Stephane Bazeille, Emmanuel Battesti, and David Filliat

A Light Visual Mapping and Navigation Framework for Low Cost Robots

Abstract: We address the problems of localization, mapping and guidance for robots with limited computational resources by combining vision with the metrical information given by the robot odometry. We propose in this paper a novel light and robust topo-metric SLAM framework using appearance based visual loop-closure detection enhanced with the odometry. The main advantage of this combination is that the odometry makes the loop-closure detection more accurate and reactive, while the loop-closure detection enables the long term use of odometry for guidance by correcting the drift. The guidance approach is based on qualitative localization using vision and odometry and is robust to visual sensor occlusions or changes in the scene. The resulting framework is incremental, real-time and based on cheap sensors provided on many robots (a camera and odometry encoders). This approach is moreover particularly well suited for low power robots as it is not dependent on the image processing frequency and latency and so, it can be applied using remote processing. The algorithm has been validated on a Pioneer P3DX mobile robot in indoor environments and its robustness is demonstrated experimentally for a large range of odometry noise levels.

Keywords: Visual loop-closure detection, topological SLAM, path following.

Stephane Bazeille: Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova (Italy) stephane.bazeille@gmail.com
Emmanuel Battesti, David Filliat: ENSTA ParisTech - INRIA FLOWERS team, 828 boulevard des Marechaux, 91762 Palaiseau Cedex (France) firstname.lastname@ensta-paristech.fr

1 Introduction

In order to navigate in an unknown environment, a robot requires the ability to build a map and to localize itself using a process named Simultaneous Localization And Mapping (SLAM) [42]. The field of SLAM can be broadly divided into topological and metrical approaches. The topological approach models the environment as a graph of discrete locations and often leads to simple solutions [19],[8]. It is usually light and suitable for many kinds of environments and for human interaction. Building the map requires precise

sensor data association to detect when the robot comes back to a previously visited place (a process called loop-closure detection [2],[12]). Its main drawback is the lack of geometric information about the robot surroundings that reduces path planning to searching a path in the corresponding graph. On the contrary, metrical maps are explicitly based on measured distances and positions (e.g. [42],[13]). The representation of the environment is geometric and clearly corresponds to the real world. With these maps, localization can be done continuously and planned navigation is easier and more precise. The major problem is to ensure geometrical consistency between position and perception which is computationally expensive and makes the map more difficult to build.

Number of approaches have attempted to capitalize on the advantages of the two representations. For instance, local metrical maps can be embedded into graphs to enhance scalability [17]. Other graph-based solutions can be used to infer a precise metrical position of the robot, while still allowing for large scale mapping [29]. In this paper, we propose such an approach that builds topo-metric maps of the environment using a camera and robot odometry (Fig. 1). This approach keeps the simplicity of topological approaches by using only simple and fast appearance based image processing, while making it possible to navigate more robustly in the map thanks to the metric information provided by the odometry.

Vision sensors are often used for topological mapping. They provide indeed many advantages such as small size and price, light weight and low energy consumption, and above all, a rich environmental information that is usable as the only information about the environment. Vision sensors could therefore make it possible to perform navigation on small robots, but the embedded processing power limits the algorithmic complexity of the solutions. A possibility to solve this problem is to execute part of the processing on remote computers, but the approach should then deal with communication bandwidth and latency. Contrary to visual odometry (e.g. [33]) or metric mapping approach (e.g. [13]) that require high frame-rate image processing and cannot deal with remote processing the topo-metric approach proposed in this paper is cheap, light and as well suitable for remote computation and communication latencies as robot odometry allows to guide the robot between availability of the visual information.

Contributions: This paper merges and presents advances on our previously published work on topo-metric mapping using vision and odometry for loop-closure detection [6] and path following navigation [5]. The main contributions of this paper are i) the introduction of improvements in the Bayesian filter to increase the precision in loop-closure detection in order to allow reliable path

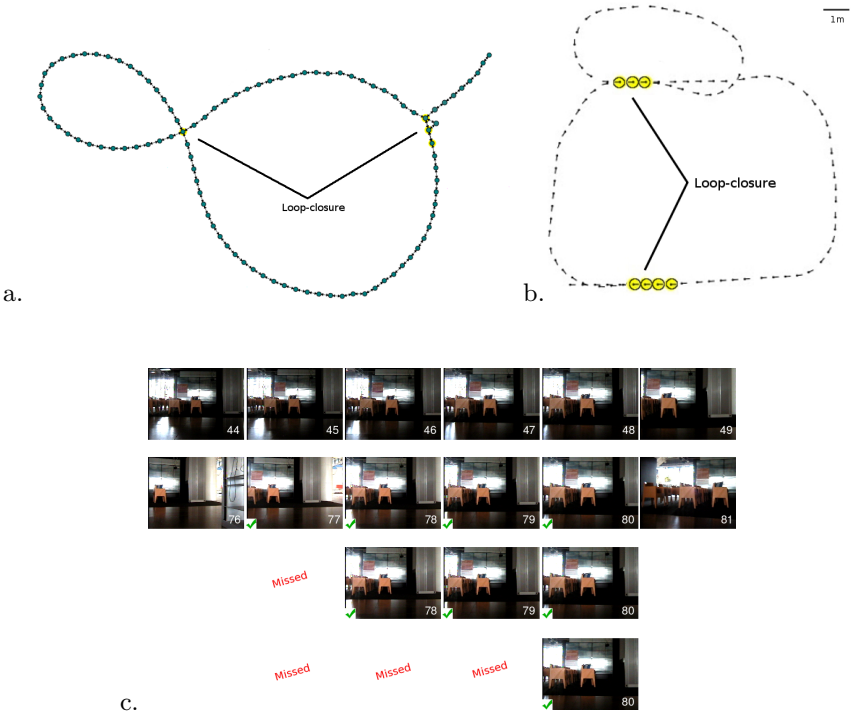


Fig. 1. Museum Sequence: Improvement of the topological map by adding metric information a. the topological map built using [2] b. the topo-metrical map built with the approach presented in this paper. Yellow circles correspond to the loop-closure locations. c. Illustration of loop-closure detection in these two maps. The first line shows images taken during the first pass with the robot, the second line shows the second pass with the ground truth loop-closure detections checked in green, the third line shows the loop-closure detected with the proposed approach and the last line shows the loop-closure detected with [2] which is not using the odometry.

following navigation, ii) the evaluation of the mapping algorithm on large maps, iii) the thorough evaluation of the path following behavior depending on robot odometry noise level, occlusions and changes in the scene, iv) a light and fast implementation of our loop-closure framework which can support remote computation.

Content: In Section 2, we present a review of related work on topological mapping and navigation. In Section 3, we recall our previous work on loop-closure detection and highlight its limitation for topo-metric mapping and navigation. In Section 4, we detail our visual loop-closure detection framework enhanced with odometry, the mapping and the navigation methods. In Section

5, we evaluate our system performances. Finally in section 6 we discuss the advantages and limitations of our solution.

