LSH for Loop Closing Detection in Underwater Visual SLAM[*]

Francisco Bonin-Font Pep Lluis Negre Carrasco Antoni Burguera Burguera
Gabriel Oliver Codina
Systems, Robotics and Vision Group, University of the Balearic Islands (UIB)
Palma de Mallorca (07122)
francisco.bonin@uib.es,pl.negre@uib.cat,antoni.burguera@uib.es,goliver@uib.es

Abstract

*Effectiveness in loop closing detection is crucial to increase accuracy in SLAM (*Simultaneous Localization and Mapping*) for mobile robots. The most representative approaches to visual loop closing detection are based on feature matching or BOW (*Bag of Words*), being slow and needing a lot of memory resources or a previously defined vocabulary, which complicates and delays the whole process. This paper present a new visual LSH (*Locality Sensitive Hashing*)-based approach for loop closure detection, where images are hashed to accelerate considerably the whole comparison process. The algorithm is applied in AUV (*Autonomous Underwater Vehicles*), in several aquatic scenarios, showing promising results and the validity of this proposal to be applied online.*

1. Introduction

The key of a successful visual SLAM lies in the data association procedure to detect loop closings, which is the difficulty of recognizing already mapped areas. The process of image registration (visual data association) usually entails the matching of visual features between the current and all the previously gathered images (or at least, all images taken inside a region of interest) and the application of any kind of recursive algorithm to eliminate outliers and to calculate the transformation between each image pair [9], [3]. In some cases, this process is too slow to be used in systems that operate at high frame rates. Other authors approach the loop-closing detection problem using BOW in a context of topological maps [2], but they need to define a vocabulary for node identification.

One alternative to accelerate and simplify considerably the loop-closing detection is to run the registration process only with those images that present evident similarities and are most likely to overlap. In this context, LSH (*Locality Sensitive Hashing*) has been used to hash features/descriptors into tables, reducing significantly the matching time [7], [8].

This paper presents a new fast method to find the images of a sequence that have a great probability to present a certain overlap or match with a query image, indicating a possible loop-closing. The algorithm is based on LSH and, in contrast to other LSH-based algorithms that hash descriptors, ours hashes complete images: every candidate image is characterized as a set of just 5 vectors, and 5 indexes (integers) in 5 hash tables. The image registration process is limited, only, to those images very close to the query image, according to their corresponding indexes in the hash tables. This process leads to a drastic reduction of the running time of the loop-closing detection, at every iteration of the SLAM process.

2. Locality Sensitive Hashing

LSH [4] (and in particular E^2LSH [1] which is the version used in this work) is a class of hashing algorithms for probabilistic approximate of the K-NNS (*K-Nearest Neighbor Search*) problem, which means, the ability to return the K points of a database that are closest to a certain query point. The idea behind is to use a set of hashing functions that generate similar values (i.e. collisions) for vectors that are close to each other.

A family of points H with a defined function family g is *locality sensitive* if the probability of collision of points q and v ($q, v \in H$) increases if the distance between them decreases. The problem, indeed, consists on finding all candidate points v_j that have a hight degree of probability to collide with the query point q. For a given point v that $\in H$, the algorithm maps it to a set of buckets $g_i^k(v) = (a_1, ...a_k), \forall g_i \in H$. Afterwards, each bucket is hashed, generating an index $h(a_1, ..., a_k$ in its corresponding hash table: $h(a_1, ..., a_k) = [(\sum_{i=1}^{k} r_i a_i) mod(prime)] mod(tableSize)$, where r_i are random numbers that $\in \Re$ and $tableSize$ is the size of the hash tables (equal to the number of candidate points). Points that present similar indexes in the hash tables are considered to collide.

In our context, points are images, and the problem consists in using E^2LSH to detect all candidate images of a

[*]This work is partially supported by the Spanish Ministry of Economy and Competitiveness under contracts PTA2011-05077 and DPI2011-27977-C03-02, FEDER Funding and by Govern Balear (Ref 71/2011).

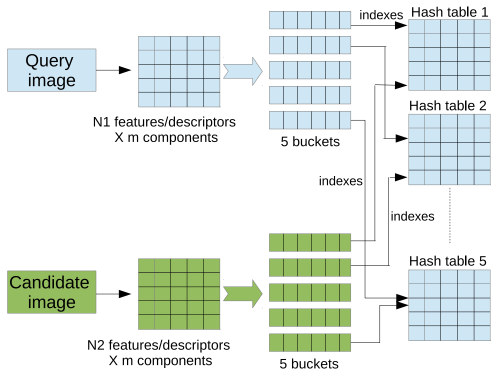

Figure 1. The E_2LSH-based algorithm for loop closing detection

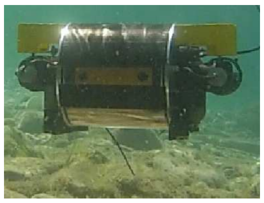

Figure 2. The micro-AUV Fugu-C.

