Probabilistic Appearance-Based Mapping and Localization Using Visual Features

Emilio Garcia-Fidalgo and Alberto Ortiz

Department of Mathematics and Computer Science
University of Balearic Islands, 07122 Palma de Mallorca, Spain
{emilio.garcia,alberto.ortiz}@uib.es

Abstract. An appearance-based approach for visual mapping and localization is proposed in this paper. On the one hand, a new image similarity measure between images based on number of matchings and their associated distances is introduced. On the other hand, to optimize running times, matchings between the current image and previous visited places are determined using an index based on a set of randomized KD-trees. Further, a discrete Bayes filter is used for predicting loop candidates, taking into account the previous relationships between visual locations. The approach has been validated using image sequences from several environments. Whereas most other approaches use omnidirectional cameras, a single-view configuration has been selected for our experiments.

Keywords: Topological Mapping, Localization, Visual Loop-Closure.

1 Introduction

A number of appearance-based localization and mapping solutions have been proposed along the last decade. This approach represents the environment in a topological way as a graph, where each node represents a distinctive visual location visited by the robot and edges indicate connectivities between locations. Using this representation, the loop closure problem can be solved comparing images directly, avoiding the estimation and maintenance of the position of feature landmarks.

Although many works assume the availability of omnidirectional images, and typically operate offline [1,2,3,4], many others make use of cheaper/easier to acquire ordinary imaging configurations [5,6,7,8]. Our approach belongs to this latter class.

Referring to the description of the relevant areas of the environment, the *Bag-of-Words* (BoW) approach [9] has become quite popular relatively recently. Cummins and Newman developed FAB-MAP [5], where a Chow-Liu tree is used for modelling the dependencies between visual words. Angeli *et al.* [6,7] extended the BoW paradigm to incremental conditions and relied on Bayesian filtering to estimate the probability of loop closure. Despite its well-known general performance, the BoW paradigm is affected by perceptual aliasing [10] and usually requires an offline training stage for building the visual vocabulary.

J.M. Sanches, L. Micó, and J.S. Cardoso (Eds.): IbPRIA 2013, LNCS 7887, pp. 277–285, 2013.

Other approaches use global descriptors, such as e.g. Gist [11]. Singh and Kosecka [12] computed Gist descriptors in omnidirectional images of urban environments for detecting loop closures. Bayes filtering is not considered in this work. Liu and Zhang [13] applied Principal Component Analysis (PCA) to Gist descriptors previously to computing the observation likelihood in a particle filter.

Rather than BoW or global descriptors, some authors have used local invariant features directly [8,10,14,15]. Zhang [10] presented a method for selecting a subset of visual features extracted from an image. A location is represented by a set of features that can be matched consecutively in several images. The problem of this approach is that the number of features to manage increases while new images are added, and a linear search for matching becomes intractable. This drawback is overcome in [8] by indexing features through a KD-tree structure. The approach we follow in this paper complies with these guidelines. However, we perform mapping and localization, while they focus their work exclusively on localization.

More specifically, we present an appearance-based framework for visual mapping and localization using local invariant features directly. Given a new image, we assess how different from the current location is using a similarity function, which is based on the number of matchings and their correspondent distances. These matchings can be obtained efficiently building a set of randomized KD-trees [8]. If the image is not similar enough, the probability of loop closure is computed through a Bayesian framework before considering the expansion of the map with a new node. A further condition derived from epipolar geometry is checked next to validate if the image closes a loop or is a new location in the map. A single monocular camera is employed in all our experiments. The rest of the paper is organized as follows: Section 2 explains the basics of our algorithm, Section 3 shows experimental results obtained from different datasets and Section 4 concludes the paper.

2 Algorithm Overview

Storing and handling all the images perceived by a robot during a visual localization and mapping task is typically intractable for real scenarios. To reduce the number of images to consider, a number of them, called keyframes, are carefully selected online. These keyframes represent visually distinct locations of the environment. In our map, each node corresponds to a keyframe, and each keyframe is described by its SURF [16] features.

Our approach considers that the robot is located in the last keyframe acquired while the environment appearance does not change. Given a new image, SURF features are extracted and then a similarity function is evaluated for the image and the current keyframe. If they are similar enough, the robot keeps at the current location. Otherwise, the image needs to be processed in order to determine if a loop has been closed, what requires updating the current location to a previously visited place, or else a new location (keyframe) is defined.

