Word Ordering and Document Adjacency for Large Loop Closure Detection in 2D Laser Maps

Jeremie Deray, Joan Solà, and Juan Andrade-Cetto

***Abstract*—We address in this paper the problem of loop closure detection for laser-based simultaneous localization and mapping (SLAM) of very large areas. Consistent with the state of the art, the map is encoded as a graph of poses, and to cope with very large mapping capabilities, loop closures are asserted by comparing the features extracted from a query laser scan against a previously acquired corpus of scan features using a bag-of-words (BoW) scheme.**

Two contributions are here presented. First, to benefit from the graph topology, feature frequency scores in the BoW are computed not only for each individual scan but also from neighboring scans in the SLAM graph. This has the effect of enforcing neighbor relational information during document matching. Secondly, a weak geometric check that takes into account feature ordering and occlusions is introduced that substantially improves loop closure detection performance.

The two contributions are evaluated both separately and jointly on four common SLAM datasets, and are shown to improve the state-of-the-art performance both in terms of precision and recall in most of the cases. Moreover, our current implementation is designed to work at nearly frame rate, allowing loop closure query resolution at nearly 22 Hz for the best case scenario and 2 Hz for the worst case scenario.

***Index Terms*—Localization, SLAM, Range Sensing.**

I. Introduction

FOR several decades SLAM has been an extensively active field of research. Initially developed to give robots autonomy with regards to the navigation task, it has lately focused on other applications beyond robotics, such as augmented reality [17] or medical imagery [9]. Loop closure detection is an essential module of any SLAM system. It is needed to reduce the uncertainty in the estimated map that accumulates during open loop mapping.

Loop closure detection has been tackled with geometric methods (see *e.g.* [11]), or with appearance-based methods. Appearance can be considered either globally [24], [30], [20], [5] or as a set of local distinctive features [27], [22], [18] possibly extracted from different sensor modalities [2]. After the initial work of the computer vision community on the use of bags of words (BoW) for object recognition [29], [3], [23], the SLAM community found in BoW an efficient manner to query large corpus of places visited by a robot while mapping [1], [19], hence its amenity for the solution of the loop closure problem. More recently, state of the art visual SLAM algorithms have relied on BoW for their loop closure and re-localization modules. ORB-SLAM [21] for instance uses DBoW2 [6], whereas LSD-SLAM [4] relies on FAB-MAP [8].

Unlike DBoW2 or FAB-MAP, which use images, our work focuses on the creation of a BoW for the treatment of 2D laser range data. This is motivated by the fact that many robots still use rangefinder sensors for navigation, especially mobile bases for industrial applications. There is little published work on appearance-based place recognition using 2D laser scans, possibly due to the fact that reliable feature detectors/descriptors were developed later than their image based counterparts. The local feature Fast Laser Interest Point Transform (FLIRT) is robust to scale and orientation changes [31] and thus allows a direct application of BoW for the problem of place recognition [32]. As for global descriptors, the Geometrical Landmark Relations (GLARE) [13] encodes the geometrical relations of FLIRT corners in an histogram of relative distance over relative orientation. Extending GLARE, the Geometrical Surface Relations [14] descriptor considers every reading of the 2D laser scan rather than extracted corners. Recently, the Fast Adaptive Laser Keypoint Orientation-invariant (FALKO) [16] has been proposed as a local binary feature, claiming faster and more reliable operation than FLIRT.

Most modern SLAM algorithms such as [4], [21] are composed of three distinct modules:

- *Odometry module* - tracking the sensor/robot motion and selecting key-frames to be added to the pose-graph.
- *Core module* - building the actual pose-graph and eventually solving it.
- *Loop-closure/re-localization module* - detecting loop closures and re-localizing the sensor/robot.

Our work focuses on the third module. It is designed to be agnostic to the SLAM front-end, and limits its interaction with the core module to:

- receiving new key-frames' raw data sensor and its direct adjacency with the previous key-frame.
- informing of loop-closure detections, in the form of pose-graph constraints.

In this paper, we elaborate on the general application of BoW with FLIRT features for loop closure detection for the particular case of 2D laser data. First, by considering feature

Manuscript received: September, 8th, 2016; Revised December, 9th, 2016; Accepted January, 8th, 2017.

