Incremental vision-based topological SLAM

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

***Abstract*— In robotics, appearance-based topological map building consists in infering the topology of the environment explored by a robot from its sensor measurements. In this paper, we propose a vision-based framework that considers this data association problem from a loop-closure detection perspective in order to correctly assign each measurement to its location. Our approach relies on the visual bag of words paradigm to represent the images and on a discrete Bayes filter to compute the probability of loop-closure. We demonstrate the efficiency of our solution by incremental and real-time consistent map building in an indoor environment and under strong perceptual aliasing conditions using a single monocular wide-angle camera.**

I. Introduction

Simultaneous Localization And Mapping (SLAM, [1]) is today one of the most active research area in robotics. The SLAM problem consists in localizing a robot while simultaneously building a map of the environment. Two different approaches exist to address the SLAM problem. The first one models the environment using a *metric* map, enabling accurate estimation of the robot's position. It provides a dense representation of the environment and is particularly well suited to precise trajectory planning. In the second approach, the environment is segmented into distinctive places that form the nodes of a graph (or *topological* map) and whose neighboring relations (i.e. whether or not a place is accessible from another one) are modeled using the edges of this graph. Topological mapping relies on a higher level of representation than metric mapping, allowing for symbolic goal-directed planning and navigation. It also provides a more compact representation that scales better with the size of the environment.

Of particular interest when addressing the SLAM problem is the ability of detecting loop-closures: it consists in correctly associating current measurements with information stored in the map when the robot is coming back to an already mapped part of the environment. Defined more precisly in the topological mapping case, loop-closure detection entails finding the node to which current measurements pertain when the robot enters a previously visited place. Accordingly, loop-closure detection is a data association problem. In [2], we proposed a vision-based framework to overcome this difficulty so as to reinitialize a metric SLAM algorithm when a loop has been closed: at each new image acquisition, the loop-closure probability is computed, making it possible to detect those images that come from the same location in real-time and in an incremental fashion, even under strong perceptual aliasing conditions. In addition, we proposed in more recent work [3] an extension of [2] that enables the use of several image representations and that has been validated on both indoor and outdoor image sequences.

In this paper, we present a real-time, online, appearance-based topological SLAM algorithm relying on [2] to efficiently handle loop-closures with a monocular handheld wide-angle camera. When a new image is acquired, localization is attempted by searching for loop-closures among the nodes of the topological map. In case of success, the loop-closing node is updated with the information coming from the current view. Otherwise, a new node containing this information is added to the map. Loop-closure detection is done according to the method detailed in [2]: images are quantized based on the visual bag of words scheme [4], with a discrete Bayes filter used to estimate the probability of loop-closure. Epipolar geometry [5] helps discarding outliers in an ultimate validation step when this probability is above some threshold.

In section 2, we give a review of related work on topological SLAM. Our approach is detailed in the 3 following sections. Experimental results are reported in section 6 and discussed in section 7, before the conclusions of the last section.

II. Related Work

In an early work on topological SLAM [6], a Partially Observable Markov Decision Process (POMDP) model is used to estimate the position of a robot as a probability distribution. More recently, the authors of [7] adapted this approach to perform hybrid topological-metric SLAM using a 360° laser scanner, also enabling loop-closure detection capabilities. However, POMDP models are generally not suited to adaptative online map building since they need to be learned in an offline process ([8]) or set manually from prior information about the environment's geometry, appearance and topology (e.g. the environment is made of corridors and rooms, corridors' junctions are at right angles and the robot is expected to be in a corridor most of the time, [6], [7]).

Inference has been investigated by the authors of [9] and [10] to address the topological SLAM problem: topologies are sampled over the space of topological maps and matched

with measurements ([10]) or actions ([9]) in order to accept or discard each individual sampled map. This process can be run online along with information acquisition but the complexity involved by the sampling step only allows mapping of environments with few distinct places (i.e. map size is limited to 15 nodes).