Algorithm 1. Appearance-Based Mapping and Localization

```
/* Variables */
I = {I_0, ..., I_{N-1}} : Sequence of N input images.
G : Graph representing the environment topology.
T : KD-tree for feature indexing.
S_j^i : Similarity between images i and j.
k : Current keyframe index.
F_t : Set of SURF features obtained from image I_t.
c : Candidate keyframe index for closing a loop.
M_j^i : Set of matchings between images i and j.
M_i : Set of matchings between image i and all keyframes in the graph.
E_j^i : Set of matchings surviving the epipolarity constraint-based filter.

k = 0
F_0 = describe(I_0)
addNode(G, 0)
updateTree(T, F_0)
for t = 1 to N − 1 do /* While there are images */
    F_t = describe(I_t)
    M_t = match(F_t, T)
    S_t^k = similarity(M_t^k)
    if S_t^k < MIN_SIM then /* Current view differs from current keyframe */
        c, E_t^c = detectLoopClosure(G, M_t, F_t)
        if numberOfElements(E_t^c) > MIN_MATCHES then /* Loop Closure Detected */
            addLink(G, k, c)
            k = c
        else /* New node (keyframe) is added to the map */
            addNode(G, t)
            addLink(G, k, t)
            updateTree(T, F_t)
            k = t
        end if
    end if
end for
```

An offline version of the approach is outlined as Algorithm 1. In detail, `describe` extracts and describes SURF keypoints from an image, `addNode` updates the topology of the environment adding a new location, `addLink` creates a bidirectional connection between two locations in the map, `updateTree` adds a set of SURF descriptors to the index and trains it and `match` performs a nearest neighbour search for a set of query SURF descriptors and filters the resulting matches using the distance ratio test [17]. `MIN_SIM` is a threshold that indicates the minimum similarity required to keep at the same location and `MIN_MATCHES` is another threshold that represents the minimum number of matchings required to state that the epipolarity condition holds between the current image and a candidate to loop closure. The functions `similarity` and `detectLoopClosure` are key parts of our approach. The following subsections explain each of them in detail.

2.1 Image Similarity

Given two images I_i and I_j and the set of matchings between them, M_j^i, we define a first similarity value between them $S_d\left(I_i, I_j\right)$ as:

$$S_d\left(I_i, I_j\right) = 1 - \frac{\sum_{m \in M_j^i} \mathrm{dist}\left(m\right)}{\#M_j^i \times D_{max}}, \tag{1}$$

where D_{max} is the maximum distance between two SURF descriptors[1], $\#M_j^i$ is the cardinal of set M_j^i and dist is the distance between two matched features. Notice that S_d always takes values between 0 and 1.

An additional similarity function based on the number of matchings $S_m(I_i, I_j)$ is defined as follows:

$$S_m(I_i, I_j) = \frac{\#M_j^i}{\min(N_i, N_j)}, \tag{2}$$

where N_i and N_j are the number of keypoints of respectively I_i and I_j, and min is the minimum operator. We combine (1) and (2) in order to obtain a final similarity value as:

$$S(I_i, I_j) = \alpha\, S_d(I_i, I_j) + (1-\alpha)\, S_m(I_i, I_j), \tag{3}$$

where α is a weighting factor. We have set this value experimentally to 0.65, giving more importance to the term accounting for the matching distances. The closer S is to 1, the more similar these images are. Notice that actually S is independent of the image descriptor: (1) can be particularized for any descriptor recalculating D_{max}.

2.2 Loop Closure Detection

The loop closure condition is assessed whenever the image captured by the robot is not similar enough to the current keyframe; otherwise the map is extended with a new location. A discrete Bayes filter is used to detect loop closure candidates, exploiting the known neighbourhood relationships of the environment. We want to estimate the most likely locations given the current image I_t and the current location.