This paper was recommended for publication by Editor Cyrill Stachniss upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Spanish Ministry of Economy and Competitiveness project ROBINSTRUCT TIN2014-58178-R, the EU H2020 LOGMATIC project 687534, and the Industrial Doctoral program of the Catalan Agency for Management of University and Research Grants.

The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain jderay,jsola,cetto@iri.upc.edu

Jeremie Deray is also with PAL Robotics, Barcelona, Spain.

Digital Object Identifier (DOI): see top of this page (in the final version of the publisher).

ordering during document retrieval. And second, by updating feature frequency scores considering not only one document at a time but also neighboring documents in the SLAM graph of poses. These two contributions have the effect of substantially improving document retrieval scores compared to previous 2D loop closure implementations.

The rest of the paper is structured as follows. Section II gives a short overview of the BoW scheme and sets the necessary formulations for the introduction of our "weak" geometric check in Section III, which takes into account the ordering of features within a 2D scan. Section IV explains how to compute robust feature frequency scores taking into account adjacent documents within the SLAM graph topology. Section V describes the experimental setup and demonstrates the performances of our contributions. Finally we draw conclusions and propose further work in Section VI.

II. Bags of Words for Place Recognition

In the BoW framework for recognition, the objective is to find the document in a database with the largest similarity score to a query document. For that end, it includes two distinct elements. First, a *vocabulary*, $W = \{w_1, ..., w_k\}$, composed of cluster centers or *words*, w_k, representing the feature space. In our case, each word corresponds to a unique FLIRT feature. This vocabulary of features or words is built offline from a set of maps using hierarchical k-means [23], [7]. The second element consists of a *database* composed of *documents*, $D = \{d_1, ..., d_N\}$, where each document d_j represents the BoW associated to a sensor reading at a known pose of the robot in the current map. That is, the set of FLIRT features in the vocabulary detected in a particular scan and their local coordinates.

The database keeps a record of each word occurrence in every document by means of two frequency scores. The *term frequency* (*tf*) refers to how frequent a single word is within a document, and the *inverse document frequency* (*idf*) refers to how frequent is a single word in the whole database. Given a word w_i in document d_j, these frequencies are computed as follows:

$$tf_{ij} = \frac{n_{ij}}{\sum_i n_{ij}} \ , \tag{1}$$

$$idf_i = \log\left(\frac{|D|}{\sum_j |n_{ij} > 0|}\right) \ , \tag{2}$$

where n_{ij} is the occurrence of the word i in document j, $|D|$ the size of the database and $|n_{ij} > 0|$ evaluates to 1 if w_i occurs in d_j and 0 otherwise. The *weight* of every word w_i in each document d_j is given by its *tf-idf* score, which is computed with

$$x_{ij} = tf_{ij} \cdot idf_i \ . \tag{3}$$

A document is characterized by its *signature*, a vector containing its *tf-idf* weights, $sig_j = [x_{j1}, x_{j2}, ..., x_{jk}]^T$. The document comparison is performed by computing the cosine similarity of their signatures:

$$sim_{lm} = \frac{sig_l^T \, sig_m}{\|sig_l\| \|sig_m\|} \ . \tag{4}$$

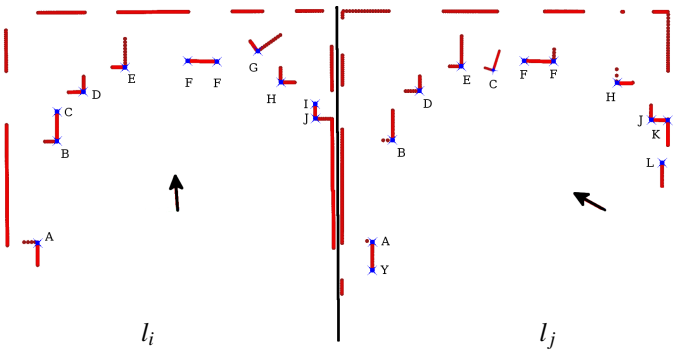


Fig. 1: Environment observed from two different viewpoints. Arrows mark the position and orientation of the sensor.