2 Related work

Among all the SLAM approaches, we are more particularly interested in vision-based topological SLAM. Many approaches to the topological SLAM problem are based on appearance and rely on omni-directional vision [8],[23],[43],[44]. A similarity distance between images is defined to set the edges of the map, with similar images considered as originating from the same place and thus as corresponding to the same node. These approaches provide an efficient segmentation of the environment, since omni-directional images enable to recognize a place independently of the robot orientation. Others approaches ([2],[12],[30],[28],[32]) use perspective cameras which impose more constraints on location recognition, but can be used efficiently as there are some constraints on the robot path such as road or corridors.

During mapping, the problem of recognizing if a new image taken by the robot corresponds to a new place or an already existing place is called loop-closure detection. Several efficient approaches for this problem have been proposed recently [2],[12],[28],[22] many of them relying on the bag of visual words approach [40] that encodes images as an unordered set of local features. It has to be noted that among all these methods, a large part needs camera calibration, however few methods exists based on uncalibrated cameras [37],[21]. Our method falls in this last category as only the knowledge of the camera field of view is required.

A limitation of topological maps for robot guidance is the lack of information about obstacles, and about free space around the robot. This is why navigation methods using topological maps have been limited to following previously learned paths. A first method to follow such a path is visual servoing which uses the feedback information extracted from images to control the motion of the robot [26]. Those methods generally require camera calibration (homography, fundamental matrix, Jacobian, removal of lens distortion: e.g., [9],[38]). Also some approaches make assumptions on the environment (artificial landmarks, vertical straight lines, parallel walls) or sometimes need more than one camera or specific cameras (omni-directional for example) [15],[11]. Calibration free methods without any environment knowledge have been developed [10], they are based on image features tracking, and used qualitative comparisons of images to control the motion. Such methods are interesting in

that they are purely visual, but they still require heavy computations and are highly dependent of image data quality. Tracking errors, temporary absence of visual information, or changes in the scene lead quickly to system failures.

Another approach to guide robots from a topological maps is to complement the map with metrical information so that the robot metrical position can be estimated and used for guidance. This approach leads to topo-metric maps [7],[6]. A common approach for this is the use of visual odometry. This approach is appealing because it uses the same sensor and it processes image sequences taken between nodes to estimate the robot displacement [19],[33],[29],[39]. However, as for visual servoing, this approach is computationally intensive, requires high-frequency image processing, and the results depends strongly on the quality of image matching. A last approach is to use the robot mechanical odometry [18],[36]. While this requires the integration of a second sensor on the robot, it can make the system more robust and lighter from a computational point of view as this relieves the visual system from high frame rate computation. Moreover, odometry allows localization for a short time in absence of visual information due to vision failures (dark or dazzle areas, blurry image, occlusions), important changes in the scene that has been learned (objects changes, people...) or network lag in case of remote visual processing.

The main limitation of odometry is the cumulative error due to the integration of noisy measurements that makes the position estimate more and more inaccurate over time. As a consequence, long term use of odometry requires complementary information to enable a correction of this cumulative error. Loop-closure detection provides such an information and can be used to estimate a consistent topo-metric map, an approach also known as pose-graph SLAM [41]. In these approaches, an optimization algorithm is used to correct odometry and estimate the position of nodes that best satisfied loop-closure and odometry constraints. Several relaxation methods have been applied to this problem [25],[16] and efficient recent algorithms can now handle large maps [20],[35],[24],[27],[31] and can also be robust to false loop-closure detection [41].

Our method is different from all the presented ones as it proposes in a single framework: no calibration, low computation with remote processing possibility, and the possibility to build large scale maps and to navigate inside with low cost platforms.

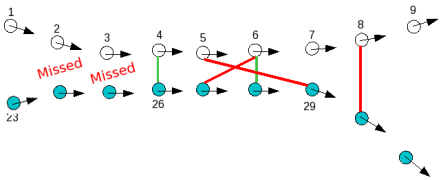


Fig. 2. Illustration of loop-closure detection limitations for topo-metrical mapping. A wrong matching is highlighted in red. It shows a system not reactive enough, not temporally consistent and inaccurate in terms of robot position. In this particular case we expected the loop-closure: 3-25, 4- 26, 5- 27, 6 -28 and 7-29.

3 Baseline system and its limitation

The approach proposed in the current paper relies on visual loop-closure detection [2],[12],[28],[22]. In particular, we started from the approach presented in [2, 3] who developed a real-time incremental topological SLAM approach without any prior information about the environment, using a monocular calibrated camera. This method was based on visual loop-closure detection using a Bayesian filter and the bag of visual words approach for fast images comparison. It presented many advantages such as its simplicity, its speed, the lack of learning stage, and its efficiency (low false alarm rate). However the lack of metrical information (Figure 1.a) made the map ill-posed for robot navigation.

A straightforward solution to add a metrical information was to extend this algorithm by including the robot's odometry information as the relative position between nodes on each link of the graph and by applying a relaxation algorithm each time a loop-closure is detected to enforce map consistency. However, in practice, this approach often leads to maps that were inconsistent with the real world because of several limitations of this purely vision-based loop-closure detection approach.

The first reason was **the lack of temporal consistency** of the loop-closure detection. We show a typical example on Fig. 2 where the matching obtained by [2] was 4-26, 5-29, 6-27 and 6-28 , but the ground truth was 4-26, 5-27, 6-28,7-29. While these loop-closure detections are visually correct as the same scene is present in the matched images, the detections are not temporally consistent.

Secondly, **the responsiveness** of the algorithm was too low. At least two or three images were needed to validate the first loop-closure as the idea was to enforce a very low false alarm rate (see Fig. 1). This is an important limitation

in indoor environments, where common trajectories are mostly seen when a door is crossed for a very short distance. In these cases, many loop-closure are missed and too few constraints are created for map relaxation and odometry correction.

Finally, **the loop-closure definition** itself was not precise enough. In the original approach [2], as in others such as [12], the loop-closure were validated using multiple-view geometry between the matching images. This policy defines loop-closure by the fact that the robot sees the same scene, but does not enforce that the robot positions are very close. Moreover, the scale ambiguity in multiple-view geometry prevents to easily recover the real robot displacement. This strategy prevents for false positive detection but cannot detect well the loop-closure corresponding to the smallest robot displacement. In order to include the graph relaxation algorithm, we therefore had to define a more constrained loop-closure validation stage to only accept loop-closure detections with very close robot positions (see section 4.1.4).

All these problems have to be solved to allow robust navigation. To solve the responsiveness and consistency problems, in [6], we replaced the evolution model of the Bayesian filter which applied a diffusion of the probability over the neighboring locations by an odometry based evolution model (see Fig. 5). Through a probabilistic model of odometry, the evolution model can now take into account not only the nodes topological proximity, but also their relative position. We also proposed a new validation method that imposes closer robot positions for loop-closure and gives information to navigate and in [5] we presented an implementation of a path following algorithm using the maps previously built in a basic scenario.