Most of the recently developped approaches to the topological SLAM problem are based on appearance and rely on omnidirectional vision ([11], [12], [13], [14]). A similarity distance between images is defined to set the edges of the map, with very similar images considered as originating from the same place and thus as corresponding to the same node. Appearance-based approaches provide an efficient segmentation of the environment, since omnidirectional images make it possible to recognize a place from distant points of view. However, none of the approaches listed above meet both the online and real-time requirements: either input information is processed in a previous offline step ([11], [12], [14]) or the complexity of the image similarity computation is untractable in real-time conditions ([13]).

The authors of [15] present a real-time vision-based framework to perform topological SLAM using a single monocular camera. The approach relies on the bag of words paradigm [16]: images are quantized as a set of unordered elementary features (the visual words) taken from a dictionary (or codebook). The dictionary is built by clustering similar visual features extracted from the images into visual words. Using a given dictionary, images are simply represented by the frequencies of the words they contain. In [15], images are represented as vectors of visual words statistics taken from an offline-built visual vocabulary and a vote procedure makes it possible to efficiently find the past images that look like the current one. This approach is in many points very similar to our previous work ([2]) regarding loop-closure detection. Still, the implementation of the visual bag of words scheme proposed in [15] relies on an offline process for the vocabulary construction.

The main contributions of the work reported here are twofold. First, our method is based only on appearance and uses a single monocular camera, whereas most of the appearance-based approaches take omnidirectional or panoramic images as input. Second, the framework proposed here is fully incremental and processing is performed in real-time.

III. Topological SLAM

The topological map is a graph whose nodes correspond to distinct locations in the environment and whose edges model time neighboring relations between the nodes. The challenge in this appearance-only approach is to decide when to add a new node to the map when a new image is provided by the camera. As stated by the authors of [10], a topology is a set partition over the set of measurements (i.e. multiple images may correspond to one and the same node in the map). Therefore, in order to infer the correct topology from the measurements, we must be able to detect when an image comes from an already visited location and thus

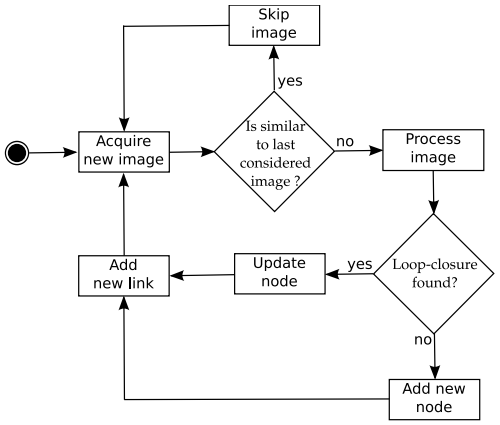

Fig. 1. Overall process diagram (see text for details).

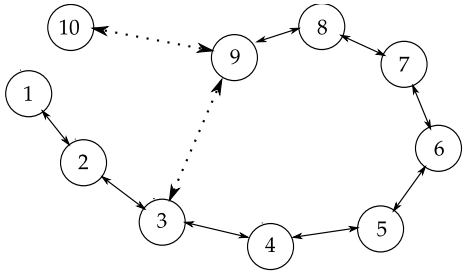

Fig. 2. Time neighboring relations between nodes: in the graph shown here, node 9 is the last added node. When a new image is considered, either a new node will be added (node 10) or an existing one will be updated (node 3). In both cases a new edge connected to node 9 will be added.

pertains to an existing node: this is the loop-closure detection problem as defined for the topological mapping case. From this observation, we add a new node to the map only when no loop-closure has been detected.

The overall processing of our SLAM algorithm is illustrated in the diagram shown in figure 1. When a new image is acquired, it is first compared to the last considered image to determine if it has to be taken into account (e.g. when the camera is standing still images can be skipped, see section V). Then, Bayesian loop-closure detection following the work detailed in [2] is attempted. If successfull, the loop-closing node is updated with the visual information coming from the current image. Otherwise, a new node containing this information is added to the map. In both cases, a new link with the last updated or added node is created: edges model the time order in which locations are travelled by the camera (see figure 2).