Given a new sensor reading (a query scan), its feature descriptors are extracted, quantized into words, and its signature compared to those of every document in the database; the N most similar documents are returned by the BoW scheme.

Finally, a consistency check needs to be made to assert which if any of the returned documents is a good match to the query scan. The most simple way to do so is by computing the rigid transform between the matching features in the query scan and those in each of the returned documents. This transformation can be computed using Random Sample Consensus (RANSAC) to be robust to outliers, and the candidate whose inlier set has the smallest residual error below a given threshold is considered a loop closure.

III. Weak Geometric Check

In contrast with other sensing modalities, 2D laser range data present a natural (counter-)clockwise ordering of its local features which can be easily exploited to reinforce the computation of scan similarity. We present in this section some empirical observations regarding this ordering, and how to use these observations in an algorithm that computes the best feature correspondence assignment between two scans. The proposed algorithm produces a similarity score that, when combined with (4), produces a significant improvement in loop closure detection.

A. Observations

The local features extracted from a 2D scan (quantized into words) can be ordered clock-wise in a sequence. The ordering of the features allows to infer some minimal geometric information when comparing scans. This ordering must remain the same for a given scene observed from slightly different viewpoints (see Fig 1). As the viewpoint change increases, features can disappear, shift their location in the sequence or reorder in the following manner: Rotations of the sensor over its main axis induce a pure shift of all the features in the sequence. Forward translations produce, in environments with a predominance of objects in the back-ground, an expansion of the features away from the motion direction, shifting them towards the lateral parts of the scan. In the expansion, new features may appear as details get larger. Backward translations produce the opposite, a contraction of the feature, shifting

them toward the scan center. In the contraction, new features may appear at the extremums and disappear as details get smaller. In all these cases, the ordering of the features through consecutive scans is preserved. In environments with richer perspectives, *i.e.* with objects at significantly different depths, or in the presence of moving objects, nearby obstacles produce occlusions to known features, and can eventually lead to a change in the order of the features in the sequence.

Since we are interested in matching static parts of the environment, and to be robust in the presence of occlusions, our similarity score should concentrate on features that only shift their location in the sequence, and disregard those which change order. With this in mind, we propose an efficient method to align sequences that present these types of variations, encoding the scans with a hidden Markov model, and computing their optimal alignment as a Viterbi path [34].

B. Feature sequence encoding as a hidden Markov model

Feature matching is done directly on words. So, a given descriptor quantized into a particular word w, can only match features also quantized as w and in no case could match another word in the vocabulary. This is exemplified in Fig. 2 (top), which shows the clockwise ordering of words extracted from scans l_i and l_j in Fig. 1 and their correct matches.

This allows us to define the problem of scan alignment as that of finding the path that maximizes the sequence of feature matches in a hidden Markov model. Consider the query laser scan l_i and its extracted words $w_{1i},\ldots,w_{Ni}$ as the set of states S_N in the model. Consider also the candidate match l_j with its words $w_{1j},\ldots,w_{Mj}$ as a set of observations O_M. We can define our HMM such that:

- We have equal initial probabilities $\delta_{s_n} = \frac{1}{N}$.
- The transition from one state to another solely goes forward with respect to the clockwise ordering of the states. Self transitions have a lower probability to enforce the importance of alignments $\phi_{s_n|s_n} = \frac{0.5}{F}$, $\phi_{s_n|s_{n+x}} = \frac{1+\frac{0.5}{F-1}}{F}$, and $\phi_{s_n|s_{n-x}} = 0$, where F is the number of states following the currently evaluated state in the ordered sequence.
- The output probability is defined such that a word mismatch has null probability whereas a word match has equal probability. Hence our emission probabilities are $\theta_{s_n|o_m} = \frac{1}{C}$ for a match, and $\theta_{s_n|o_m} = 0$ for a mismatch, where C is the number of matches of the currently evaluated word.

Fig. 2 (middle) gives an unnormalized representation of the HMM produced by the matching of words in scans l_i and l_j. Black downward pointing arrows indicate feasible transitions, and red upward pointing arrows indicate non-feasible transitions. Each cell is then filled by the product $\phi_{s_n-x|s_n} \cdot \theta_{s_n|o_m}$, where s_n is the currently evaluated state, s_{n-x} is the previous most likely state and $\theta_{s_n|o_m}$ the output probability. Columns are filled recursively based on the previous iteration.

