental learning in [10]. The method proposed in [11] is based on scale-invariant visual keypoints to localize the robot through an image database representing the environment. Localization is performed by finding the image in the database that best match the current image. Robustness to lighting modifications is obtained by using temporal coherency of localizations. The system presented in [12] is using similar information in a two stage approach to enhance localization precision. These two systems rely on an a priori database describing the environment. However, the authors of [13] and [14] use similar approaches with online

acquisition of the image database, but localizing at the image level and not segmenting the environment at a higher level such as the rooms.

All these systems perform localization in a passive way, localizing the robot for each acquired image. However, in topological navigation, the current position is not modified by rotating the robot's camera. Conversely multiple images taken by moving the camera could be used for the estimation of the current position, as done in the work reported here. Active perception exploiting this property has been used with metric localization systems (e.g. [15]) but is not common in topological systems. To our knowledge, only [16] presents such an active localization scheme that searches for informative images to localize the robot, with a method similar to the one presented in this paper.

As in vision-based topological localization approaches, panoramic vision is often used to achieve visual homing ([17], [5]). However, in [18], while originally using panoramic vision, the authors report an adaptation with a standard camera without loss of performance, but requires an estimate of the robot's absolute direction by an external mean. Using a standard camera, most authors rely on metric information: for example, the system presented in [13] uses an estimation of geometric transformation between images to guide the robot. The research field of visual servoing also provides homing methods when applied to mobile robotics: using feature tracking and local 3D reconstruction between images, the authors of [19] control a robot to reproduce a path only specified by the image sequence acquired on this path. Few approaches however are not using metric information: [20] use a qualitative approach relying on feature tracking and qualitative control and [14] rely on image matching to choose the robot direction.

Finally, most of these methods either rely on supervised learning through an initial data acquisition phase ([9], [11], [12], [18], [16], [5]) or on autonomous segmentation of the environment ([14], [19], [13]). However, in a dynamic world, this initial or autonomous training cannot be guaranteed to provide robustness to environment and lighting modifications, and the ability to update the underlying model on-line, as proposed in this paper, is crucial. Some of these systems were therefore adapted to incremental learning with user supervision ([10]), by integrating small databases of new data. Our system integrates new user labelled data at a finer scale, requiring new data only when navigation is not possible. This is more similar to the concept of Human Augmented Mapping ([21], where only range sensing was used), where the robot incrementally discovers its environment guided by a human supervisor.

The main contribution of our work is therefore the integration of a purely appearance-based approach to qualitative localization at the level of rooms and visual homing to guide the robot between the recognized rooms. This is made possible using a standard perspective camera and without using any metric information by the coupling of active perception and incremental learning with user interaction.

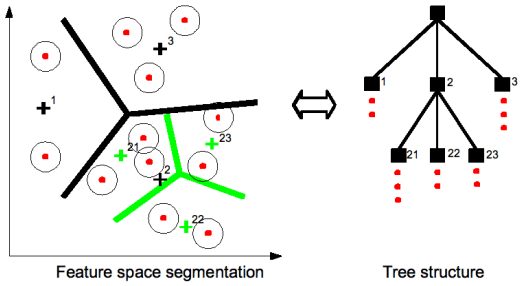

Fig. 1. Illustration of the tree structure with $k = 3$. Left: The crosses illustrate the node centres, the circles illustrate the words. Right: Only the word centres are stored in the leaves of the tree.

III. Incremental Bag of Words method for robotics

We adopted the popular "bag of visual words" approach to represent images in our system. Our contribution is an adaptation of this method in a purely incremental setup well suited for robotics, including the construction of a fast search structure for the visual words.

Bags of visual words is a popular method for image categorization [22] that relies on a representation of images as a set of unordered elementary visual features (the words) taken from a dictionary (or codebook). Using a given dictionary, a classifier is simply based on the frequencies of the words in an image, thus ignoring any global image structure. The term "bag of words" refers to document classification techniques that inspired these approaches where documents are considered as unordered sets of words. Several applications also exist for robotics (e.g. [12], [23]).

