

Visual topological SLAM and global localization

Adrien Angeli, Stéphane Doncieux,
Jean-Arcady Meyer
Université Pierre et Marie Curie - Paris 6
FRE 2507, ISIR, 4 place Jussieu, F-75005
Paris, France.

firstname.lastname@isir.fr

David Filliat
UEI - ENSTA
32, bvd Victor, F-75015 Paris, France.
david.filliat@ensta.fr

Abstract—Visual localization and mapping for mobile robots has been achieved with a large variety of methods. Among them, topological navigation using vision has the advantage of offering a scalable representation, and of relying on a common and affordable sensor. In previous work, we developed such an incremental and real-time topological mapping and localization solution, without using any metrical information, and by relying on a Bayesian visual loop-closure detection algorithm. In this paper, we propose an extension of this work by integrating metrical information from robot odometry in the topological map, so as to obtain a globally consistent environment model. Also, we demonstrate the performance of our system on the global localization task, where the robot has to determine its position in a map acquired beforehand.

I. INTRODUCTION

Over the last years, vision in robotics has become more and more important, due to the remarkable characteristics of the vision systems available at low costs. The small size, low weight and low energy requirements of a simple camera make it an integrated sensor that can be easily embedded on most mobile robots, while vision provides a rich qualitative description of the environment that is suitable for robotics applications like place recognition ([1], [2], [3]). Moreover, vision can be employed for the extraction of metrical information about the environment, as in certain SLAM solutions ([4]).

SLAM (Simultaneous Localization And Mapping, [5]) is the process of localizing a mobile robot while concurrently building a map of the environment. Historically, the field of SLAM has been divided into metrical and topological approaches. In the former case, the environment is represented using a metrical map where the robot can be localized in a continuous manner. In the latter family of approaches, the environment model is a graph of discrete locations: the nodes of this topological map identify distinct places in the environment, while edges link them according to their similarity or distance. Number of approaches have attempted to capitalize on the advantages of the two representations. For instance, metrical maps can be embedded in graphs of higher level to enhance scalability ([6]). Also, other graph-based solutions can be used to infer a precise metrical position for the robot, while still allowing for large scale mapping ([7]).

In previous work [8], we have demonstrated how a vision-based loop-closure detection method (i.e. BayesianLCD, [1]) could be turned into a reliable incremental and real-time

topological SLAM solution, using appearance information from a single monocular camera only. One limitation of this work was the lack of metrical information which lead to the impossibility to use the map to guide a robot. We have enhanced this mapping solution with the addition of such information, taking advantage of the odometry measurements provided by a mobile robot. Also, we have adapted this framework to the context of global localization, as this problem can be considered as a particular case of loop-closure detection where the robot is assumed to be in known terrain. We demonstrate the quality of our approach using image sequences acquired with a single monocular camera on a Pioneer 3 DX mobile robot, in indoor and urban environments, and under strong perceptual aliasing conditions (i.e. when several distinct places look similar).

II. RELATED WORK

Several approaches have been designed to add metrical information in a visual topological map. A first solution is to match images coming from neighbouring nodes to estimate the robot displacement between these nodes ([7], [9], [10]) using *visual odometry* [11] and to store this displacement in the edges. Another similar method is to rely on *visual servoing*, which makes it possible to directly guide the robot toward the position of a neighbouring node, without explicitly computing the corresponding relative positions [12]. Other authors ([13], [14], [15]) use the odometry measurements provided by a mobile robot during the movement between nodes. Depending on the scenario, this last approach may be more relevant than the aforementioned vision-based techniques, as it can still provide an estimation of the robot's position in situations where vision is no longer reliable (e.g. during temporary sensor occlusion, or in featureless scenes such as those caused by a dark spot in the environment). It also has the advantage of being computationally simpler, as it does not require any image processing.