In the current paper, we summarize in the next section the strategies previously exposed, and present several improvements: a Nth order Bayesian Filter, the use of a faster feature detection and a static dictionary in the likelihood computation, and a new fast validation method based on 2D motion. These modifications improve the whole loop-closure detection framework and allow guidance.

We validate all these improvements on larger maps than previously reported and demonstrate the robustness of our framework by showing navigation in difficult conditions such as: noise on odometry or lack of visual information (people crossing in front of the robot or objects moved in the scene between the learning and the path following phase). Finally, we describe a light and fast implementation of our loop-closure framework which can be used for mapping when the robot is tele-operated and for autonomous navigation when a map is available. The method is robust to noise on odometry and makes it possible to perform the execution of all image processing on

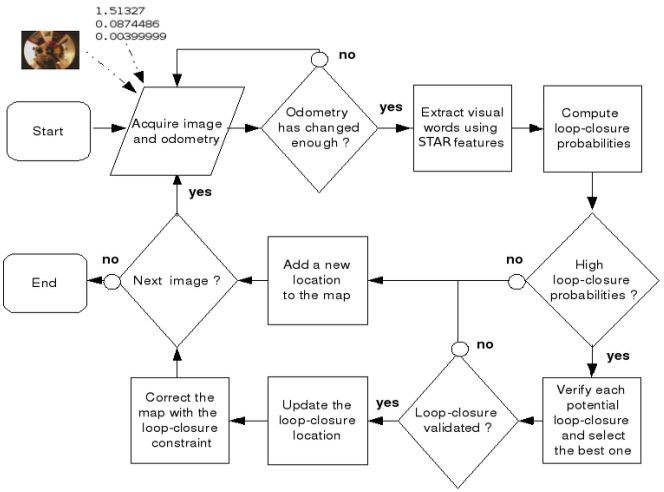


Fig. 3. Processing diagram of the topo-metric map building algorithm.

a remote computer, which makes this approach applicable to low cost robots with wireless connection.

4 Mapping and navigation framework

Our approach (summarized in fig. 3) processes images and odometry in order to build a topo-metric map of the environment. It is based on a loop-closure detection algorithm [2] using a maximum *a posteriori* scheme which detects when the robot comes back to a previously visited location. Such detection allows consistent mapping during environment exploration and robust localization during autonomous navigation.

4.1 Odometry enhanced Visual Loop-Closure Detection

For each node N_i in the current map, we compute the probability of loop-closure with the current image I_t:

$$p(S_t = i | u^t, I^t, M) \tag{1}$$

where $S_t = i$ is event "image I_t comes from node N_i", I^t is the set of images gathered by the robot, u^t is the set of odometry measurements from all the

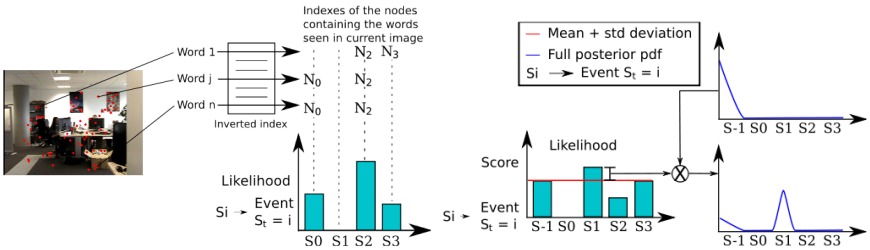


Fig. 4. Left: Fast likelihood computation by using bags of visual words and an inverted index. Right: Computation of the *a posteriori* probability based on the likelihood and the *a priori* probability.

times when images were acquired and $M = N_0, ..., N_n$ is the map. Bayes rule, marginalization and Markov assumption lead to the incremental computation of the *a posteriori* probability as follow (we omit map dependency in all terms for simplicity):

$$p(S_t|u^t, I^t) = \eta. \underbrace{p(I_t|S_t)}_{\text{likelihood model}} . \underbrace{\sum_{j=0}^{n} \underbrace{p(S_t|u_t, S_{t-1}=j)}_{\text{evolution model}} \underbrace{p(S_{t-1}=j|u^{t-1}, I^{t-1})}_{a\ priori\ \text{probability}}}_{\text{prediction}=p(S_t|u^t, I^{t-1})} \tag{2}$$

4.1.1 Likelihood computation

The key point for fast loop-closure detection is the ability to rapidly compute a precise likelihood of the current image for all the nodes of the map. For this, we use a bag of visual words approach [40] that represents images as a set of canonical visual features taken from a codebook. During mapping, we construct an inverted index that links each visual word to the nodes in which it has been seen. Using this inverted index, the current image can be quickly compared to all the maps nodes through a voting method whose result is used to estimate the likelihood (Fig. 4, see [2] for details).

In this paper, we optimize the speed of this process by using a static vocabulary tree [34] and the STAR [1] feature detector coupled with the SURF descriptors [4]. A vocabulary containing 10000 words was created on a set of randomly sampled images in indoor environments for the experiments reported in this paper. While even faster feature detectors and descriptors could be used (such as FAST + BRIEF [22]), the proposed approach was accurate enough and already enables to reach a computation time compatible with real-time

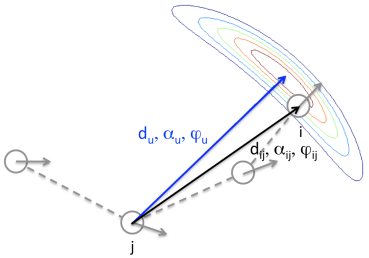


Fig. 5. The evolution model including odometry and relative node positions.

constraints (see section 5.1). A proper comparison of the performances of the feature detector and descriptors on our framework will be addressed in our future work.

4.1.2 Evolution model

By integrating odometry in the evolution model we predict precisely the evolution of the *a priori* probability and therefore we enhance accuracy of loop-closure detection compared to [2] (see Fig. 1 for an example). Starting from a given node, we distribute the probability to each neighboring location in the map depending on the deviation of these nodes relative positions with the robot displacement since the last update d_u, θ_u, ϕ_u measured by odometry (Fig. 5). We used the standard motion model for robot odometry [42], assuming Gaussian noise on the robot displacement measured in polar coordinates:

$$p(S_i|u, S_j) = G_{\mu_d,\sigma_d}(d_{ij} - d_u) \times G_{\mu_\theta,\sigma_\theta}(\theta_{ij} - \theta_u) \times G_{\mu_\phi,\sigma_\phi}(\phi_{ij} - \phi_u) \quad (3)$$

where $u_t = d_u, \theta_u$ gives the odometry displacement in polar coordinates in the frame of the previous robot position and ϕ_u is the variation of robot direction during movement. $G_{\mu,\sigma}(X)$ is the Gaussian distribution of mean μ and variance σ^2. $d_{ij}, \theta_{ij}, \phi_{ij}$ is the relative position between nodes i and j.