The words used in image processing are local image features such as SIFT keypoints (Scale Invariant Feature Transform) [24]. 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 vector quantization techniques such as k-means. The output of this discretization is the dictionary. Instead of building the dictionary off-line on an image database as is performed in most applications, we introduce an incremental dictionary construction ([8]) that makes it possible to start with an empty dictionary and build it as the robot discovers its surroundings. Our system therefore makes no a priori hypothesis on the type of environment it will face. The words in our system are balls of fixed radius in the feature space. Dictionary construction entails adding a new word centred on any feature that does not belong to an already existing word. The size of the balls is called the dictionary radius and influences the dictionary size, the algorithm performances and computation time (see [8]).

When using bag of words techniques with large vocabularies as is done in our system, searching for the word corresponding to a feature is a time consuming process. We therefore developed a tree dictionary structure to accelerate this operation. This structure is similar to that of [25], but built incrementally (figure 1). Each internal node of the tree has a set of k children, each defined by a centre in the feature

```
word_list = Search (node i, feature f)
  word_list = []
  if  (i is a leaf)
  | foreach word w in i
  | | if (dist(f,center(w)) < word_radius)
  | | | word_list = [word_list w]
  else
  | [s,d] = sort_children(f)
  | for j=1 to p
  | | if (d(j) < word_radius)
  | | | word_list = [word_list Search(s(j), f)]
  return word_list
```

Fig. 2. Search algorithm pseudo code. $dist(f, g)$ computes the distance between two features f and g. $word\_radius$ is the size of the words in the dictionary. $s, d = sort\_children(f)$ returns the list s of children sorted according to the distance of their frontier to feature f and the corresponding list of distances d. p is the maximum number of children to consider. See text for details.

space. Each child stores the word centres that are the closest to its centre, thus partitioning the feature space of the parent node by the Voronoi diagram of the k children centres.

The building process is fully incremental and simply begins with an empty root node. Any new word that should be added to the dictionary is directly added to the leaf node to which its centre belongs. If the number of words stored in this leaf is above a threshold n_w, the leaf is split in k children. The centres of the children leaves are defined by applying k-means to the n_w words centres. We applied this procedure in the experiments of this paper with $n_w = 500$ and $k = 10$. Although this procedure does not enforce a balanced structure to the tree[1], therefore potentially penalizing the search efficiency, experimental results show that the trees are always nearly balanced with a depth variation among branches of less than 2 and a limited impact on search speed.

As shown by Beis and Lowe [26], searching for words in these structures in high dimension (e.g. 128 for SIFT descriptors) leads to a complexity similar or even worse than that of naive linear search because a large number of nodes is examined, thus compromising any interest in the use of a tree structure. This scaling problem was solved in [26] in the case of kd-trees by the design of a fast approximate search procedure. We use a similar method, by limiting the number of children to be explored in each node to $p < k$ and by searching first in the children whose frontiers are the closest to the searched feature (Figure 2). This procedure affords a very fast search – at the cost of a low percentage of errors. For example, in the experiments reported in this paper, the search for the words corresponding to a SIFT feature in a dictionary of 15000 words with $p = 3$ took in average 1.4 ms with an error rate of around 0.6%.

This search procedure rely on the use of L2 distance for the calculation of the distance to the node frontier. However, in some cases, the use of another distance is preferred. For example, color histograms are better compared using the diffusion distance [27] we use in this paper (see below). As this distance does not stem from a dot product, rapidly calculating the distance between a feature and a node frontier is not possible. It is therefore not possible to estimate if a neighboring node has to be searched or not. For these cases, we devised another approximate search strategy that *exhaustively* explores a given number $q < k$ of children for each node, starting with the children whose *centres* are the closest to the feature. Experiments with diffusion distance and color histograms show that this procedure leads to a small search time with very few errors. In the experiments reported in this paper, the search for the words corresponding to a H histogram feature, in a 15000 words dictionary with $q = 3$ took around 2.0 ms with an error rate of 0.6%.