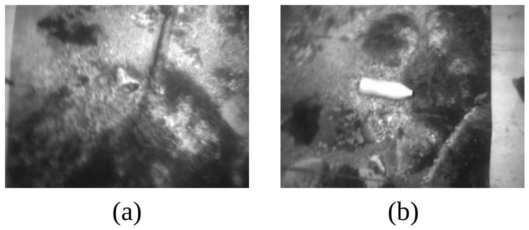

Figure 3. Experiment in the pool: 2 queries.

sequence that are very likely to collide with a query image. In the context of visual SLAM, the query image will be the image grabbed at the current robot pose, and the candidates will be all those images grabbed previously during the vehicle motion.

Five different visual descriptors-dependent bucketing functions have been defined. Let us denote m as the number of visual features in one image, and $g_i^N, i \in \mathbb{N}, 1 \leq i \leq 5$, as each bucketing function, being N the output vector size:

a) Similarly than [6], g_1^N accounts for the number of descriptors located at each region delimited by N randomized m-dimensional hyperplanes that share a common centroid.

b) g_2^N and g_3^N return, respectively, the module and orientation of each local centroid (the centroid of each group of descriptors located at each region) with respect to the global hyperplanes centroid. g_4^N returns the dispersion (variance) of each group of descriptors with respect to their local centroid.

c) g_5^L returns the normalized histogram of the descriptors [5], where each descriptor component is quantized according to L quantization levels.

In summary, visual features are extracted from the query and all candidate images. The set of descriptors of each image gives rise to 5 vectors (buckets) and their corresponding indexes on the respective hash tables. Those candidate images that present indexes in any table close to the indexes of the query are considered to close a loop. The idea is illustrated in figure 1, where the query image collides with the candidate image according to the closeness of their indexes in the fifth hash table.

3. Experimental Results

Some image sequences where grabbed in several underwater scenarios with the low-cost micro-AUV Fugu-C (see fig. 2). Fugu-C was developed at the University of the Balearic Islands, and it is equipped with two stereo rigs (one looking downwards), an IMU and a pressure sensor. For each sequence, several query images have been selected to find the candidates that most likely present overlaps. In all the experiments, the robot navigated at a constant distance to the bottom.

In the present implementation, and according to [1], $prime = 2^{32} - 5$, $1 \leq r_i \leq 2^{29}$. 7 hyperplanes gave the best results in terms of success rate, SIFT features/descriptors were used due to their robustness in front of changes in rotation, illumination conditions and scale, and 2 was the maximum difference between the indexes of the query and the candidates to consider a collision.

Figure 3 shows two queries, captured during one of the experiments in a pool located in the University. The robot moved in a zig-zag trajectory. The pool bottom was covered by a printed digital posters of a real marine context. Each query image was compared with all the images of the sequence (2686 candidates). Figure 4 shows in (a) to (h) eight images that were selected as collisions with figures 3-(a) and 3-(b),

Notice how all of them have a common area with the corresponding query, regardless they present rotations and translations.

Figure 5 shows two queries corresponding to one of the experiments in the sea, in the port of Valldemossa (Mallorca). In this experiment the robot ended at the same starting point identified with an artificial marker. Now, each query image was compared only with all the other images preceding in time (2635 for 5-(a) and 584 for 5-(b), which makes a total of 3219 comparisons), simulating a real SLAM process, where, to search for loop-closings, each current image is compared with the images previously captured during the route. Figure 6 shows in (a) to (h) eight images selected by the algorithm as collisions with figures 5-(a) and 5-(b).