4.1.3 Nth Order evolution model

In order to improve the accuracy of the maximum a posteriori scheme we use a Nth order Bayesian filter. To do so, we take into account N previously computed probabilities and the evolution model is applied on the corresponding



Fig. 6. Examples of loop-closure acceptance: First column: match accepted after epipolar geometry validation at a door crossing (Gostai sequence) with more than 1 meter difference between robot positions. Second and third columns : matches accepted using the new validation proposed in this paper with 5cm error.

displacements. The N predictions are combined to give the final prediction according to:

$$p(S_t|u^t, I^{t-1}) = \max_{l=1...N} p(S_t|u_{t-l+1}, S_{t-l}) p(S_{t-l}|u^{t-l}, I^{t-l}) \quad (4)$$

where u_{t-l} is the odometry displacement since time $t-l$. In this paper, we used $n = 4$. This modification makes it possible to improve probability prediction when the movement of the robot is different on a second pass in the same area. It also makes it possible to use different image sampling rates during mapping and later reuse of the map.

4.1.4 Validation of the loop-closure candidates

To get an accurate position detection, we do not just verify and validate the more probable loop-closure position which sometimes do not correspond to the closest node in term of position as it depends on features similarity and on our motion. In order to improve this, the Bayesian filter presented above is used to extract a small subset of potential loop-closure locations whose probability is above a threshold. And then, to find the previous position the closest to the current one, we verify all these locations with a 2D image motion computation based on the STAR keypoints and we select the loop-closure which shows the smallest translation, rotation and scale variation under a threshold. In order to discard outliers in STAR keypoints matching, the 2D motion (translation, rotation and zoom in image plane) is computed using RANSAC, accepting the result only if the number of matching points is above a threshold (30 in the experiments reported in this paper). As shown in figure 6, this validation

is restrictive to guaranty that the current robot position is very close to the previous one.

4.2 Topo-metric mapping

The topo-metric map is constituted of a set of nodes associated with an image and linked by edges. Each node is associated with an absolute pose in the map (x, y, θ), where x and y are the 2D position coordinates and θ an angle representing the direction of the robot when the image was taken. The edges are associated with a relative position between two nodes defined by (d, α, ϕ), where d and α are the polar coordinates of the second node in the coordinate frame of the first, and ϕ is the difference angle between the two nodes direction.

As shown on Fig. 10.a, the geometric consistency of the map deteriorates over time due to the odometry drift. In order to keep the map consistent, we use a relaxation algorithm each time a loop-closure is detected to estimate the nodes absolute position that best satisfy loop-closure and odometry constraints and correct the odometry drift. In this paper, we used the open-source implementation of the TORO algorithm proposed by [24]. Again, more efficient approaches could be used to improve scalability ([27],[31]), but we preferred TORO that gives good results with real-time computation for the typical maps of indoor environments.

4.3 Robot guidance in topo-metric maps

Using the method presented above, it is possible to create a topo-metric map of an environment while tele-operating the robot. We now present our method to enable autonomous navigation using this topo-metric map and the same loop-closure framework. The objective is to be able to guide the robot to the position corresponding to any node of the map, while being robust to temporary loop-closure detection failure, noise on odometry or changes in the scene.

4.3.1 Qualitative localization

The localization uses the same loop-closure detection method as mapping, but the incremental part of the system which adds new locations in the graph and the module which relaxes the topo-metrical map are disabled. Robot lo-

calization is continuously computed as the position of the robot at the last loop-closure location to which we add the relative odometry recorded during robot motion since this point in time. This coupling makes it possible to estimate the robot position even in the absence of loop-closure detection during some time, and still corrects the long-term odometry drift each time a loop-closure is detected.

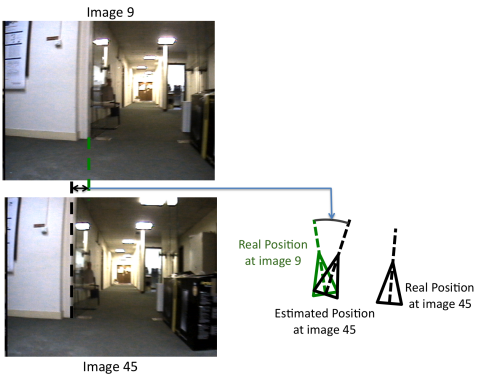


Fig. 7. Illustration of the qualitative visual localization. Left: a loop-closure is detected between two images with 33 pixels of x-axis translation. Given a camera field of view of 45 degrees and 320x240 images, the computed robot rotation is 4.727 degrees. Right: estimated qualitative robot position and true positions.

As it is difficult to estimate precisely the real robot position given matching images when loop-closure are detected, we consider a qualitative localization to navigate in the map in the map. These difficulties are due to the unknown scale factor when computing transformation between two images and to the similarity of image transformation under small camera translation and rotation.

We therefore assume that the image motion is due to a pure rotation of the robot (neglecting translation). This is not true in general, but as will be shown in the next section, this leads to corrections in the path following strategies that guide the robots back to the correct path whatever the real position of the robot is. When a loop-closure is detected, the x-axis translation in pixels extracted during the validation of the loop-closure is used to estimate the angle between the current direction and the direction recorded during the mapping phase (the conversion factor is deduced from a very simple camera calibration as only the horizontal field of view is required). An example is shown Fig. 7.

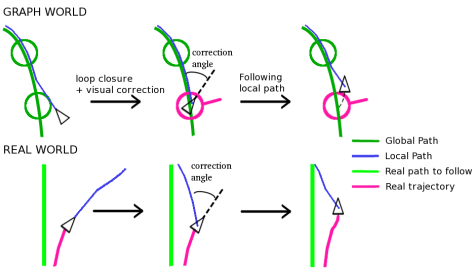


Fig. 8. Path following navigation. Left: The robot follows the local path but deviates from the true trajectory because of the odometry drift. Middle: The visual loop-closure detection framework gives a qualitative localization of the robot in the graph taking into account the deviation in direction. As a consequence, in the real world, the local path is modified and the robot corrects its trajectory in order to stay on the desired path. Right: The robot follows the local path and regain the true trajectory.

4.3.2 Path following navigation

We use Dijkstra algorithm [14] to compute the shortest global path between the node that is the closest to the current robot position and the destination node. This planning takes into account the robot direction in each node to ensure that the robot will travel in the same direction as the mapping phase, thus making it possible to detect loop-closure and correct the estimated position.

During robot motion, a local path is computed joining the current qualitative robot position to the global path. A PID controller applied to the robot direction is then used to compute the velocity of each wheel to follow at best the local path. As stated earlier, the closed-loop operation of this path following strategy will enforce loop-closure detection that will correct the qualitative position estimate. This strategy is also robust to the error made by this qualitative estimate as the localization error always leads to a local path that will guide the robot in the correct direction to reach the global path (Fig. 8).