The metrical information that relates neighbouring nodes may be used directly to guide the robot between nodes ([10], [12]). However, it is also possible to capitalize on this information to build a globally consistent map of the environment. This can be achieved by using a relaxation algorithm that relies on the relative information between the nodes to estimate a global position for them ([13], [14], [15]). Similar approaches were also applied to build metrical

maps of the environment ([9], [16]), following the seminal work of [17]. In particular, the relaxation method proposed in [9] allows to rapidly converge to low average error when considering 3D-6DoF camera poses.

The topological global localization problem consists in determining the node corresponding to the actual robot's position, without any a priori information on this position. Several vision-based techniques ([10], [18], [19]) consider this problem in a simple image-to-nodes matching scheme, where the location of the current image is determined as the location of the most likely node in the map. To this end, a similarity measure between an image and a node is defined (i.e. this generally entails counting the number of correspondences between them), while some authors ([10], [18]) also rely on a final multiple-view geometry validation step in order to confirm the retrieved location. In the aforementioned approaches, global localization is achieved in a *maximum likelihood* (ML) scheme, which may suffer several limitations and lead to transient errors in the presence of perceptual aliasing.

In order to circumvent these limitations, Bayesian filtering methods can be employed, leading to a *maximum a posteriori* (MAP) scheme that ensures the time coherency of the estimation (i.e. information from past estimates is fused with current ones). The authors of [20] and [21] use MAP frameworks to estimate the probability of the location of the current image. Before the first localization attempt, this probability is uniformly distributed over all the nodes of the map. Then, an iterative *predict-update* procedure helps refining the estimation of this probability, as the robot moves and acquires new images. To this end, a time evolution model predicts the probability distribution at time t , given this distribution one step before, while an observation model is used to update the probability of each node, by computing the likelihood of the current image given the description of this node. This update step relies on an image-to-nodes matching scheme that is similar to those used in ML approaches. At each iteration of the filtering process, the location of the current image can be determined confidently when the probability of a particular node is high.

Finally, learning techniques can also be employed to address the visual topological global localization problem, as shown in [22] and [3] where a monocular camera is used to recognize the different rooms of an indoor environment.

III. TOPOLOGICAL SLAM

The environment model used in this paper is an enhancement of the model described in [8]. It consists in a topological map of the environment (i.e. the graph of the locations linked in order of traversal) that is constructed from image sequences, and where each node is characterized using the *bags of visual words* paradigm.

A. Model overview

Bags of visual words is a popular method for image categorization [23] that relies on a representation of images as a set of unordered elementary visual features (the *words*)

taken from a *dictionary* (or codebook). Over the last years, this method has been successfully adapted to several robotics applications (e.g. [2], [10], [19]).

An example of the visual features typically used for image characterization in the bags of visual words scheme is the *Scale Invariant Feature Transform* (SIFT, [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 like *k-means*. The output of this discretization is the dictionary. Instead of building the dictionary off-line on an image database, as performed in most applications ([2], [10], [19], [23]), we rather rely on an incremental dictionary construction mechanism [3]. This makes it possible to start with an empty structure that is filled as the robot discovers its surroundings: our system therefore makes no a priori hypotheses on the type of environment it will face.

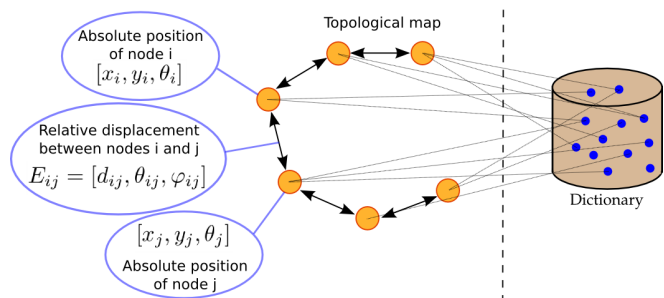


Fig. 1. Illustration of the environment model. The visual words of the dictionary (right part of the figure) are used to describe the locations of the topological map (left part of the figure). The integration of metrical information makes it possible to compute an absolute position for each node, using the relative displacements between them (see figure 2 for details about the metrical information added here).