In order to efficiently and robustly detect loop-closures, each node has to be characterized using a compact and relevant representation of the corresponding location. Moreover, since a node may be characterized by multiple images (e.g. in case of loop-closures), this representation must be extendable so as to be augmented when the node is updated. To this end, we choose to characterize a node using the collection of visual words found in the images pertaining to the corresponding location (see figure 3). In the visual bag of words implementation [4] applied here, a visual word is

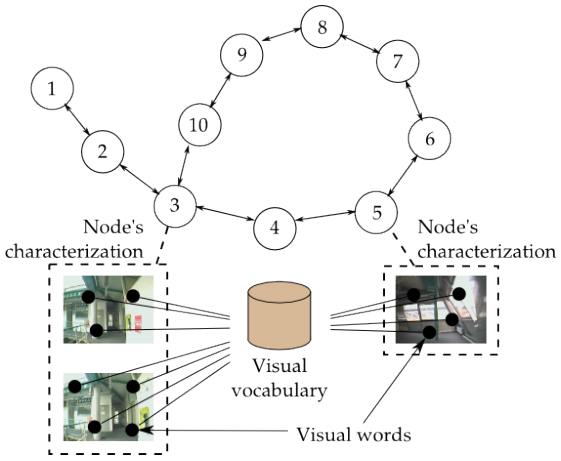

Fig. 3. Illustration of the characterization of the nodes: a node is characterized using the visual words found in the images pertaining to the corresponding location. Since node 3 in the map is a loop-closing node, the visual words from two images are used for its characterization.

obtained by incrementally combining similar visual features in an agglomerative manner: visual words are clusters of similar visual features that are stored in a visual vocabulary. In this paper, SIFT (Scale Invariant Feature Transform [17]) keypoints are used as visual features: 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.

IV. Bayesian Loop-Closure Detection

As explained earlier, loop-closure detection helps deciding if we should add a new node to the map or update a previous one when considering a new image. The structure of the map and its coherence regarding the distinct locations of the environment thus strongly depends on the robustness of loop-closure detection: if a loop-closure is missed or erroneously detected, the overall topology will no longer be consistent. However, there can be small divergences between the infered map and the true topology as long as the true topology is globally respected (e.g. a small time delay between a loop-closure's occurrence and its detection is acceptable).

The Bayesian loop-closure detection method introduced in [2] is used here. The approach consists in detecting loop-closures based on the similarity between images with particular attention payed to the time coherence of the detection. To this end, a discrete Bayes filter is employed to compute the probability of loop-closure each time a new image is considered. In this paper, the discrete Bayes filter is adapted to find the node N_j of the map whose characterization is similar enough to the current image I_t to consider that I_t comes from the location corresponding to N_j. In a probabilistic framework, and using the quantized representation z_t of I_t (i.e. z_t is the collection of visual words found in I_t), this can be expressed as searching for the node N_j of the map $M_{t-1} = \{N_0, \ldots, N_n\}$ whose index satisfies:

$$j = \text{argmax}_{i=-1,\ldots,n} p(S_t = i | z_t, M_{t-1}) \quad (1)$$

where $S_t = i$ is the event that image I_t comes from the location corresponding to node N_i, or put more simply it is the event that I_t comes from N_i. We also introduce $S_t = -1$ to account for the no loop-closure event at time t. Note that solving equation 1 relies on the map built until time $t-1$ (i.e. M_{t-1}): the update of the map, leading to M_t, is done afterwards, according to the solution obtained for equation 1 (see figure 1). As shown in [2], solving equation 1 requires the incremental computation of the *full posterior*, as follows:

$$p(S_t | z_t, M_{t-1}) = \eta\, p(z_t | S_t, M_{t-1}) \sum_{j=-1}^{n} p(S_t | S_{t-1} = j, M_{t-1})\, p(S_{t-1} = j | M_{t-1}) \quad (2)$$

where η is a normalization term. The recursive aspect of the mathematical formulation proposed in equation 2 stems from the fact that $p(S_{t-1}|M_{t-1})$ is a factored rewriting of $p(S_{t-1}|z_{t-1}, M_{t-2})$, the posterior at time $t-1$, given that M_{t-2} has been updated with z_{t-1} to form M_{t-1}.