In our model, the states represent topological locations in the map, while the transition function determines the probability of going from one state to another. Given an image I_t at time t, we denote z_t as the set of SURF descriptors extracted from I_t. These are the observations in our filter. For each state, the probability of being at this location at time t given all previous observations up to time t is:

$$P\left(L_i^t | z_{0:t}\right) = \eta\, P\left(z_t | L_i^t\right)\; P\left(L_i^t | z_{0:t-1}\right), \tag{4}$$

where η is a normalizing factor, $P(z_t|L_i^t)$ is the observation likelihood and $P(L_i^t|z_{0:t-1})$ is the probability distribution after a prediction step. The term $P(z_t|L_i^t)$ models the probability of acquiring certain observation z_t at location L_i at time t. In our model, this probability is directly related to the matchings between the current image and the keyframe at location L_i:

$$P\left(z_t | L_i^t\right) = \eta\, \frac{\#M_t^i}{N_t}, \tag{5}$$

where η is again a normalizing factor, M_t^i represents the matchings between the current image and the keyframe of location L_i and N_t is the number of features

[1] Using a 128-element SURF descriptor, this value has been set to $2\sqrt{128}$.

found in I_t. These matchings can be efficiently obtained using the KD-tree-based index.

The second term in Equation 4 can be written as:

$$P\left(L_i^t|z_{0:t-1}\right) = \sum_{j\in\mathcal{L}} P\left(L_i^t|L_j^{t-1}\right) P\left(L_j^{t-1}|z_{0:t-1}\right) , \tag{6}$$

being $\mathcal{L}$ the set of locations. In this equation, $P\left(L_i^t|L_j^{t-1}\right)$ models the probability of transition from a location L_j to another L_i between two consecutive instants. We want to represent the fact that the closer two locations are in the graph, the higher is the probability of transition. Then, this term is modelled as:

$$P\left(L_i^t|L_j^{t-1}\right) = \eta\, e^{\frac{-\mathrm{dist}(L_i,L_j)}{\sigma^2}} , \tag{7}$$

where dist (L_i, L_j) is the shortest path in number of steps required to go from Li to L_j and σ^2 is the variance of distances for location L_i. The shortest path in the graph is computed using the Dijktra's algorithm. Each time a new location is added to the graph, the state vector of the filter is augmented and the transition matrix $P\left(L_i^t|L_j^{t-1}\right)$ is recomputed.

Once the full posterior distribution has been calculated, best w candidate locations are selected. This set is denoted as $C = \{c_1, \ldots, c_w\}$. For each c_x, an epipolarity constraint between image I_{cx} and current image I_t is assessed in order to validate if they can come from the same view after a camera rotation or translation. Using a RANSAC procedure, matchings not fulfilling the constraint are discarded. We denote the remaining ones as $E_t^{c_x}$. For selecting a final candidate c, a function that involves the probability of being at a certain location and the number of matchings surviving the epipolarity constraint is evaluated:

$$c = \arg\max_{c_i\in C}\{\#E_t^{c_i} \times P\left(L_{c_i}^t|z_{0:t}\right)\} . \tag{8}$$

The final decision about wheter I_t closes a loop with keyframe c or is a new location is taken in accordance to a comparison between E_t^c and a threshold.

3 Experimental Validation

Several experiments have been carried out in order to validate the suitability of our framework for visual localization and mapping tasks. Two datasets from indoor and outdoor environments have been used, in order to verify our method under different environmental conditions. Each dataset contains odometry information, which has been only used for visualization purposes, since our method is purely based on appearance.

3.1 Indoor Environment

The first dataset is publicly available for download[2]. This sequence was recorded at the Computing Science Centre of the University of Alberta. It comprises a total

[2] `http://radish.sourceforge.net`

of 512 images of 640 × 480 pixels, and completes a loop around the third floor. Images were captured with a Dragonfly IEEE1394 digital camera from Point Grey Research mounted in an iRobot Magellan Pro robot. An image was taken after an approximate 15 cm translation or 5 degree rotation, whichever came first.

For the experiment, this sequence was concatenated three times, the first two in forward direction and the last one in reverse direction. The goal was to verify that the map was created during the first loop and remained unaltered during the remaining 1024 frames. That is to say, the idea was that the next loops only assigned images to previously seen places (loop closures) and no new keyframes were added to the map, even in the reverse direction.