The input information used to build the dictionary of the environment model described in this paper is the SIFT descriptor ([24]): interest points are detected as maxima over scale and space in differences of Gaussians convolutions. The keypoints are memorized as histograms of gradient orientations around the detected point at the detected scale. The corresponding descriptors are of dimension 128 and are compared using L2 distance.

In [8], we have shown how the model can be learned on-line, in real-time and without any a priori information about the environment. To this end, when a new image is acquired, our Bayesian loop-closure detection algorithm (i.e. BayesianLCD, [1]) is used to determine the robot's location, so as to update the topological map. In case of successful detection, the image is considered as pertaining to the loop-closing location. Otherwise, it is used to define a new location. An edge is added to the map between the current node and the previously recognized one. Then, the visual dictionary is updated, by adding all the features of the current image that did not match existing words.

B. Image selection strategy

In our previous work, images taken from a hand held camera were processed at 1Hz for map building. In order

to avoid loop-closure detections due to the resemblance between consecutively acquired images, frames exhibiting too much similarities with the last considered image were discarded. In this paper, as the camera is mounted on a mobile robot, it is possible to use information from odometry measurements as an additional constraint to decide which image to process and, as a consequence, how the nodes of the map should be distributed. Hence, we now also impose that the robot must have moved a given distance or rotated a given angle (50cm and $\frac{\pi}{6}$ radians in the reported experiments) for the image to be considered.

C. Embedding metrical information in the map

Each node of the graph has an associated 2D position and orientation $[x_i, y_i, \theta_i]$ initialized to the robot odometry position when the node is created. A variance v_i is also associated to each node and is initialized with the variance of the previous node in the map plus 2% of the distance travelled by the robot since this previous node. When a node is added or recognized in the map, a new edge is created to link this node with the previous one (i.e. the node where the robot was last located). The relative metrical position of the two nodes obtained through the robot odometry measurements is memorized in this link (see figure 2):

$$E_{ij} = [d_{ij}, \theta_{ij}, \varphi_{ij}]$$

a variance v_{ij} is also associated to the edge. In this paper, it is taken as 2% of the edge length d_{ij} . Note that uncertainty is modelled very simply here by a single value for both position coordinates and orientation, which provides surprisingly good results in our experiments as odometry has a reasonable precision, notably on orientation. However, for larger scale experiments, a more precise model (e.g. the one presented in [25]) would be required.

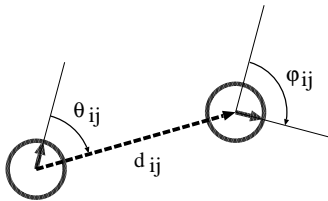


Fig. 2. Odometric information stored in the edges of the topological map.

After a loop-closure detection, the robot is assumed to have returned exactly at the position of the previous passing (i.e. the position of the loop-closing node), thus relative position and uncertainty for the loop closing edge is taken from odometry information like for any other edge. This is a reasonable assumption given that loop-closures are only detected between close monocular views of a given place, thereby exhibiting only small variations between the corresponding positions and orientations. A solution relying on the relative image position given by the multiple-view geometry algorithm (see section III-D) would be hardly feasible here due to scale ambiguity.

As a consequence of the cumulative noise of odometry, the graph is not coherent after loop closing. Thus, a relaxation algorithm is employed to estimate the position of each node that best satisfies the constraints imposed by the relative odometric information. The algorithm we used for relaxation is simple, as the maps we are building have a relatively small number of nodes (at most few hundreds in the experiments reported hereafter), and as the nodes only have 3 degrees of freedom. We use the iterative algorithm described in [13], to which we added the estimation of the orientation for each node.