4.3.3 Remote processing implementation

An interesting characteristic of this algorithm is that path following can be performed even in the temporary absence of loop-closure detection. Therefore image processing for loop-closure detection does not need to be performed at high frame-rate, nor in real-time. We took advantage of these facts to implement a remote image processing strategy that makes the approach suitable for robots with very low computational resources.

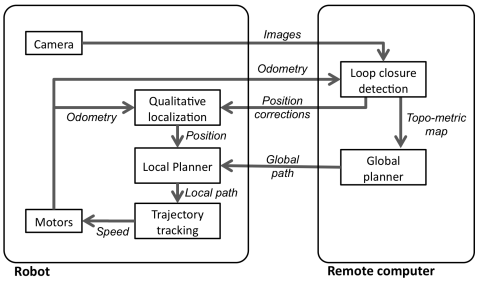


Fig. 9. Simplified diagram of urbi components used by the path following system.

In this implementation, only the local path planning, qualitative localization and PID controller are implemented in real-time on-board the robot. Low resolutions images (QVGA) are transmitted using wireless connection to a remote computer in order to detect loop-closure. When a loop-closure is detected, the loop-closure position and its time stamp is transmitted back to the robot that will use this as the new reference for qualitative localization. Using this strategy, network delays in image transmission and loop-closure detection processing time are not an issue as correction are retroactively applied.

The system has been implemented using *Urbi*, an open-source software platform to control robots. It includes a C++ component library called *UObject* to interface motors, sensors and algorithms. We also use the *urbiscript* language to connect the components together using embedded parallel and event-driven semantics. Figure 9 depicts the repartition of the main software modules used for our robotic path following system. Experiments using this path following module are shown in the the second part of the next section, in the first part we will present some mapping results.

5 Experimental results

5.1 Mapping

Experiments have been performed to validate the loop-closure detection framework in indoor environments using a Pioneer P3DX mobile robot equipped with a Canon VC-C50i camera of 45 degrees of horizontal field of view. The robot was guided to perform loops in indoor environments showing strong perceptual aliasing conditions. The images and the odometry information were taken each time the robot moved at least 25 cm or turned of at least 10

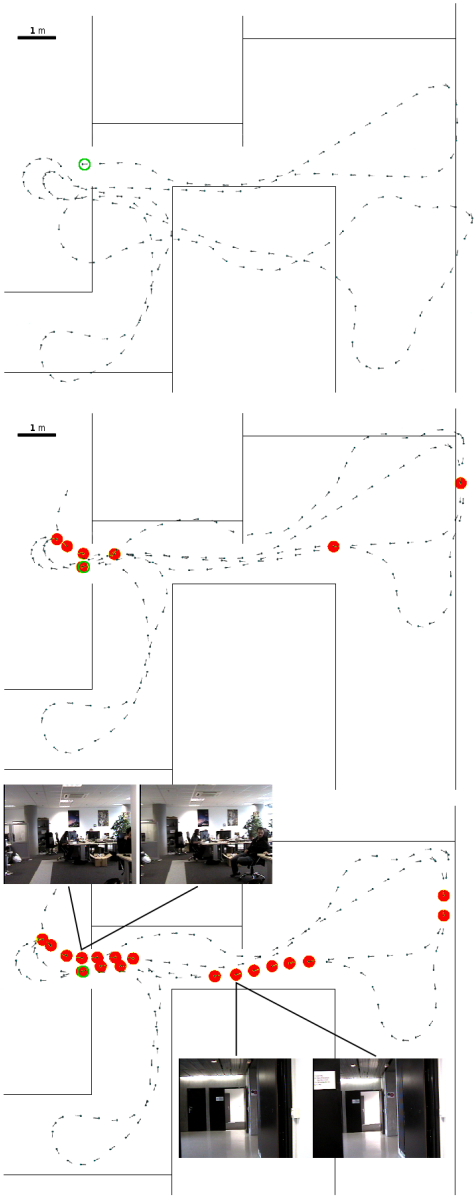


Fig. 10. Gostai sequence: example of topo-metric mapping results a. raw odometry, b. topo-metric map using the loop-closure framework from [2] (7 loop-closures) c. map using the proposed model (18 loop-closures).

Table 1. Compared results of the loop-closure detection system presented in [2] and the approach presented in this paper on four sequences.

	Science Museum	Gostai Offices	UEI Lab	6 rooms
Images	112	169	350	1325
Corresponding figure	1	10	11	12
Distance (m)	38	82	98	255
Loop-closure Ground Truth	9	25	9	≈400
LCD in [2]				
True Positive	4	7	3	-
False Positive	1	2	1	-
CPU Time/image	370ms	410ms	500ms	-
New LCD				
True Positive	7	18	7	352
False Positive	0	0	0	0
CPU Time/image	13ms	15ms	16ms	22ms

degrees. This sampling rate makes it possible to describe the environment without saving too much redundant information, it corresponds to an average acquisition time of one image each 0.7s with the average speed of the robot around 0.4 m/s. The computer used for experimentation was based on an Intel Xeon 3GHz, and we use a small images size: 320x240 pixels (QVGA).

Table 1 shows the performances obtained with the approach proposed in this paper (New LCD) over the one presented in [2] for loop-closure detection coupled with map relaxation. We can see the detection rate is well improved and the inaccurate loop-closure are completely removed.

Figure 10 shows some loop-closure details to highlight the improvements in the proposed method. In particular, we can see the increase in the number of loop-closure detection and that the temporal consistency of the detections is improved as there is no more gap between validated loop-closure images.

To illustrate the system performance and show the robustness of the method we used in parallel the laser SLAM system Karto(1). Figure 11 shows a map created in an environment with a long corridor where the final map is consistent but not perfectly aligned with the ground truth because of the small number of real loop-closures. Note that this error is not problematic at

1 http://www.kartorobotics.com

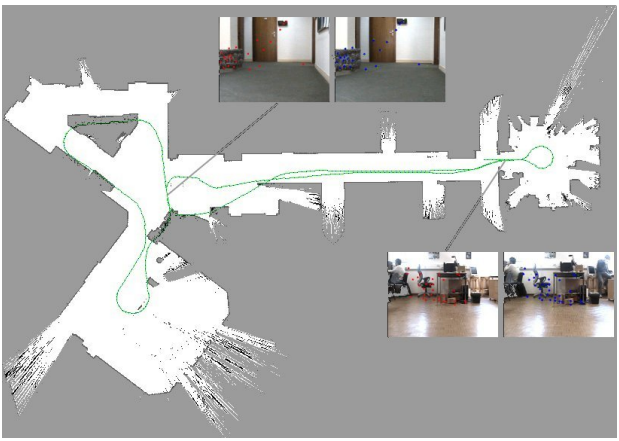


Fig. 11. Correction of the odometry drift with loop-closure detections. The trajectories have been displayed on the 2D laser SLAM map for reference ('UEI Lab' sequence, 350 images, 7 loop-closure). While the global orientation still shows some drift, the local map structure is coherent and the position of the robot at the end of the trajectory is correctly recognized as the starting position.