As shown in [8], performances can be improved by integrating several feature spaces. To this end, a dictionary is built for each feature space, and the classifiers integrate the words taken from all the dictionaries (see next section). In this paper, two feature spaces using complementary image characteristics were used:

- SIFT keypoints [24]: interest points are detected as the maximum over scale and space of the convolution by differences of Gaussian. Keypoints are described by histograms of gradient orientations around the detected point and are invariant in scale and rotation. The descriptor used are of dimension 128 and are compared using L2 distance.
- Local color histograms: The image is decomposed in a set of overlapping windows of several sizes in order to provide some scale invariance. 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 taken each 20 pixels and 20x20 pixels taken each 10 pixels. The descriptors are of dimension 16 and are compared using diffusion distance [27].

IV. System overview

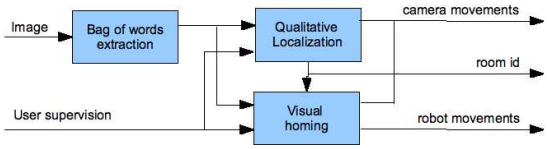

Fig. 3. Functional overview of the system.

Our navigation method uses the same bag of words image representation for qualitative localization and visual homing (figure 3). A module therefore transforms images coming from the robot's camera into their bag of words representations, incrementally building the corresponding dictionaries. These representations are used by the localization module to predict the room identity and by the visual homing module to estimate the direction from the current room (estimated by the localization module) to the room requested by the user. In these two modules, the classifiers should be trained incrementally, i.e. they should be able to process new examples and add new categories without the need to reprocess all the previous data. To achieve that, we used voting methods in which training simply entails updating word statistics, and classifying simply entails reading these statistics.

[1] as is usually required in kd-trees for example

As will be detailled in the next subsections, the localization and visual homing modules both use active perception strategies, potentially requesting new images with a different camera orientation to perform their task. These two modules also perform learning incrementally using discontinuous supervision from the user. The user can provide the room identity to the localization module at any time, while the correct goal direction is requested by the visual homing module.

A. Localization and mapping module

The map in our approach is composed of statistics associated to the visual words, i.e. the rooms in which each word has already been seen in the examples used for training.

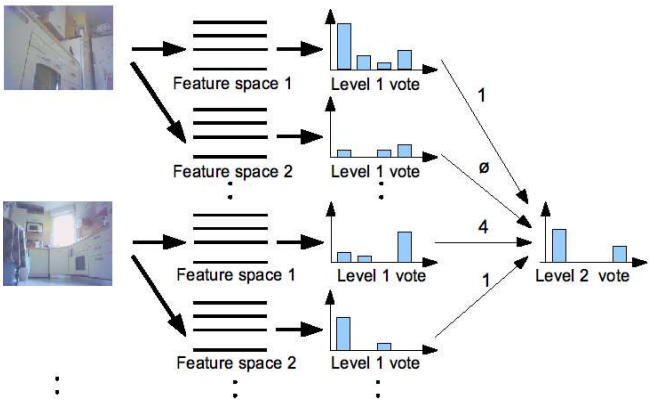

Fig. 4. Illustration of the two stage voting method used for qualitative localization.

A two stage voting method implementing the active perception procedure is used to estimate the robot position (figure 4). In a given position, a first picture is taken from the current head direction. The words found in the image vote at a first level for the rooms in which they have already been seen. Each word votes using its normalized *inverse document frequency*, giving more weight to the words that correspond to fewer locations:

$$idf = log(N/n_i)/log(N)$$

where N is the total number of rooms and n_i the number of rooms in which the word i has been seen.

A quality of the vote result is calculated as the relative difference between the maximum and the second maximum:

$$quality = \frac{v_{Winner} - v_{Second}}{\sum_j v_j}$$

where v_j is the number of votes for room j.

In order to filter out non-informative images that bring noise in the estimation, the winning room votes at the second level (with its quality) only if the quality is above a threshold, 0.1 in this paper (see [8] for an evaluation of the threshold influence).

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 (5 in all experiments). The recognized room is then the room with the highest score. The new images taken for localization are taken with a new random head direction without moving the robot's body.

The associated 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 the room label given by the user. Images that have been used for localization and new images taken from random head directions are used for learning (for a total of 10 images in the experiments reported). Learning these images entails simply memorizing that the corresponding words have been seen in the current room. The succession of localization events in different rooms, at different positions and under different lighting condition, learning when errors are committed, eventually converges to a correct representation of rooms and to stabilization of the recognition performances (see Results section).