From equation 2, it can be seen that the estimation of the full posterior requires the computation of the conditional probability $p(z_t|S_t, M_{t-1})$, which is considered as a likelihood function $\mathscr{L}(S_t|z_t, M_{t-1})$ of S_t: we evaluate, for each entry $S_t = i$ of the model, the likelihood of the currently observed words z_t (see section IV-B). Also, we can observe that a time evolution model $p(S_t|S_{t-1} = j, M_{t-1})$ is needed to sum the full posterior calculated one step before over all possible transitions between $t-1$ and t (see section IV-A).

A. Transition from $t-1$ to t

As explained in [2], the time evolution model gives the probability of transition from one state i at time $t-1$ to every possible state j at time t, enforcing the temporal coherency of the estimation and limiting transient detection errors. In this paper, we introduce a new image acquisition policy to skip consecutive images that are too similar (see section V), which is what may happen when the camera is standing still. Then, when an image is considered for processing, it is assumed that the camera has moved. The probability of transition from state i to state j with $i, j > -1$ is thus modeled using a sum of Gaussians, whereas a single Gaussian was used in our previous work. The sum of Gaussians is set to give more emphasis to states $j = i-1$ and $j = i+1$ when considering state i, the new image acquisition policy letting us assume that the probability of staying in state $j = i$ is low (see figure 4).

B. Likelihood in a voting scheme

The likelihood function $\mathscr{L}(S_t|z_t, M_{t-1})$ is obtained using a simple and efficient voting scheme (see [2] for details). An inverted index helps associating each visual word in the visual vocabulary with the nodes of the map it characterizes. Therefore, when an image is processed, the found visual words vote for the nodes they characterize using the *tf–idf* coefficient [18]: this procedure makes it possible to quickly

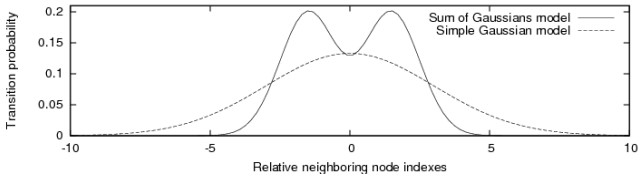

Fig. 4. Sum of Gaussians vs single Gaussian for the time evolution model: the sum of Gaussians model (solid line) gives more emphasis to neighboring states than the single Gaussian model (dashed line), which puts more emphasis to the centre.

vote for the nodes whose characterization is similar to the current image. The particular case of the no loop-closure event is easily handled by adding an entry to the inverted index corresponding to a virtual node characterized with the mostly seen visual words.

C. A posteriori hypotheses management

When the time evolution model has been applied and the product with the likelihood done, the full posterior is normalized. We then select as possible loop-closure hypothesis the node whose probability is above some threshold (0.8 in our experiments). Since the posterior may be diffused over neighboring nodes rather than peaked over a single one, the score's sum over neighboring nodes is used instead of a single probability score. The selected hypothesis is next submitted to the epipolar geometry ([5]) validation step to discard outliers: a RANSAC procedure entails finding a consistent camera viewpoint transformation between one image of the selected node and the current frame by matching the corresponding SIFT features using a threshold on the average reprojection error. If successfull, the loop-closure hypothesis is accepted and the loop-closing node is augmented with the visual words from the current image. Otherwise, if no hypothesis has been fully validated, a new node characterized with these visual words is added to the map.

V. Local Image Similarity

In this paper, a simple method to compute local image similarity has been introduced to overcome some of the limitations of our previous work regarding loop-closure detection. First, every acquired image was considered for processing, provoking loop-closure detections when the camera is standing still for a while. Second, in the discrete Bayes filter implementation proposed before, a cache mechanism was used to delay the "release" of a hypothesis: since each image is similar to its neighbors in time, immediately releasing a hypothesis would result in local loop-closure detections when the next image would be processed. The size of this cache was empirically set to 10 images and was dependent on the camera frame rate and on the velocity of camera motion. Thus, every hypothesis was released only after 10 images had been processed, making it impossible to check for loop-closures between I_i and $I_{i-10}, \ldots, I_{i-1}$ (see [2] for details). Using such a fixed cache size could result in local loop-closure detections in case of slower than expected camera displacement.