An iteration of the algorithm is made of three steps applied to each node i of the map:

- Step 1 – Estimate the position of node i from each neighbouring node j :

$$(x'_i)_j = x_j + d_{ji} \cos(\theta_{ji} + \theta_j) \quad (1)$$

$$(y'_i)_j = y_j + d_{ji} \sin(\theta_{ji} + \theta_j) \quad (2)$$

$$(\theta'_i)_j = \theta_j + \varphi_{ji} \quad (3)$$

and estimate variance of node i from node j :

$$(v'_i)_j = v_j + v_{ji}$$

- Step 2 – Estimate the variance of node i using harmonic mean of the estimates from the neighbours:

$$v_i = \frac{n_i}{\sum_j \frac{1}{(v'_i)_j}} \quad (4)$$

$$(5)$$

where n_i is the number of neighbours of node i .

- Step 3 – Estimate the position of the node as the mean of the estimates from its neighbours:

$$x_i = \frac{1}{n_i} \sum_j \frac{(x'_i)_j v_i}{(v'_i)_j} \quad (6)$$

$$y_i = \frac{1}{n_i} \sum_j \frac{(y'_i)_j v_i}{(v'_i)_j} \quad (7)$$

$$\theta_i = \arctan \left(\frac{\sum_j \frac{\sin((\theta'_i)_j)}{(v'_i)_j}}{\sum_j \frac{\cos((\theta'_i)_j)}{(v'_i)_j}} \right) \quad (8)$$

$$(9)$$

These three steps are repeated until the total change in the nodes coordinates falls under a given threshold, or a maximum number of iterations is reached (20 in our experiments). The first node of the map is considered as the reference frame: its position is fixed at $[0, 0, 0]$ and its variance is fixed at a small value. This algorithm was proven to converge [13], as it corresponds to the minimization of a quadratic energy function of a spring network equivalent to the topological map. It is also fast enough to be executed during the time separating two image processing during map construction.

D. Topological global localization

In this section, we propose to derive the probabilistic framework employed for loop-closure detection in BayesianLCD for the task of global localization. The main difference is that in this new context, we wish to recover the location of the robot in an environment model obtained beforehand, and it is assumed that each acquired image is taken from an already visited place. As a consequence, the “novelty” event that is used in loop-closure detection to take new locations into account is not required. In our previous work [1], this novelty event was managed by the addition of a virtual location in the model which was updated at each new image acquisition, in order to represent a potential new location to which this image could pertain. In the task considered here, this virtual location mechanism is no longer necessary. Also, when performing global localization, the environment model is held fixed, and so neither the map nor the visual dictionary should be updated after the processing of an image.

The probability that the current image comes from an already visited location can be recursively evaluated using a discrete Bayes filter, as follows:

$$p(S_t|z_t, M) = \eta p(z_t|S_t, M) \sum_{j=0}^n p(S_t|S_{t-1} = j, M) p(S_{t-1} = j|M) \quad (10)$$

where η is a normalization term, $M = \{N_0, \dots, N_n\}$ is the set of nodes forming the topological map, and z_t is the set of visual words found in current image I_t . $S_t = i$ is the event that I_t comes from the location corresponding to N_i . Computing the *full posterior* $p(S_t|z_t, M)$ according to equation 10 using a discrete Bayes filter then makes it possible to find the node N_j whose characterization is similar enough to I_t to consider that I_t comes from N_j .

As usual in classical Bayesian filtering problems, the estimation of the full posterior requires a time evolution model $p(S_t|S_{t-1} = j, M)$, and an observation model $p(z_t|S_t, M)$.

The time evolution model makes it possible to predict the probability distribution at time t , given this distribution at time $t-1$, and according to possible displacements in the map between $t-1$ and t . Here, it is simply represented as a sum of Gaussians over the nodes neighbouring a given location (see figure 3): according to the image selection strategy given in section III, it is assumed that the robot has moved between two images, implying that it is more likely to be situated in a different node rather than in its last location.

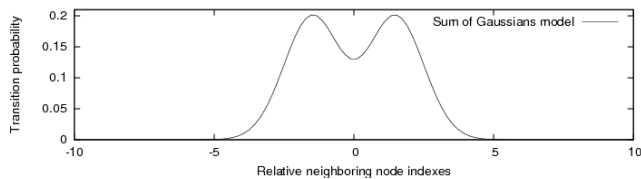


Fig. 3. Sum of Gaussians for the time evolution model: the sum of Gaussians model gives more emphasis to neighbouring states than to centre one, making it adapted to a non-stationary system.