Unlike [12], the HMM is built based on the inner ordering of two independent set of features extracted from raw sensor readings, whereas [12] builds a HMM based on the inner ordering of two independent sequences of key-frames, hence not impacting the frame-to-frame similarity measure. Once the HMM is built, the goal is then to find a sequence of states that maximizes the probability of a path across it.

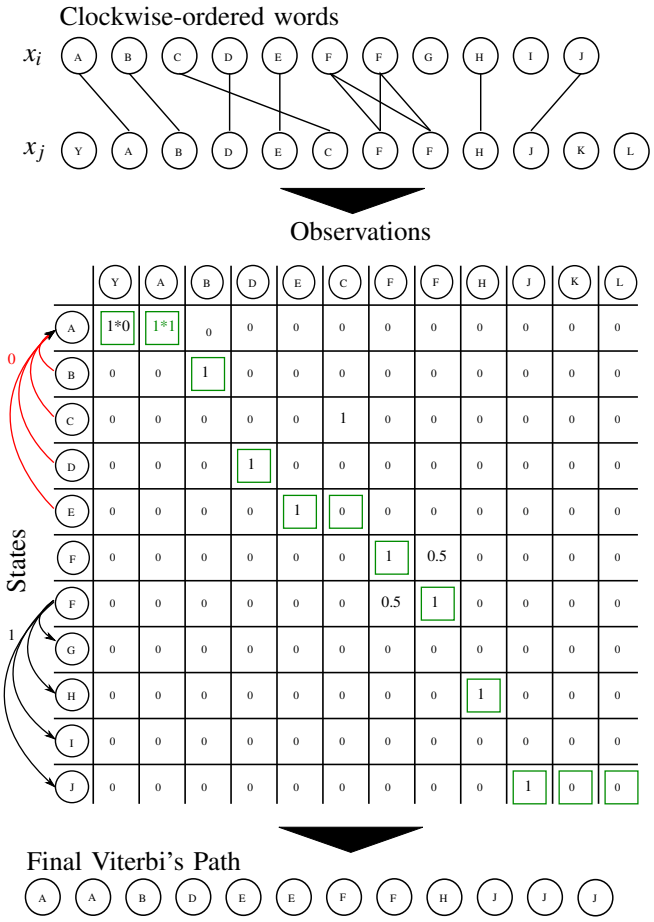


Fig. 2: Top: Clockwise ordered words of two scans and their matches. Middle: The resulting hidden Markov model. Each cell represents the product $\phi_{s_n-x|s_n} \cdot \theta_{s_n|o_m}$ e.g cells *Y-A* & *A-A*. Green squared cells represent the best path across the complete graph. Bottom: The final sequence of states given the observations O_n

C. The Viterbi algorithm

In order to find the most probable path at a reasonable cost in terms of computation, we propose the use of the Viterbi algorithm [34]. This dynamic programing algorithm searches recursively for the most likely sequence of states given a sequence of events, by computing for each observation the partial probability with respect to the previous state that optimally induced the current state. Such sequence is called the Viterbi path. It is commonly used in speech recognition, speech synthesis and decoding [26], [28].

Crossing edges in Fig. 2 (top) highlight mismatches. These might occur either because different features are quantized to the same word (e.g., words C and F), or because the feature is on a moving object. The work in [32] does not discard such mismatches while constructing the offset histogram, and thus

they can not be taken into account to compute a consistent relative transform. Thanks to the constraint of forward state transition, the Viterbi algorithm naturally discards such crossing edges. Note that in our example, crossing edges for the sequence *C-D-E* can be resolved in two different ways, either by removing the match *C* and keeping *D* and *E*, or removing the latter keeping only *C*. The Viterbi algorithm maximizes the sequence of states in the Viterbi path and hence it would prefer, in this case, to keep matches *D* and *E* and discard *C*.