Table 1 shows some data of these collisions: a) the candidate image shown in Figure 6, b) the index difference between the candidate and the query (D.H.I.), c) the hash table where this difference is less than 2, d) the number of feature matchings between the candidate and the query

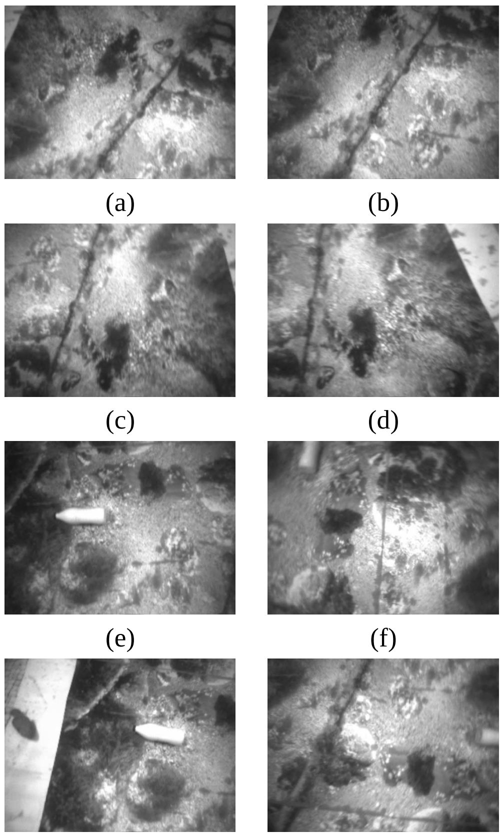

Figure 4. (a) to (d): Collisions with figure 3-(a); (e) to (h): Collisions with figure 3-(b).

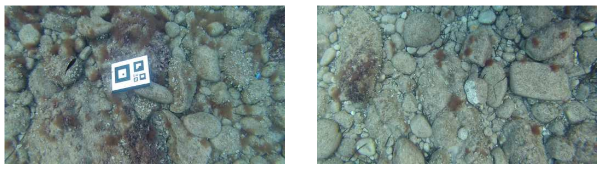

Figure 5. Experiment in the sea: queries.

Candidate	D.H.I.	Table	N_{br} of F.M	% G.O.
query: Fig. 5-(b)				
fig. 6-(e)	1	2	132	0.3515
fig. 6-(f)	2	5	353	0.4628
fig. 6-(g)	1	2	407	0.8016
fig. 6-(h)	2	1,2	516	1
query: Fig. 5-(a)				
fig. 6-(a)	2	1	195	1
fig. 6-(b)	2	4	97	0.3334
fig. 6-(c)	2	5	324	0.4431
fig. 6-(d)	2	5	369	0.4431

Table 1. Hash and Feature Matching Data.

(N_{br} of F.M.) and, e) their geometrical overlap (G.O.).

The geometrical overlap is calculated taking into account: a) that the robot is moving at a constant distance to the bottom and that this distance is known, b) the resolution of the rectangular images and the relation pixel/m, c) the pose and the orientation of the vehicle, calculated using the SLAM algorithm of [3], applied off line. The considerable number of feature matchings and the geometrical overlap show a good correlation with the existence of loop-closings, validating the results given by the LSH procedure.

Table 2 shows the number of candidate images that are close to the query (index difference < 2 in any table) and the number of candidate images that present more than 30 feature matchings (N_{br} of F.M.) with the query. It was experimentally verified that 30 feature matchings was the minimum to compute a reliable registration between two images, using for example, an algorithm based on minimizing the 3D-2D re-projection error. Furthermore, it was also verified experimentally that candidates with less than 30 feature matchings with respect to the query had minimum geometrical overlaps or no overlap at all.

Thereafter, the feature matching process is executed solely between the query image and those images that have a hash matching smaller than 2. All the images that present a hash matching smaller than 2 but less than 30 feature matches with respect to the query are considered as false positives. All the images with a hash matching smaller than two and with more than 30 feature matchings, are considered to be, in principle, as true positives.

It is important to remark that LSH is a method designed to find the nearest neighbor of a query point, which means that, it can miss some points close to the query and it can hit points that are far from the query, even if the number of hash tables is incremented. However, it is always able to retrieve one or several elements very close to the query.

Let us define TFMET (*Total Feature Matching Execution Time*) as the time employed to match features between one query image and all the candidates, and THMET (*Total Hash Matching Execution Time*) as the the time to perform the E^2LSH matching between the query and all the candidates. THMET includes: the time to compute the 5 buckets and their corresponding indexes of all the candidates and the query, plus the time to compare the 5 indexes of each candidate with the query indexes, and the time to compute the feature matchings between the query and those images that have been selected as collisions.

Table 3 shows the TFMET and the THMET for the two query images of the experiment in the pool and for the experiment in the sea. It is evident that the running time em-

Query Image	index diff. < 2	N_{br} of F.M. > 30
Fig. 3-(a)	29	5
Fig. 3-(b)	41	12
Fig. 5-(a)	20	10
Fig. 5-(b)	29	19

Table 2. Feature Matchings vs Hash Matchings.