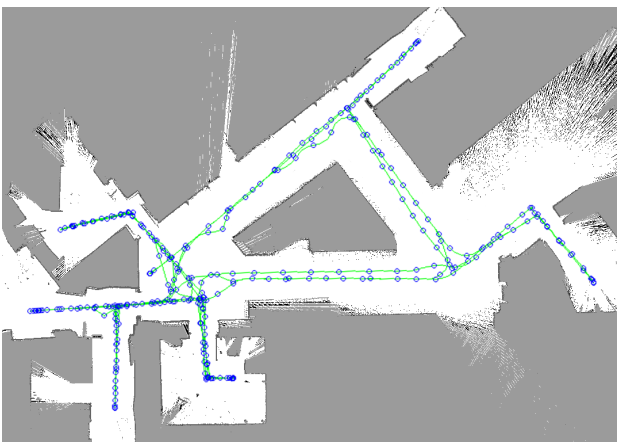


Fig. 12. A large topo-metric map in a 6 rooms environment showing the consistency of the produced map ('6 rooms' sequence, 1325 images, 352 loop-closures).

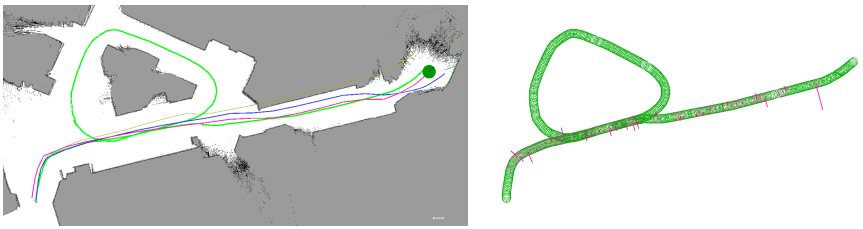


Fig. 13. Example of vision based path following. Green circles represent learned places and magenta circles show the loop-closure locations detected during the navigation. The line in the middle of the magenta circles corresponds to the x-axis translation computed from the matching images.

all as the map topology is consistent and only relative information between nodes will be used for guidance. As a consequence, the small orientation error on the corridor will not affect navigation. Finally, figure 12 illustrates a map created in a larger indoor environment with a central hallway and 6 rooms. This map contains 973 nodes and shows the applicability of our algorithm in an environment of the typical size of standard houses.

5.2 Guidance

Given a map constructed while manually guiding the robot, we have performed autonomous navigation experiments. In all the figures, the reference system gives the real trajectory in a laser map during the learning (green lines) and during the path following run (magenta lines). The odometry recorded during the learning is shown in yellow. The blue line is the odometry recorded during the path following run.

Fig. 13 shows loop-closure detection and estimation made by the qualitative localization procedure while following a path in an indoor environment. The green circles correspond to the topo-metrical map recorded during the learning phase. The magenta circles correspond to the loop-closure locations detected during the path following phase. The magenta line in the circles are proportional to the horizontal translation estimated between the matching images that is used by the qualitative localization to estimate robot's direction. This example illustrates the shortest path computation by avoiding the large loop during replay. Moreover, the images where acquired each 5 cm during mapping and each 25 cm during replay. Thanks to the use of the odometry model and the Nth order Markov model, this does not prevent our algorithm

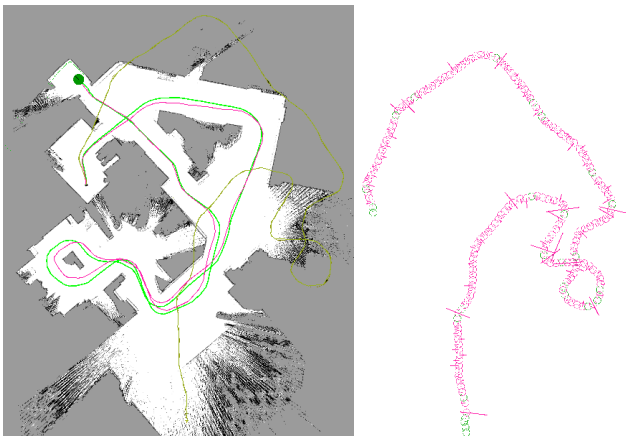


Fig. 14. Large vision based path following (≈ 100 m).

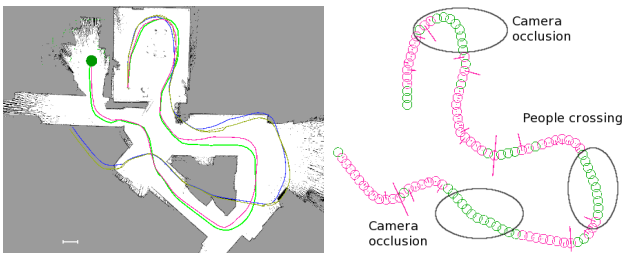


Fig. 15. Example of vision-based path following with camera occlusions or people crossing just in front of the robot.

to correctly detect the loop-closures and successfully guides the robot to the goal position.

Fig. 14 shows another experiment in which the system corrects the odometry drift using the loop-closure and guides the robot to the prescribed goal. The image processing rate was roughly 2Hz and the average speed of the robot in autonomous navigation around 0.4 m/s. The loop-closure detection rate during path following was on average 60% and as shown the path following works well. The missed loop-closure occurs mainly because while turning or deviating from the trajectory some images are too close to a wall or obstacle, that is to say presenting poor features or features different from the learning stage ones. This makes the loop-closure probability drop below the threshold or prevent the validation stage from accepting the potential loop-closure. Another reason of the missed loop-closure during path following is that occa-

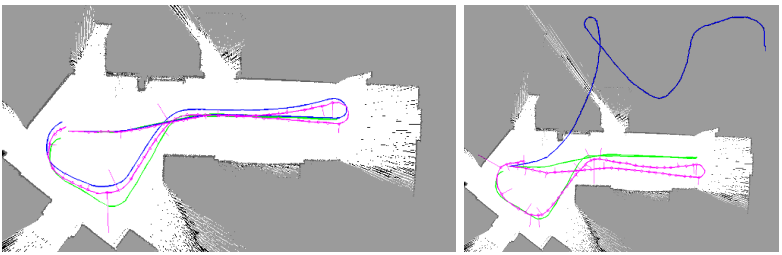


Fig. 16. Two examples of vision based path following using the same map (in green). Raw odometry is in blue, replayed trajectory in magenta. Left: path following using the pioneer 3DX odometry. Right: path following using simulated noise of 6σ.

sionally some images are blurry because of the vibrations of the platforms or fast turns.