The observation model computes the likelihood of the currently observed words z_t given the descriptions of all the locations of the environment model. In other words, it evaluates the relevance of each position hypothesis $S_t = i$, based on the observed similarities between I_t and N_i . To this end, each word of the current image votes for all the locations in which it has been seen, using a score derived from the tf-idf [26] coefficient (i.e. the product of the frequency of a word in a location by the inverse frequency of the locations containing this word). Once all the words of the current image have voted, the more likely locations are those receiving the more important number of votes, and their likelihood score is obtained from these votes.

Finally, when the sum of the probabilities taken over neighbouring locations is above a threshold (i.e. 0.8 in the following experiments), a multiple-view geometry algorithm [11] is employed to verify that a consistent camera transformation can be found between the current image and the retrieved location. This final validation step makes it possible to discard false alarms (i.e. locations that look similar to the current image but that do not share a consistent structure with it). More details regarding the observation model and this ultimate verification procedure can be found in [8].

IV. EXPERIMENTAL RESULTS

A. Mapping



Fig. 4. Images from the sequence used in the reported experiment. Note that some images are almost featureless (bottom row, centre).

Experiments were conducted using a Pioneer 3 DX mobile robot from MobileRobots Inc. equipped with an on-board camera providing images of size 320x240 pixels (automatic exposure control). The robot’s trajectory started with a small loop around a room, before taking one longer loop in a corridor. Along this trajectory, 209 images were selected (through the appearance and position threshold described in section III) and processed for mapping (see figure 4).

During this experiment, 7 loop-closures were correctly detected, and in spite of strong perceptual aliasing in the environment, no false detections were made (i.e. when a loop-closure is detected whereas none occurred). The final map contains a total of 202 nodes. The robot took 5m10s to complete the whole trajectory, while the total computation time was 2m58s: all images were thus processed in the required time frame. Figure 5 shows that the relaxation algorithm effectively compensated the odometry drift and map