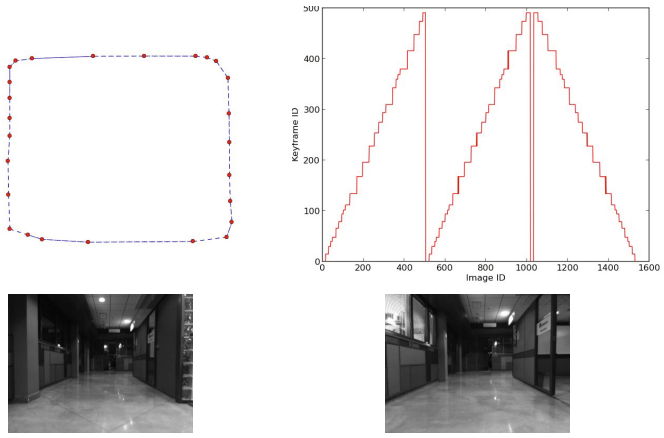

Fig. 1. Results for the indoor dataset. (Top, Left) Final environment topology. Red points indicate selected keyframes in the image sequence and blue lines show links between these keyframes. (Top, Right) Relation between query images (horizontal axis) and its correspondent matched keyframes (vertical axis). Loop closure detected by our framework: image 506 in the sequence (Bottom, Left) closes the loop with image 0 (Bottom, Right).

Results can be seen in Fig. 1. A total of 26 keyframes were selected from the input image sequence. As expected, all keyframes were added during the first loop. Next loops only assigned their images to the previous locations. This can be seen in the upright image. The first loop was closed at image 506 and the second one at frame 1017. The small gap that can be observed in the plot is due to the fact that the sequence does not begin and end exactly at the same place. Images closing the first loop in the sequence are shown in Fig. 1(bottom).

3.2 Outdoor Environment

An outdoor public dataset has also been processed[3]. This dataset is a high-quality stereo sequence with a resolution of 674 × 187 pixels, captured by a

[3] `http://www.cvlibs.net/datasets/karlsruhe_sequences.html`

Pointgrey Flea2 firewire camera and recorded from a moving vehicle around the city of Karlsruhe. Only the left images of the sequence were taken into account. The sequence includes a single loop in an outdoor environment, which we want to detect with our approach.

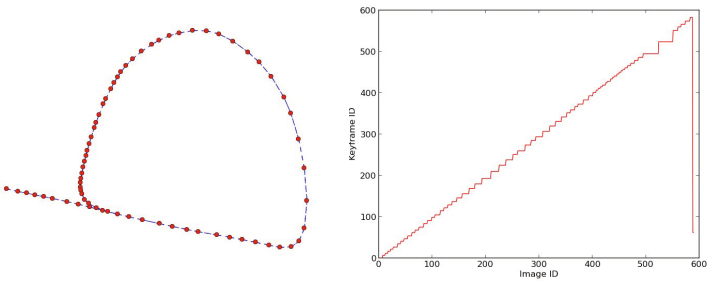

Fig. 2. Results for the outdoor dataset. (Left) Final environment topology. Red points indicate selected keyframes in the image sequence and blue lines show links between these keyframes. (Right) Relation between query images (horizontal axis) and its matched keyframes (vertical axis).

Results are shown in Fig. 2. A total of 70 keyframes were selected from the input sequence. This number usually depends on the frame rate of the input images and the speed of the vehicle. In this case, the velocity is not constant due to traffic issues in the city. This fact makes more difficult finding similar images in consecutive frames, and explains the dispersion of the keyframes in some places. As can be seen in the plot of Fig. 2, a loop closure is detected between image 588 and 62.

3.3 Combining Environments

An experiment was performed concatenating the previous sequences. Using the indoor sequence, an environment map is generated. The images in the outdoor sequence were used as query images. As expected, no loop closures were detected. Instead, the outdoor map was generated starting from the last keyframe detected in the indoor sequence. This experiment shows the behaviour of our framework under the presence of unexplored areas, adding new keyframes to the final graph when the current image is not similar enough to the known visual locations.

3.4 Map Representativeness

One last experiment was performed to assess the quality of the generated maps. The idea was to verify how representative of the environment the maps were independently of the frame rate or the speed of the robot. For each sequence, two loops were concatenated. As well as for the previous experiments, the first loop was used to build the map. Nevertheless, 1 out of N frames were removed

Table 1. Results for the last experiment. The *Total Keyframes* column represent the number of locations created during the first loop and the *Added* column indicates the number of keyframes added during the second loop. See text for further details.