B. Visual homing module

This module learns, for each room, several visual homing strategies that can guide the robot to the different neighboring rooms. A homing strategy makes it possible to infer the local direction to take to reach the goal from any position in the room. Goal reaching is performed by iterating predictions of the goal direction from the current camera image and movement of the robot in this direction for a fixed distance.

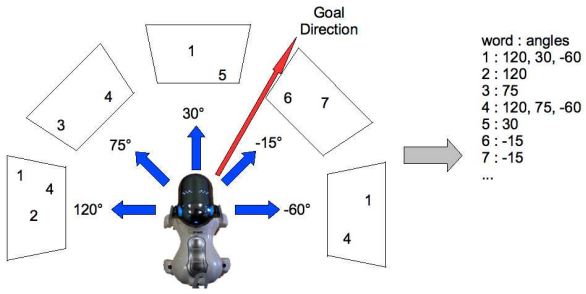

Fig. 5. Illustration of the visual homing learning procedure.

The learning procedure is triggered each time it is not possible to predict the goal direction. The procedure first asks the user for the local goal direction. Five images are then captured by moving the robot head from one side to the other. The visual words from each image are associated with the relative direction between the robot head and the goal (figure 5). A homing strategy is therefore memorized as a list of angles for each word in the dictionaries. For each word, the mean and standard deviation of the associated angles are estimated.

Predicting the goal direction from an image is performed using a voting method. The directions around the robot are discretized with a step of 20 degrees. Each word found in the image vote for the bin corresponding to its mean associated direction. Words which are found in different parts of the