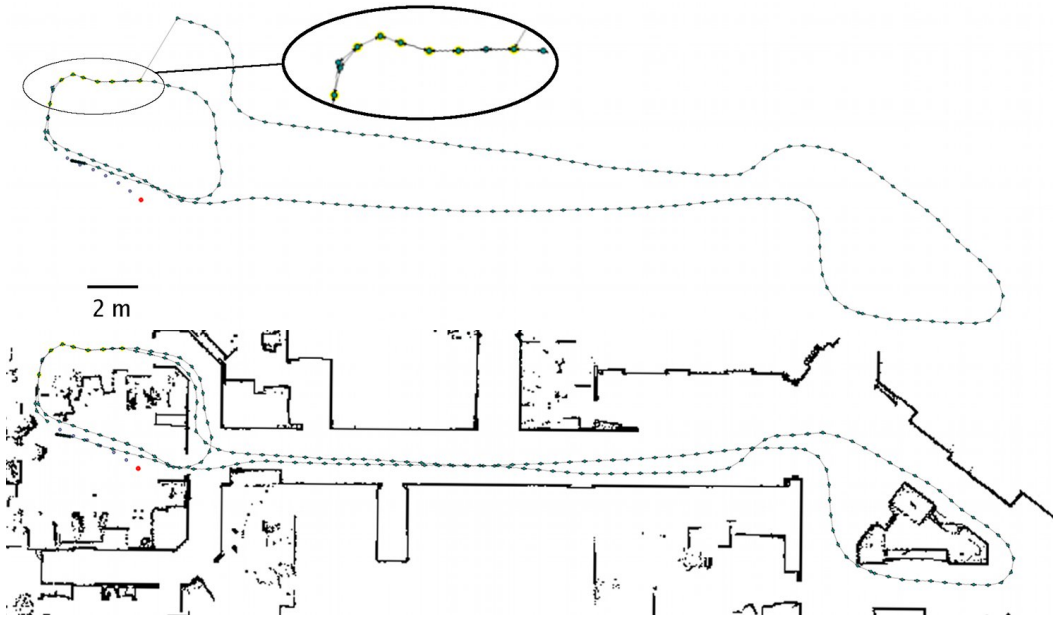


Fig. 5. The map constructed during the reported experiment without relaxation (top) and with relaxation overlaid on a metrical map of the environment (bottom). The yellow circled nodes indicate nodes where loop-closures were detected. The red dot indicate the final robot position estimated with the robot odometry from the last node of the map.

inconsistencies. As a consequence, the resulting topological structure is coherent with a metrical map constructed using a traditional laser range-finder based SLAM algorithm.

B. Topological global localization

Global localization has been performed in both indoor and outdoor image sequences, under strong perceptual aliasing conditions. To learn the environment model, a first passing is done in the environment, visiting all the places it contains once. After that, images from a second passing in those places are randomly selected to attempt global localization. Each time such a new random image is selected, the probability of the position of the robot is uniformly distributed over all the nodes of the map. Then, our discrete Bayes filter (see section III-D) is employed to refine this probability with the acquisition of the following consecutive images in the sequence, according to the image selection strategy given in section III, until a correct location is found (i.e. when the corresponding probability is higher than 0.8, and the multiple-view geometry validation step is satisfied). The number of images required before recovering a correct location is the *number of trials*. After that, following consecutive images not discarded by the image selection strategy are still being processed, as long as their locations are also correctly determined: the number of successfully tracked images is the *number of trackings*. Once tracking is lost, a new random image is picked, and global localization is attempted again.

The experimental results for the two aforementioned image sequences are presented in table I, which gives the mean number of trials before success (“#IMG-Loc”) and the mean number of successful trackings (“#Trackings”) over 100 global localization attempts (corresponding standard deviation values are given in brackets). Also, table I gives the

mean processing time per image, the total number of images in each sequence (“#IMG”), and the number of images used to learn the environment model (“#L-IMG”).

TABLE I
GLOBAL LOCALIZATION PERFORMANCES

Sequence	#IMG	#L-IMG	#IMG-Loc	#Trackings	CPU time/IMG
Indoor	327	190	5 (4)	5 (3)	115ms
Outdoor	531	230	2 (1)	12 (10)	644ms

Table I shows that the mean number of trials is small in both sequences, and most notably in the outdoor one: this is due to the good reliability of the SIFT features in the outdoor scenes. As a consequence, the number of successful trackings is also higher in this case. Tracking usually fails in situations such as sudden rotation of the camera around the vertical axis (e.g. when turning around corners in the indoor environment), or when the scene is partially obstructed (e.g. due to the presence of pedestrians and cars in the outdoor sequence). In both cases, the output of the Bayes filter usually continues to correctly detect the loop closure, but the lack of feature correspondences between previous and actual views cause the multiple-view validation step to fail, thus provoking the rejection of the corresponding hypothesis.

It is important to notice the high standard deviation values for the indoor sequence. The reason for this is the higher level of perceptual aliasing, but also the characteristics of this environment (i.e. medium sized corridors, with curved shape and suddenly appearing corners) that make it difficult to rapidly recognize a place and track the following images confidently. Finally, processing outdoor images takes longer due to the more important number of features they contain.

V. DISCUSSION

We showed the capacity of our system to build consistent visual topological maps in real-time using a simple yet efficient relaxation algorithm to integrate odometry information. When compared to graph-based metrical SLAM solutions like [7], our system estimates metrical information with less precision (notably due to the simplistic odometry error model), but offers very robust data association that makes global localization, mapping and loop-closure detection possible in the same unified framework. Data association is performed here at the location level, relying on appearance information only by considering the image as a whole, thereby offering robustness in challenging environment subject to strong perceptual aliasing or in large featureless areas. A limitation of our approach is however that information on relative position of nodes coming from vision is very sparse as it is only obtained from detected loop-closure events, thus relying on a reasonably precise odometry in between.