The local image similarity is defined between a node N_i and the current image I_t as the percentage of visual features extracted in I_t that are visual words characterizing N_i. The higher the percentage, the higher the similarity.

Based on this criterion, a newly acquired image is accepted for processing only if the local similarity with the last added or updated node is below 90%, allowing to skip very similar consecutive images. Moreover, the cache mechanism is now also governed by the local image similarity: each node is kept in cache as long as its local similarity with the current image is above 20%. A node is thus released and effectively taken into account in the map only when it is different enough from the current image to avoid local loop-closure detections.

Note that we did not differenciate here the addition of a new node to the map from the update of an existing one. In both cases, a new node is first created, characterized with the visual words found in the current image, and immediatly pushed in cache. Then, when the node is released, two different treatements are possible: either it is used to update the characterization of an existing node (i.e. in case of a loop-closure), or it is added as is in the map. In the first case, updating an existing node with a released one can be considered as merging their characterizations.

VI. Experimental Results

Experimental results were obtained from a video sequence lasting 247 seconds and acquired at 1Hz using a handheld wide-angle camera. During the travel of the camera, several loops were closed in a particularly challenging indoor environment with strong perceptual aliasing (see figure 8 for examples of the images composing the sequence).

The trajectory of the camera is shown superimposed on the floor plan of the environment in figure 5, left part. The travel begins with a first loop in the blue area, before entering the magenta area. After that, the camera comes back into the blue area and goes straight ahead to the red area. It then comes back again to the blue area before discovering the green area. The travel ends near the 8^{th} white circle. On the right part of the figure is shown the resulting topological map, for which the same color convention is used in order to easily identify mapped areas. It can be seen that all the loops corresponding to a return of the camera into the blue area are correctly detected, as well as the multiple loops done inside the blue area. This is shown by the yellow color of the trajectory on the floor plan but also by the yellow circles that highlight the loop-closing nodes of the map. Note that a loop-closing node may correspond to multiple loop-closure: for example, the camera passed 4 times around the 6^{th} white circle, causing several nearby loop-closing nodes to encode for multiple similar images.

When considering the topological map more carrefully, we can observe that there is some delay between the occurrence of a loop-closure and its detection. This can be seen each time the camera is coming back to the blue area: the true loop-closing node (i.e. the node corresponding to the effective return of the camera in an already visited place) precedes the loop-closing node selected by the SLAM algorithm. For