To better demonstrate the robustness to vision failure or missed detection, on Fig. 15 the camera was occluded for short times and people were crossing the robot path in front of the robot. This experiment shows that the system is robust to vision failures up to 20 images that is to say approximatively 5 meters. For this experiment with dynamic obstacles and camera occlusions, the average loop-closure detection rate felt down below 40% and the robot was still able to follow the trajectory and reach its goal. It has to be noted that the main odometry drift appears when the robot is turning fast, and that camera occlusions during these turns are much more difficult to compensate. This robustness is however sufficient to accommodate reasonable environment changes such as people crossing in front of the camera or large furniture displacement which could prevent the robot from detecting loop-closure for few seconds. It has to be noted that environmental changes should even affect less the loop-closure detections as it should remove only few features whereas the camera occlusion drastically remove any features.

The Pioneer 3DX platform has a quite efficient odometry compared to other wheeled robots or humanoid robots. To quantify the robustness to odometry drift we performed navigation experiments while simulating several level of odometry noise. This was achieved by taking the displacement computed by the laser SLAM and corrupting it every 50ms with a Gaussian noise with the same mean and variance:

$$d = d + G(\sigma, \sigma); \theta = \theta + G(\sigma, \sigma); \phi = \phi + G(\sigma, \sigma) \tag{5}$$

The variance of these noises were taken as multiple of a base value of $\sigma = 5.10^{-4}$. This noisy displacement value was used as the odometry by our

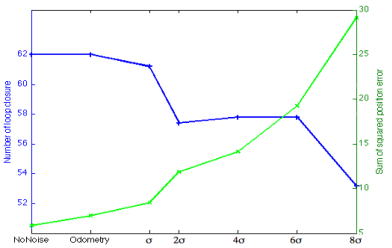


Fig. 17. Number of loop-closure detected (blue) and sum of squared differences between reference path and path following (green) for several levels of simulated odometry noise.

path following system. In all experiments, the same topo-metric map was used, created using the laser SLAM displacements. To obtain representative results, we executed the path following 5 times for each noise level and plotted the average result over these five cases. Figure 16 shows two of these experiments, with the Pioneer robot odometry and with a simulated noise using 6σ variance.

Figure 17 shows the number of loop-closure detected during these experiments. While the number of loop-closure decreases with noise, this number remains high enough to ensure position correction for path following. This figure also shows the error in position between the trajectory executed during mapping and the trajectory executed during replay, measured with the laser SLAM position. This position error clearly increases with the odometry noise but remains small enough to ensure path following success in all trails up to 6σ noise. Figure 18.a shows that at 8σ, 3 out of 5 trials lead to a failure, the robot hitting a corridor wall during path following. In this same figure, we can see that using only the raw odometry, 4 out of 5 trials lead to failures with the first noise level of 1σ and that all trials fail after the 2σ level. In comparison, our qualitative position estimate makes it possible successfully follow the path in all situations up to 6σ noise. This noise applied is quite important which demonstrates that our algorithm can be working on many wheeled robot providing odometry. The robustness to odometry noise also lets us predict that the same framework could also be applied on different platforms like legged robots for example known to provide poor quality odometry. This possibility will be tested in our future work.

Finally, figure 18.b shows the mean correction applied by our algorithm during path following. It can be noted that corrections increase with the odometry noise, but that these corrections do not converge to zero with a noise free odometry. This is caused by the sampling policy of images: as images are not taken at the same position during mapping and path following, each loop-closure will produce a direction correction, in particular during rotation, even

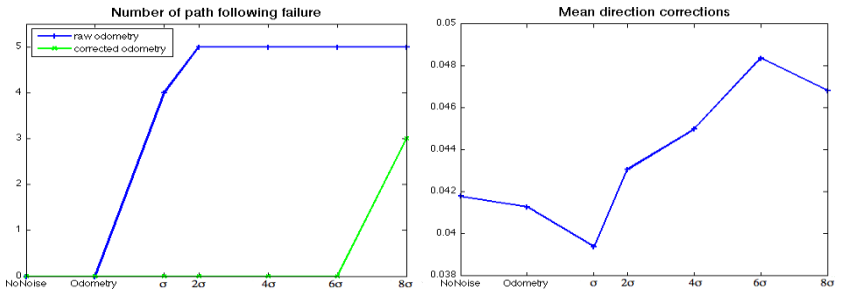


Fig. 18. a. Number of failed path following over five trials using raw odometry (blue) and the qualitative position estimated by our algorithm (green) for several levels of simulated odometry noise. b. Mean direction correction applied during path following, in radians.

if the robot estimated position is perfect. However, it can be seen in the path following error (Fig. 17) that these corrections do not prevent efficient path following in the case of a precise odometry, and make path following possible in the case of a noisy odometry.

6 Discussion

The algorithm proposed in this paper is well adapted for mapping in indoor environments as the path is more constrained. Indeed in indoor environment, doors are a required route to go from one room to an other and therefore forces the robot to close loops, even with our restricted definition. Outdoor environment have also been tested successfully but usually provide less loop-closure detection as the path are often less constrained. The qualitative guidance approach is however as efficient outdoor as it is indoor as the robot is guided along already executed routes, and is able to close loop frequently and effectively limit the odometry drift.

Our path following system uses only relative information from the odometry and between current images and past images. This is linked to the fact that the qualitative localization itself is relative to the position of loop-closure detections. As a consequence, an error in the global metrical position of a node such as in figure 11 where the map diverges from the true corridor direction, will not prevent the robot to correctly follow a path inside this corridor. Another consequence is that the use of a relaxation algorithm to obtain a globally consistent map is not required by the path following system as long as we only follow trajectories really executed during map building. In our

application, this relaxation algorithm is however important in the prediction step of the Bayesian filter and when following trajectories computed by the Dijkstra algorithm. Indeed, a coherent map is required to correctly predict position probability with the odometry probabilistic model and the global metrical path followed using the qualitative localization must not contain discontinuities.

As shown in figure 13 another interest of using an odometry model an a qualitative metrical localization is the possibility to use different image sampling policies between mapping and autonomous navigation. In conjunction with the Nth order Markov model, this makes it possible to create high-density maps, sampling images every few centimeters, during supervised navigation, while using a lower sampling frequency to reduce computational burden during autonomous navigation.

Finally, the current model only allows to directly reproduce paths, therefore assuming the absence of dynamic obstacles. In dynamic environments, it would be possible to couple our method with a local obstacle avoidance strategy based on sonars for example. The qualitative localization based on optometry would make it possible for the robot to come back to the global path after avoiding the obstacle, which would not be possible with a direct visual servoing approach. Adding such an obstacle avoidance strategy could even improve the number of loop-closure detection by forcing the robot to always use similar trajectories (centering the robot along corridors for example), thus improving the precision of the qualitative localization.