D. Scoring

Once the Viterbi path is obtained, the candidate is scored based on three criteria:

- the number of correct matches that have not been discarded by the Viterbi algorithm,
- the number of sequences of consecutive words that have a correct match, and
- the distribution of matches in the laser scans. The wider the better.

The second point is similar to the concept of phrases in [32], where a phrase represents a sequence of consecutive words, analogous to a n-grams model.

Considering such sequences and weighting them according to their length we can add an extra layer of constraints to our geometric check. These criteria evaluate respectively to: $score_{jk} = \frac{|M|}{|C|}$, where $|M|$ is the number of correct matches and $|C|$ the number of features in the candidate scan; $weight_{jk} = \frac{|CM|}{|C|}$, where $|CM|$ is the number of sequences of consecutive correct matches, e.g., sequences *A-B* & *D-E* in Fig. 1 (bottom); and $ratio_{jk} = \frac{Id_r - Id_l}{|C|}$, where Id_r and Id_l are the indices of the rightmost- and leftmost- correct matches in the Viterbi path, respectively. These three criteria are aggregated into a final geometric score,

$$g_{jk} = \frac{score_{jk} + weight_{jk}}{2} \cdot ratio_{jk} \ . \tag{5}$$

While querying the BoW database, both the *tf-idf*-based similarity in (4) and the geometric score in (5) are computed for each document in the database. The two are then aggregated into a single similarity term,

$$sg_{jk} = sim_{jk} \cdot g_{jk} \ . \tag{6}$$

This aggregated similarity term is then used to rank BoW candidates instead of (4).

IV. Pose-Graph Database Augmentation

In this section we detail our second contribution, which is a topological augmentation of the BoW database. By *augmentation* we refer to the fact of benefiting from common features in adjacent poses in the pose-graph of our map for the computation of the *tf-idf* weights. Since the pose-graph is computed by our SLAM front end, our database augmentation involves no computation overhead.

In pose-graph SLAM, every node holds a robot pose and a sensor measurement, and every edge between two nodes represents a spatial constraint –a relative transform– usually computed from the sensor measurements. The most likely map is obtained by jointly optimizing for all pose constraints in the graph.

A. Topology-based Database Augmentation

Database augmentation taking the form of a similarity graph has been proposed in [25] and [33] for the task of image recognition. Graph edges are created by matching image features and asserting an affine transform between images through RANSAC. Direct edges represent document adjacencies; documents connected to an adjacent document then represents 2-adjacencies, and so on. The set E_j of adjacencies of document d_j is used to emphasize the *tf* weight of the document,

$$m_{ij} = n_{ij} + \sum_{k \in E_j} n_{ik} \ , \tag{7}$$

$$atf_{ij} = \frac{m_{ij}}{\sum_i m_{ij}} \ . \tag{8}$$

These normalized scores (8) constitute the *adjacency tf* used as a direct drop-off replacement for (1) in (3), so that the *tf-idf* weight in (3) becomes

$$x_{ij} = atf_{ij} \cdot idf_{ij} \ . \tag{9}$$

While for object recognition the database augmentation is based on object appearance similarity, in the case of place recognition within a SLAM framework the topological distribution of the places matters. Since an edge in a pose-graph SLAM is computed from sensor readings and represents a spatial constraint, it embeds both the appearance-based similarity required by the BoW scheme (consecutive nodes share some common features) and the topological information that we want to emphasize by the database augmentation.

Whereas object recognition usually considers a pre-trained database for which an offline database augmentation can be computed [25], [33], in the case of place recognition within a SLAM framework the database together with its augmentation are constructed online. Using the SLAM pose graph built online by another module of the SLAM framework allows for a database augmentation at no cost.

Finally, [33] identifies useful features (features belonging to a transformation inlier set) from the document adjacencies and discards the others. Since we build the database online, we keep all of them, as they can become useful later on during mapping.

V. Experiments

We describe in this section our experiments, carried out over four standard 2D laser datasets (three indoors and one outdoor). Table I lists the datasets together with their details. First, we evaluate our contributions both separately and jointly against our own implementation of the classical *tf-idf*-based BoW and publicly available Geometrical FLIRT Phrase algorithm (Gflip) [32] using the same experiment as [32]. Second, we evaluate the robustness with respect to changes in the environment using synthetic data. From here we will use the following aliases for each combination:

Dataset	# Scans	Length (m)	In/Out
FR-079	1464	390.8	Indoor
FR-CLINIC	6917	1437.6	Outdoor
INTEL-LAB	2672	360.7	Indoor
MIT-CSAIL-3rd-FLOOR	1051	382.9	Indoor

TABLE I: Public datasets used [15].