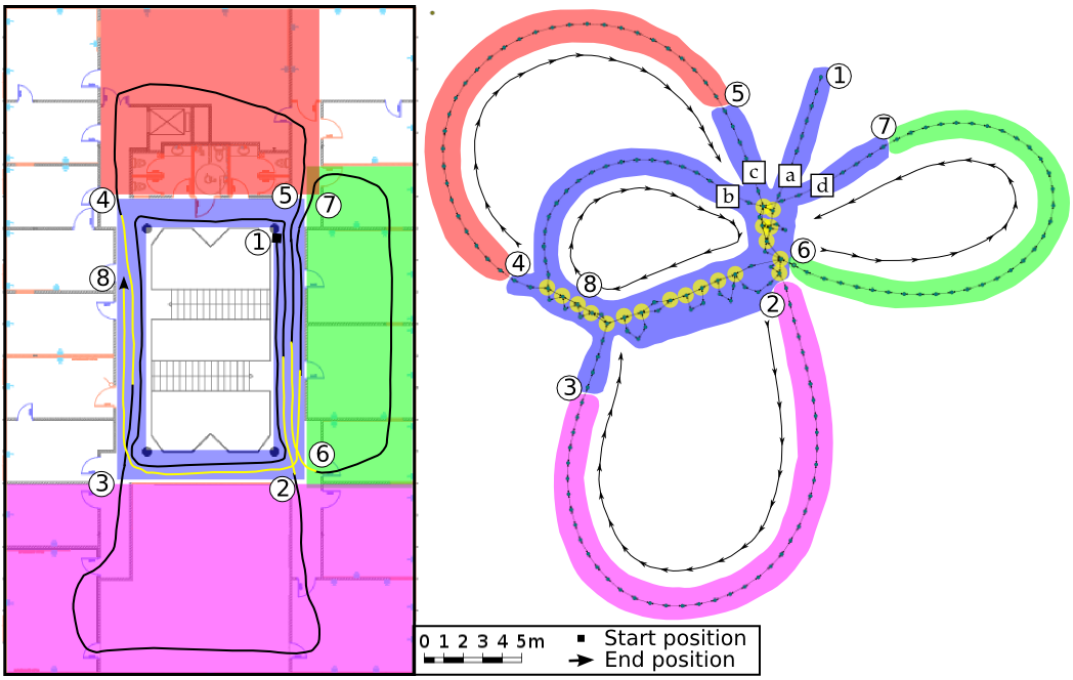

Fig. 5. Floor plan of the travelled environment superimposed with the trajectory of the camera (left part of the figure) and corresponding topological map (right part of the figure). The graph layout is performed using a simple "spring-mass" model [19]. See text for details.

example, when the camera is coming back from the magenta area to the blue one, the effective transition (near the 3^{rd} white circle) is 3 nodes away from the corresponding loop-closing node (i.e. the yellow circled node that links together the magenta and blue branches of the topological map). The reasons for this are twofold. First, we already observed in [2] that the loop-closure detection had a low responsiveness, which was partially motivated by the robustness to transient detection errors. Second, when the camera is travelling along consecutive already visited locations (e.g. "d" in figure 5), the likelihood may be divided among several nodes that correspond to the previous passings of the camera (e.g. "a", "b" and "c" in figure 5) and that all lead to a common loop-closing node. Thus, the likelihood exhibits multiple peaks (see figure 6) that prevent the full posterior from being unambiguously focused on one particular hypothesis. However, further image acquisition will help removing this ambiguity when the camera reaches the loop-closing node that joins the branches corresponding to the past passings.

In the results presented above, the overall processing has been done online and in real-time: 123s were needed to process the 247s of the sequence using a Pentium Core2 Duo 2.33GHz laptop and with 320x240 pixels image size. Figure 7 shows the evolution of the computation time per image. In [2] we noted that the overall image processing time seemed to evolve approximatively linearly with time: this is confirmed here. However, we can observe that feature extraction and word searching times are higher here: more visual features are found in the larger images used in this experiment, causing more visual words to be added to the visual vocabulary.

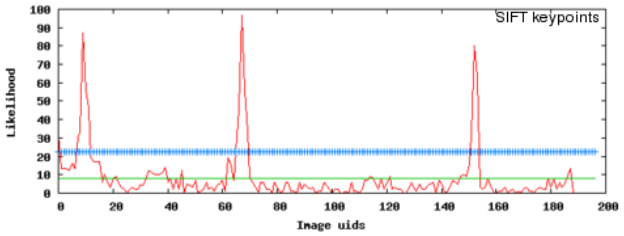

Fig. 6. An example of ambiguous likelihood: this corresponds to a situation where the camera goes back for the fourth time ("d" in figure 5) to an already visited location. From left to right, the peaks correspond to the "a", "b" and "c" passings of the camera at this location in figure 5. Hopefully, further acquired images will help removing the ambiguity.

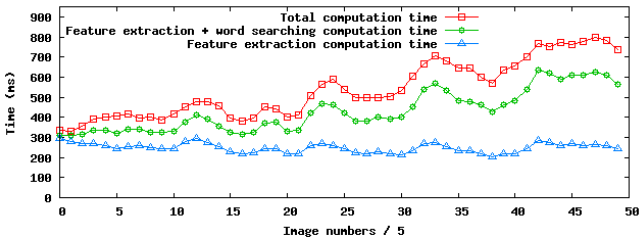

Fig. 7. Evolution of the processing time per image: given is the time needed to extract the features in the images (triangles), to which is added the time required to find the corresponding words in the vocabulary (circles), along with the total computation time per image (squares). The total computation time includes all the processings that lead to the addition or the update of a node in the map (i.e. image processing, word searching, loop-closure probability estimation and multiple-view geometry verification). To enhance readability, computation times have been averaged over 5 images.