	Indoor Dataset		**Outdoor Dataset**	
Discarded	**Total Keyframess**	**Added**	**Total Keyframes**	**Added**
None (0%)	26	0	70	0
1 out 5 (20%)	26	0	69	0
1 out 4 (25%)	26	0	66	0
1 out 3 (33%)	25	0	66	0
1 out 2 (50%)	25	0	65	0

from the image sequence to assess the sensitivity of the method with regard to the input stream. The images of the second loop were then used to localize the robot in the map. We wanted to prove that no new keyframes were added to the map during the second loop, which means that images had been associated to previous locations irrespective of the frames used to build the map. Results for this experiment are shown in Table 1 for N from 2 to 5. As expected, different numbers of keyframes are found for each N, which modifies the final topology, but does not affect the localization process, since it does not add new locations to the final map in any case.

4 Conclusions and Future Work

In this work, an appearance-based mapping and localization approach using a single monocular camera and SURF features has been presented. A similarity function based on matchings and their distances is used to compare two images. When the current image acquired by the robot changes, a discrete Bayes filter is used to obtain loop closure candidates. The election of a final candidate is directly related with the probability to be at each candidate location and the number of matchings complying with the epipolar constraints. These matchings are used for deciding if this image closes a loop or otherwise it represents a new keyframe. For managing features, an index based on a set of randomized KD-trees is used. Experiments using datasets from different environments have been reported.

Referring to future work: (a) the reactiveness of the system can be improved if several images in an sliding window are processed when deciding if an image closes a loop, avoiding jumps in localization when similar visual locations are very close; (b) matching images using binary descriptors is a solution to explore, since it can improve our approach in computational terms; and (c) the similarity function can be embedded in the Bayes filter in order to obtain loop candidates previously.

Acknowledgments. This work is supported by the European Social Fund through grant FPI11-43123621R (Conselleria d'Educacio, Cultura i Universitats, Govern de les Illes Balears).

References

1. Zivkovic, Z., Bakker, B., Krose, B.: Hierarchical Map Building Using Visual Landmarks and Geometric Constraints. In: International Conference on Intelligent Robots and Systems, pp. 2480–2485 (2005)
2. Ulrich, I., Nourbakhsh, I.: Appearance-Based Place Recognition for Topological Localization. In: International Conference on Robotics and Automation, pp. 1023–1029 (2000)
3. Goedemé, T., Nuttin, M., Tuytelaars, T., Van Gool, L.: Markerless Computer Vision Based Localization using Automatically Generated Topological Maps. In: European Navigation Conference, pp. 235–243 (2004)
4. Sabatta, D.G.: Vision-based Topological Map Building and Localisation using Persistent Features. In: Robotics and Mechatronics Symposium (2008)
5. Cummins, M., Newman, P.: FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. International Journal of Robotics Research 27(6), 647–665 (2008)
6. Angeli, A., Doncieux, S., Meyer, J.A., Filliat, D.: Incremental Vision-Based Topological SLAM. In: International Conference on Intelligent Robots and Systems, pp. 22–26 (2008)
7. Angeli, A., Doncieux, S., Meyer, J.A., Filliat, D.: Real-Time Visual Loop-Closure Detection. In: International Conference on Robotics and Automation, pp. 1842–1847 (2008)
8. Zhang, H.: Indexing Visual Features: Real-Time Loop Closure Detection Using a Tree Structure. In: International Conference on Robotics and Automation, pp. 3613–3618 (2012)
9. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: International Conference on Computer Vision, pp. 1470–1477 (2003)
10. Zhang, H.: BoRF: Loop-Closure Detection with Scale Invariant Visual Features. In: International Conference on Robotics and Automation, pp. 3125–3130 (2011)
11. Oliva, A., Torralba, A.: Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. International Journal of Computer Vision 42(3), 145–175 (2001)
12. Singh, G., Kosecka, J.: Visual Loop Closing using Gist Descriptors in Manhattan World. In: International Conference on Robotics and Automation (2010)
13. Liu, Y., Zhang, H.: Visual Loop Closure Detection with a Compact Image Descriptor. In: International Conference on Intelligent Robots and Systems, pp. 1051–1056 (2012)
14. Kawewong, A., Tongprasit, N., Tungruamsub, S., Hasegawa, O.: Online and Incremental Appearance-Based SLAM in Highly Dynamic Environments. International Journal of Robotics Research 30(1), 33–55 (2011)
15. Zhang, H., Li, B., Yang, D.: Keyframe Detection for Appearance-Based Visual SLAM. In: International Conference on Intelligent Robots and Systems, pp. 2071–2076 (2010)
16. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding 3951(3), 404–417 (2006)
17. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)