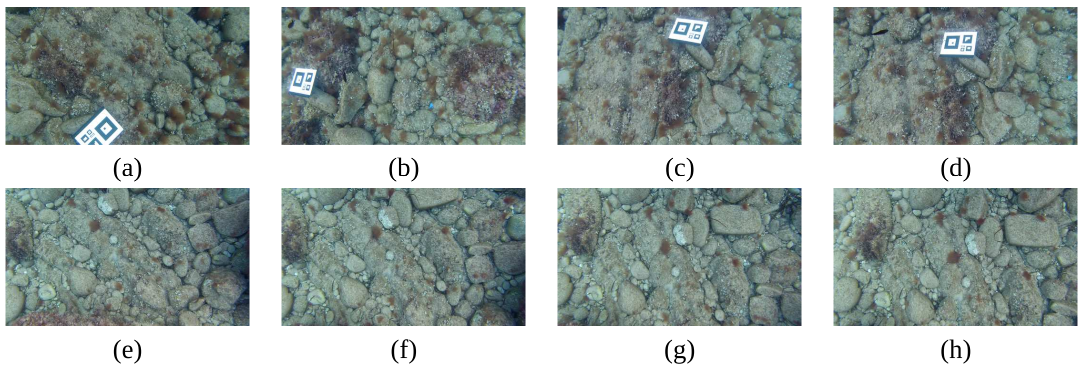

Figure 6. Collisions with figures 5-(a) and 5-(b).

Query Image	TFMET	THMET
Experiment 1, in the pool; fig. 3		
Fig. 3-(a)	98 seconds	16 seconds
Fig. 3-(b)	113 seconds	15 seconds
Experiment 2, in the sea; fig. 5		
Fig.5-(a)	798 seconds	38 seconds
Fig. 5-(b)	191 seconds	9 seconds

Table 3. Total execution times.

ployed by LSH to select the most probable candidates for loop-closing is much lower than the same time but measured when using feature matching.

4. Conclusions and Future Work

This paper introduces the application of E^2LSH for visual loop-closing detection with the main particularity that each candidate image and the query image are characterized by 5 small vectors and 5 indexes which position each image in five different hash tables. Since images are characterized by simple data structures, they need few memory resources and their comparison is much easier and faster. The proposed algorithm applied in diverse visual datasets grabbed from an AUV moving in different aquatic environments shows to last much less than a standard loop closing detector based on feature matching, although this is applied only on images captured at a certain maximum distance from the current robot pose. Images that match according to LSH also have a considerable number of feature matchings and present a remarkable geometrical overlap.

Although the system still presents a considerable percentage of false positives, the remaining true positives are sufficient to determine several loop closures useful for the SLAM.

Forthcoming work includes the refinement of the algorithm to increase the number of true positives and its comparison, in terms of running time, with other systems based LSH or BOW.

References

[1] A. Andoni, M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. *Nearest Neighbor Methods in Learning and Vision: Theory and Practice.*, chapter Locality-Sensitive Hashing Scheme Based on p-Stable Distributions., pages 55–67. The MIT Press., 2006.

[2] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *Robotics, IEEE Transactions on*, 24(5):1027–1037, 2008.

[3] A. Burguera, F. Bonin-Font, and G. Oliver. Towards robust image registration for underwater visual slam. In *Proceedings of the International Conference on Computer Vision, Theory and Applications (VISAPP).*, Lisbon (Portugal), January 2014.

[4] P. Indyk and R. Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. In *Proceedingsof the Symposium on Theory of Computing*, pages 604–613, Newyork (USA), 1998.

[5] V. Monga and B. Evans. Perceptual image hashing via feature points: Performance evaluation and tradeoffs. *IEEE Transactions on Image Processing*, 15(11):3453–3466, November 2006.

[6] S. Roy, X. Zhu, and E.-C. Chang. On preserving robustness-false alarm tradeoff in media hashing. In *Proc. SPIE, Visual Communications and Image Processing 2007*, volume 6508, pages 1–10, 2007.

[7] H. Shahbazi and H. Zhang. Application of locality sensitive hashing to realtime loop closure detection. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems(IROS)*, pages 1228–1233, San Francisco (USA), 2011.

[8] J. Straub, S. Hilsenbeck, G. Schroth, R. huilt, A. Mller, and E. Steinbach. Fast relocalization for visual odometry using binary features. In *20th IEEE International Conference on Image Processing (ICIP)*, pages 2548–2552, Melbourne (Aus), September 2013.

[9] H. Zhang, B. Li, and D. Yang. Keyframe detection for appearance-based visual slam. In *Proceedings of the IEEE IROS*, pages 2071–2076, Taipei, October 2010.