VII. Discussion And Future Work

The appearance-based approach to topological SLAM proposed in this paper compares favorably with the methods

Fig. 8. Examples of images composing the sequence. Note the strong similarity between the images from the magenta, red and green areas.

cited in section II because it is the only one to combine real-time performances and fully incremental processing. We adapted our previous work on loop-closure detection [2] to the topological SLAM context, improving some aspects regarding images selection and hypotheses management: similar consecutive images are skipped, the cache size parameter is adaptative and based on the computation of local image similarity, and only those hypotheses that do not correspond to a loop-closure are added to the model as new nodes in the map, scaling better with the number of images.

The results obtained here show the robustness of the loop-closure detection method, making it possible to build consistent topological maps using only a wide-angle monocular camera: the visual bag of words model [4] performed well without having to remove radial distortion from images, making it possible to detect loop-closures even with non-standard perspective cameras. However, monocular vision performs poorly when travelling in an already visited place with significant viewpoint changes (e.g. passing twice in the same location with opposite directions). One solution could be to use local metric information from relative transformations between camera viewpoints as a replacement for the actual time evolution model. This could be done using *visual odometry* [20], or a visual 3D-SLAM algorithm like the one presented in [21]. In a more experimental perspective, we could consider overcoming this difficulty by mounting the camera on a mobile robot that provides odometry measurements. On the one hand, purely vison-based solutions can be easily adapted to any type of mobile robot mounted with a camera, but they require robust feature tracking at frame rate over time, thus failling when the environment is poor. On the other hand, odometry measurements provided by the robot would enable detecting loop-closures even when appearance information is unusable (e.g. when the images do not exhibit salient features), but this would limit the application to robots that can provide such measurements. Furthermore, not only the addition of local metric information to the model could help enhancing loop-closure detection capabilities, it would be necessary for navigation: when planning a path in the map, local metric information encoded in the edges would be required to determine how to travel between nodes.

VIII. Conclusion

In this paper, we have presented an appearance-based approach to address the topological SLAM problem using only a single monocular wide-angle camera. We demonstrated the quality of our method by building a consistent topological map that is coherent with the topology of the environment, even under strong perceptual aliasing conditions. Results were obtained in real-time thanks to incremental processing and should be enhanced in the near future with the addition of odometry measurements.

References

[1] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium* (G. Lakemeyer and B. Nebel, eds.), pp. 1–35, Morgan Kauffman, February 2002.

[2] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Real-time visual loop-closure detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1842–1847, 2008.

[3] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *Conditionally accpeted for publication in IEEE Transactions On Robotics, Special Issue on Visual SLAM*, vol. -, pp. –, 2008.

[4] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *IEEE International Conference on Robotics and Automation*, 2007.

[5] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–777, 2004.

[6] R. Simmons and S. Koeing, "Probabilistic robot navigation in partially observable environments," in *International Joint Conference on Artificial Intelligence*, 1995.

[7] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Hybrid simultaneous localization and map building: a natural integration of topological and metric," *Robotics and Autonomous Systems*, vol. 44, pp. 3–14, 2003.

[8] H. Shatkay and L. Kaelbling, "Learning topological maps with weak local odometric information," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

[9] F. Savelli and B. Kuipers, "Loop-closing and planarity in topological map-building," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[10] A. Ranganathan, E. Menegatti, and F. Dellaert, "Bayesian inference in the space of topological maps," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 92–107, 2006.

[11] O. Booij, B. Terwijn, Z. Zivkovic, and B. Krose, "Navigation using an appearance based topological map," in *IEEE International Conference on Robotics and Automation*, 2007.

[12] 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.

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

[14] Z. Zivkovic, B. Bakker, and B. Krose, "Hierarchical map building using visual landmarks and geometric constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[15] 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.

[16] 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.

[17] D. Lowe, "Distinctive image feature from scale-invariant keypoint," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] 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.

[19] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, 1989.

[20] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, June 2004.

[21] 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.