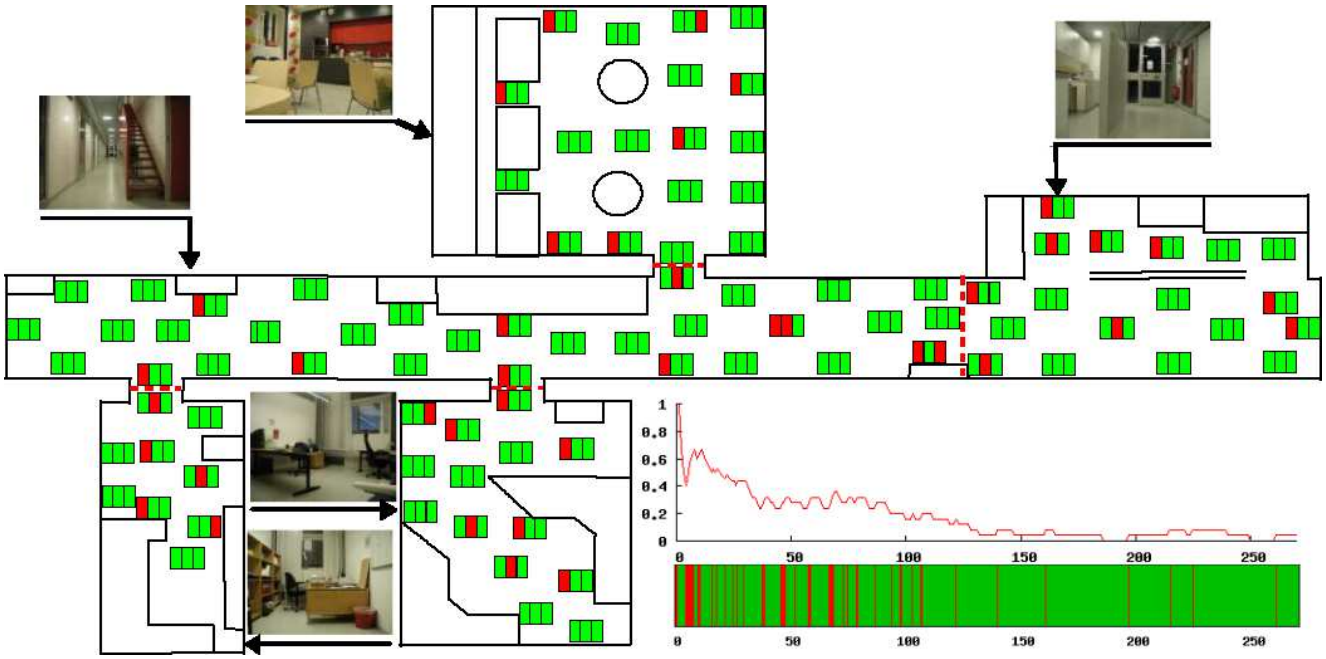

Fig. 6. Example of localization results on INDECS database. For each position, 3 rectangles show the 3 successive localization experiments performed at this point with different lighting conditions. A green rectangle corresponds to a successful localization, a red one corresponds to an error and the learning of the position for this lighting condition. The diagrams at the bottom right show the temporal succession of correct and incorrect localization results (bottom) and the evolution of the error rate on the last 25 localizations (top).

environment (e.g. words 1 and 4 in figure 5) are excluded from the vote through a threshold on the standard deviation of their associated directions. In our experiments, words with a standard deviation of more than 20 degrees were excluded. If the quality of the vote result (estimated as in the localization module) is below a threshold (0.1 in the experiments), an active perception procedure requests new images by turning the robot head 45 degrees to the left and to the right. If none of these images produce a vote with a sufficient quality, the learning procedure is performed. Otherwise, the robot is turned in the predicted goal direction, and is moved forward by a fixed distance (50cm in our experiments) before performing the procedure again.

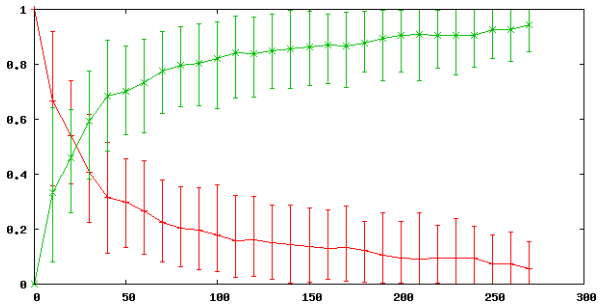

Fig. 7. Evolution of the error rate (red) and correct localization rate (green) during the last 25 localizations. The graph shows the mean values on 100 random experiments similar to the one presented in figure 6, with minimum and maximum values plotted as error bars.

V. Experimental results

A. Localization

In a previous article [8], we evaluated our localization method on a Sony Aibo robot. We present here new validation results on the publicly available INDECS database [9]. This database contains images taken at 91 different points and under three different lighting conditions (sunny, cloudy, night) in an environment made of five different rooms (figure 6). For each of the 271 positions[2] present in the database, 12 images were taken by rotating horizontally the camera of 30 degrees between images.

Evaluations were conducted by taking the 271 positions in random order, thus mimicking the incremental discovery of the environment by the robot at different positions and different time. The localization algorithm was applied for each position and learning was performed when an error was made. Figure 6 gives an example of such an evaluation sequence. In this example, 42 positions out of the 271 needed to be learned. We can see that the positions where learning was required are scattered across the whole environment, thereby naturally covering the different viewpoints in the environment. The fact that learning is most of the time performed less than one time for each point in the environment also demonstrates the robustness of our approach to lighting conditions and to minor modifications such as the presence or absence of people. The frequency of localization errors also decreases to around 4% after the first 100 localizations.

Performing these evaluations 100 times with different random position order shows that the mean number of learning

[2]We will call position a point with an associated lighting condition, thereby considering a point with different lighting as different positions where our system can try to recognize the room