7 Conclusion

We have presented a system for topo-metric mapping and navigation based on an appearance-based loop-closure detection framework combining vision and odometry. In particular, we have proposed an optimized loop-closure detection algorithm that shows high responsiveness, consistent and precise position detections while working with several kinds of uncalibrated cameras, requiring limited computational resources, and allowing the remote processing. The loop-closure detection framework makes it possible to build incrementally consistent topo-metric maps in real-time while remotely guiding the robot without assumptions about the environment. We also proposed a qualitative localization method based on loop-closure detections and odometry that makes it possible to guide a robot along routes computed from the topo-metric map. This navigation method is robust to temporary vision failures (people cross-

ing, objects moved) noise on the odometry and delays in the vision processing, thus making it applicable to low-cost platforms.

In future work, we plan to develop a global localization strategy in order to be able to re-localize the robot inside the map after it has been turned off. This strategy could be based on a standard behavior such as wall or corridor following in order to provide a first loop-closure detection and initialize localization. We also plan to study the use of a pan-tilt camera in order to make it possible to detect loop-closure with trajectories followed in reverse direction, as in the current framework, the robot can only follow trajectories in the same direction as during mapping. We finally planned to add obstacles avoidance capabilities using sonar data to be robust to dynamic obstacles and more important changes in the environment.

References

[1] M. Agrawal, K. Konolige, and M. R. Blas. Censure: Center surround extremas for realtime feature detection and matching. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *European Conference on Computer Vision*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115, Marseille, 2008. Springer.

[2] 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*, 24(5):1027–1037, October 2008.

[3] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Real-time visual loop-closure detection. In *IEEE International Conference on Robotics and Automation*, pages 1842–1847, Pasadena, 2008. IEEE.

[4] H. Bay, T. Tuytelaars, and L.V. Gool. Surf: Speeded up robust features. In *9th European Conf on Computer Vision*, volume 110, page 34659. Elsevier, 2006.

[5] S. Bazeille, E. Battesti, and D. Filliat. Qualitative localization using vision and odometry for path following in topo-metric maps. In *European Conference on Mobile Robotics*, pages 303–308, Orebro, 2011.

[6] S. Bazeille and D. Filliat. Incremental topo-metric slam using vision and robot odometry. In *Proceedings of the International Conference on Robotics and Automation*, Shanghai, 2011. IEEE.

[7] J.L. Blanco, J. González, and J.-A Fernández-Madrigal. Subjective local maps for hybrid metric-topological slam. *Robotics and Autonomous Systems*, 57:64–74, 2009.

[8] O. Booij, B. Terwijn, Z. Zivkovic, and B. Kröse. Navigation using an appearance based topological map. In *IEEE International Conference on Robotics and Automation*, Roma, 2007. IEEE.

[9] D. Burschka and G. Hager. Vision based control of mobile robots. In *Proc. of the International Conference on Robotics and Automation*, Seoul, 2001. IEEE.

[10] Z. Chen and T. Birchfield. Qualitative vision-based path following. In *IEEE Transactions on Robotics*, volume 25, pages 749–754, 2009.

[11] J. Correa and A. Soto. Active visual perception for mobile robot localization. *Journal of Intelligent and Robotic Systems*, 58:339–354, 2010.

[12] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27:647–665, 2008.

[13] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

[14] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[15] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette. Outdoor visual path following experiments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4265 – 4270, San Diego, 2007. IEEE.

[16] Tom Duckett, Stephen Marsland, and Jonathan Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12(3):287–300, 2002.

[17] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *International Conference on Computer Vision*, pages 1–8, Rio, 2007. IEEE.

[18] David Filliat and Jean-Arcady Meyer. Global localization and topological map-learning for robot navigation. In *Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, ICSAB, pages 131–140, Cambridge, MA, USA, 2002. MIT Press.

[19] 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*, pages 3872 – 3877, San Diego, 2007. IEEE.

[20] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics and Automation*, 21(2):196–207, 2005.

[21] Siyao Fu and Guosheng Yang. Uncalibrated monocular based simultaneous localization and mapping for indoor autonomous mobile robot navigation. In *Networking, Sensing and Control*, pages 663 – 668, Okayama, 2009. IEEE.

[22] D. Galvez-Lopez and J.D. Tardos. Real-time loop detection with bags of binary words. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pages 51–58, San Francisco, 2011. IEEE.

[23] T. Goedemé, M. Nuttin, T. Tuytelaars, and L. Van Gool. Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236, 2007.

[24] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems*, pages 65 – 72, Atlanta, USA, June 2007.

[25] A. Howard, M. J. Mataric, and G. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1055–1060, Maui, 2001. IEEE.

[26] S. A. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. Robot. Automat.*, 12:654–670, 1996.

[27] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *Robotics and Automation, IEEE International Conference on*, pages 3281–3288, Shanghai, 2011. IEEE.

[28] Aram Kawewong, Noppharit Tongprasit, and Osamu Hasegawa. Pirf-nav 2.0: Fast and online incremental appearance-based loop-closure detection in an indoor environment. *Robotics and Autonomous Systems*, 59(10):727 – 739, 2011.

[29] K. Konolige and M. Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transaction on Robotics*, 24(5):1066–1077, 2008.

[30] J. Kosecká, F. Li, and X. Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52:209–228, 2005.

[31] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G2o: A general framework for graph optimization. In *Robotics and Automation, IEEE International Conference on*, pages 3607–3613, Shanghai, 2011. IEEE.

[32] M. Milford and G. Wyeth. Persistent navigation and mapping using a biologically inspired slam system. *The International Journal of Robotics Research*, 29:1131–1153, 2009.

[33] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.

[34] 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*, pages 2161–2168, Washington, 2006. IEEE.

[35] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2262–2269, Orlando, 2006. IEEE.

[36] 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*, Taipei, 2003. IEE.

[37] M. Saedan, Lim Chee Wang, and M.H. Ang. Appearance-based slam with map loop closing using an omnidirectional camera. In *Proceedins of the ieee/asme international conference on Advanced intelligent mechatronics*, pages 1 – 6, Singapore, 2007. IEEE.

[38] C. Sagues and J. Guerrero. Visual correction for mobile robot homing. *Robotics and Autonomous Systems*, 50:41–49, 2005.

[39] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems*, June 2009.

[40] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, pages 1470 – 1477, Nice, 2003. IEEE.

[41] Niko Sünderhauf and Peter Protzel. Towards a robust back-end for pose graph slam. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1254–1261, Saint Paul, 2012. IEEE.

[42] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[43] C. Valgren, T. Duckett, and A. Lilienthal. Incremental spectral clustering and its application to topological mapping. In *IEEE International Conference on Robotics and Automation*, Roma, 2007. IEEE.

[44] Z. Zivkovic, B. Bakker, and B. Kröse. Hierarchical map building using visual landmarks and geometric constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2480 – 2485, Edmonton, 2005. IEEE.