Localization in our model is performed by a loop-closure detection algorithm, relying on a Bayes filter to estimate the probability that an image comes from a known location: this makes it possible to prevent temporary detection errors. The probability propagation in this filter is based on the neighbouring nodes in time, giving more importance to the nodes that were detected just before and after each node (see section III-D). In the metrical extension presented here, it would be interesting to take relative position of nodes into account for this propagation, along with a probabilistic model of the robot odometry. Such a modification would probably enhance the responsiveness of loop-closure detections, as propagation would be made in the direction of the robot's movement, instead of in direction of all the neighbouring nodes. This would hence make it possible to concentrate the probability mass more efficiently at each prediction, directing it toward the robot's next presumed location.

VI. CONCLUSION

We have presented an enhancement to our previous work on visual topological SLAM by integrating odometric information from a mobile robot to obtain globally consistent maps, and by adapting the framework to achieve global localization. In future work, we plan to use metrical information for more relevant bayesian filtering. Also, it would be interesting to compare the precision of the solution employed here with a more generic setup relying on visual odometry instead of wheel encoded odometry.

ACKNOWLEDGEMENT

The authors would like to thank Nicolas Beaufort for his contribution to the implementation of this work.

REFERENCES

- [1] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, vol. 24, pp. 1027–1037, October 2008.
- [2] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, pp. 647–665, 2008.

- [3] D. Filliat, "Interactive learning of visual topological navigation," in *Proceedings of the 2008 IEEE International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008.
- [4] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1052–1067, June 2007.
- [5] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i," *IEEE Robotics and Automation Magazine*, vol. 13, no. 1, pp. 99–110, 2006.
- [6] E. Eade and T. Drummond, "Monocular slam as a graph of coalesced observations," in *International Conference on Computer Vision*, 2007.
- [7] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *IEEE Transaction on Robotics*, vol. 24, no. 5, pp. 1066–1077, 2008.
- [8] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Incremental vision-based topological slam," in *IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS2008)*, 2008.
- [9] B. Steder, G. Grisetti, S. Grzonka, C. Stachniss, A. Rottmann, and W. Burgard, "Learning maps in 3d using attitude and noisy vision sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.
- [10] F. Fraundorfer, C. Engels, and D. Nistér, "Topological mapping, localization and navigation using image collections," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.
- [11] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, June 2004.
- [12] 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.
- [13] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3841–3846, 2000.
- [14] D. Filliat and J. A. Meyer, "Global localization and topological map learning for robot navigation," in *From Animals to Animals 7. The Seventh International Conference on simulation of adaptive behavior (SAB02)*, 2002.
- [15] P. Rybski, F. Zacharias, J. Lett, O. Masoud, M. Gini, and N. Papanikolopoulos, "Using visual features to build topological maps of indoor environments," in *IEEE International Conference on Robotics and Automation*, 2003.
- [16] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *IEEE Transactions on Robotics and Automation*, vol. 21, no. 2, pp. 196–207, 2005.
- [17] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.
- [18] O. Booij, B. Terwijn, Z. Zivkovic, and B. Kröse, "Navigation using an appearance based topological map," in *IEEE International Conference on Robotics and Automation*, 2007.
- [19] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-fine vision-based localization by indexing scale-invariant features," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 36, pp. 413–422, April 2006.
- [20] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. V. Gool, "Omnidirectional vision based topological navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 219–236, 2007.
- [21] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Image-based monte-carlo localisation with omnidirectional images," *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 17–30, 2004.
- [22] A. Pronobis and B. Caputo, "Confidence-based cue integration for visual place recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.
- [23] 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*, pp. 59–74, 2004.
- [24] D. Lowe, "Distinctive image feature from scale-invariant keypoint," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [25] T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," *Autonomous Robots*, vol. 12, no. 3, pp. 287–300, 2002.
- [26] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*, 2003.