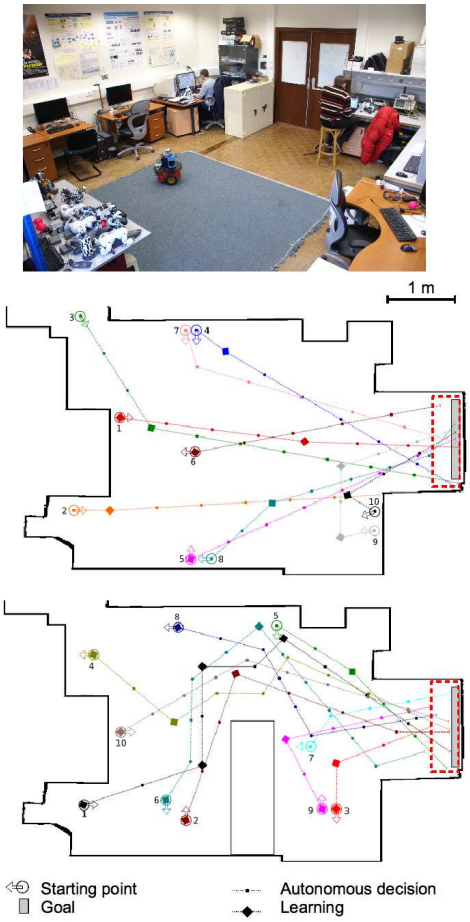

Fig. 8. Visual homing results in the main room of our laboratory, with and without a central obstacle. The goal is the dashed rectangle on the right.

event is 52.3, meaning that globally, 80% of the positions are correctly recognized. Figure 7 shows the evolution of the local rate of correct localization, showing that the localization accuracy continuously increases, reaching a level of 90% after 180 localization event. The level of 80% of correct recognition is reached after 100 localizations, with a mean of 30 trainings, corresponding to 6 positions learned in each room. Comparing these results with the ones presented in [28] (where the best classification rate was 81% on the whole dataset images with support vector machines) shows the advantage of using an active perception strategy. By automatically discarding uninformative images, and by recognizing the positions instead of all the images, our system is able to more efficiently recognize the different robot positions with a simpler machine learning algorithm. Moreover, our system only uses 20% of the images for learning instead of 33% used in [28].

B. Visual homing

Visual homing was validated in the same environment on a Sony Aibo[3] and on a MobileRobots Pioneer 3 dx. Performances were similar on the two robots. Figure 8 shows two examples of training sessions in our lab with the Pioneer 3 dx robot. The first trial was made in an open environment, with people working at their desks on the periphery. In the second, we added a large obstacle in the centre of the room to validate more complex homing strategies. In both experiments, the goal was the exit door of the room. The homing strategy was considered successful when the robot reach a 40cmx1m rectangle in front of the door. The robot was asked to reach this goal from 10 different starting positions with different orientations. In the first setup, 11 user supervisions were necessary for the robot to learn to reach the goal from all positions. In the second, more complex, setup, 15 were necessary.

In both experiments, the final homing strategy is able to guide correctly the robot to the exit door of the room. During learning, the precision obtained for the final point is low when the starting points are varied, with an error between the end-point and the rectangle centre reaching 50 cm in some experiments, but is sufficient for the robot to exit the room. The learning points are scattered in the environment, showing that the homing strategies are correctly learned as the robot is able to predict the goal direction for positions close to previous learning points. After learning, when repeating a trajectory from a given starting point, the error is smaller: 10 trials, starting from position 10 in the second environment of figure 8 leads to a mean error of 15 cm, with a maximum error of 30 cm.

VI. Discussion

Thanks to the active perception strategy, the overall performances obtained for localization are correct, using a simple appearance-based model with a perspective camera and simple learning algorithms. A limitation is the variance of the obtained results (figure 7): results can be very good (i.e. 95% of correct recognition after 100 localizations) when the user chooses correctly the localization positions, i.e. positions in open areas that rapidly covers the whole environment. But results can be bad when positions are not well chosen (i.e. less than 80% of correct recognition at the end of the experiment). However, in a realistic scenario, users have a natural tendency to guide the robot in central and open areas of the rooms, where the performances of our method are the best.

Compared to autonomous topological navigation in a similar setup ([14]), labelling places and learning homing behavior by interaction with the user has the advantage of adapting the space segmentation on-line by asking supervision to the user when the robot encounters an ambiguous viewpoint. This can be viewed as an active learning strategy, where only relevant examples are used for learning. The consequences are that less examples are required than in supervised settings [29] and that the method is stable in the long term as learning is not performed once performances are correct. A potential problem arises when a user makes errors in supervision, or tries to make the robot differentiate very similar rooms or parts of a room. In our system, this will lead to ever more requests for learning and an eventual

[3]Video available at http://cogrob.ensta.fr/indoornavigation.html

permanent confusion of the rooms. Statistics on the visual words could be analysed to warn the user in such cases.

The end point precision of the visual homing strategy is low, but is sufficient to reach a door in order to exit a room. The poor precision is linked to the fact that, contrary to more precise approaches ([13], [18], [19], [20]), only the appearance of images are used, without any metric information extraction. Our strategy also does not depend on the robot's odometry and does not require an external estimate of the robot orientation. A positive consequence is that this strategy can be used on simple platforms with low quality camera such as the Aibo robot, even with a very weak precision of movement execution. The question of when to stop a homing strategy is also important. In the experiments reported, the user stops the homing behavior when the robot reaches the door. In a more autonomous setup, localization should be attempted when homing is not possible so as to stop homing behavior if the target room is reached, or ask for user supervision otherwise.

From an implementation perspective, our system does not currently integrate planning capabilities, meaning that it can only guide the robot from one room to the neighboring ones. Integration of a complete topological map and chaining of homing strategies to go from one room to the other through a third one is the subject of future work.

VII. Conclusion

We have presented a visual topological navigation system adapted to small robots. The two modules designed to recognize rooms and guide the robot between rooms rely only on the appearance of images, without using any metric information. This simple representation is built in a fully incremental process, complemented by active perception strategies and user supervision for the learning of the navigation capabilities, making it possible to achieve efficient topological navigation on simple robots with standard perspective cameras.

Acknowledgment

The author would like to thank Jose-Luis Susa and Florian Vichot for their contribution to the implementation of the system presented here.

References

[1] 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.

[2] 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.

[3] A. J. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007.

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

[5] T. Goedeme, T. Tuytelaars, G. Vanacker, M. Nuttin, and L. V. Gool, "Omnidirectional sparse visual path following with occlusion-robust feature tracking," in *Proceedings of The sixth workshop on omnidirectional vision camera networks and non-classical cameras (OMNIVIS)*, 2005.

[6] O. Booij, Z. Zivkovic, and B. Kröse, "Sparse appearance based modeling for robot localization," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006.

[7] Y. Aloimonos, *Active perception*, Y. Aloimonos, Ed. LEA, 1993.

[8] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2007.

[9] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A discriminative approach to robust visual place recognition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS06)*, Beijing, China, October 2006.

[10] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, "Incremental learning for place recognition in dynamic environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS07)*, 2007.

[11] J. Kosecka and X. Yang, "Location recognition and global localization based on scale invariant features," in *Wokshop on Statistical Learning in Computer Vision, European Conference on Computer Vision*, 2004.

[12] J. Wang, R. Cipolla, and H. Zha, "Vision-based global localization using a visual vocabulary," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[13] F. Fraundorfer, C. Engels, and D. Nistr, "Topological mapping, localization and navigation using image collections," in *In Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2007.

[14] E. Motard, B. Raducanu, V. Cadenat, and J. Vitri, "Incremental on-line topological map learning for a visual homing application," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[15] 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.

[16] F. Schubert, T. Spexard, M. Hanheide, and S. Wachsmuth, "Active vision-based localization for robots in a home-tour scenario," in *Proceedings of International Conference on Machine Vision Applications*, 2007.

[17] A. Vardy and R. Moller, "Biologically plausible visual homing methods based on optical flow techniques," *Connection Science*, vol. 17, pp. 47–89, 2005.

[18] C. Giovannangeli, P. Gaussier, and G. Dsilles, "Robust mapless outdoor vision-based navigation," in *IEEE/RSJ International Conference on Intelligent Robots and systems*, 2006.

[19] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'07*, 2007.

[20] Z. Chen and S. T. Birchfield, "Qualitative vision-based mobile robot navigation," in *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[21] E.A.Topp and H.I.Christensen, "Topological modelling for human augmented mapping," in *In Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2006.

[22] 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.

[23] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Real-time visual loop-closure detection," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2008.

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

[25] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR06*, 2006.

[26] J. S. Beis and D. G. Lowe, "Indexing without invariants in 3d object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 1000–1005, 1999.

[27] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[28] A. Pronobis, "Indoor place recognition using support vector machines," Master's thesis, 2005.

[29] 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.