Alias	BoW	"weak" check	adjacent *tf*
tfidf	x	-	-
tfidfgraph	x	-	x
viterbi	x	x	-
vitgraph	x	x	x

TABLE II: Aliases for the different methods compared.

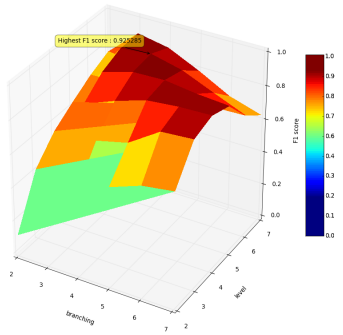


Fig. 3: Variation of the average F1 score with respect to branching factor and tree level.

A. Experiments Setup

Each dataset has been pre-processed by a state of the art SLAM algorithm [10] in order to provide a baseline pose estimation against which to assess if a detected loop-closure is correct or not. Algorithm performance is compared with the following procedure for each dataset used.

Each dataset is first used to train the BoW database. For every scan, its features are extracted, then quantized to obtain their associated BoW and signature.

During query, scans are individually removed from the database when used for query to remove the obvious one-to-one matching.

A consistency check is performed on the top N candidates. The candidate whose inlier set has the the smallest residual error, given that it meets an inliers threshold, is considered a loop closure.

To appraise the correctness of a recognition, the estimated rigid transform is compared to that of the baseline algorithm described above from the pre-processed dataset. It is considered correct if the difference between the estimated pose and that of the baseline lies within 0.5 m and 10 degrees.

The vocabulary was trained from 20.000 scans randomly sampled from a randomly selected subset of datasets among the vast database of the company PAL Robotics. These datasets were recorded in the form of rosbags at different time and places, the great majority being recorded in indoor real case scenario. Thus they are different than those used in this experiment. In average, 17.5 features were extracted per scan.

The tree architecture was chosen empirically using the aforementioned experiment. A total of 49 different trees were trained, varying the branching factor from 2 to 7 and the depth factor from 2 to 7 as well. Evaluating the F1 score (10) averaged over all experiments led to the selection of the average optimal architecture of $k = 4$ and $d = 6$ (Figure 3). As the number of tree leaves is rather small, they are parsed linearly unlike [23] in order to reduce the quantization error.

$$F_1 = 2 \cdot \frac{1}{\frac{1}{precision} + \frac{1}{recall}} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad . \qquad (10)$$

We only consider direct document adjacencies for the BoW database augmentation. Documents with 2-adjacencies did not show any improvements whereas 3-adjacencies decreased the quality of results in an obvious manner as they end up linking observation in the map which seldom shared common features.

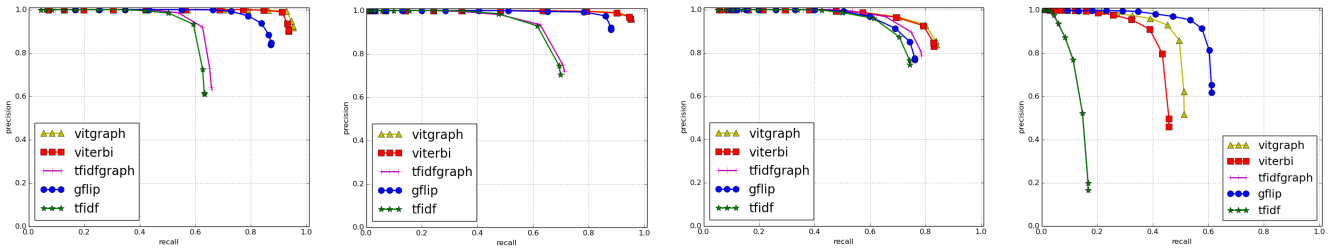
B. Results

1) Recognition Performance: Precision over recall performance results are shown in Fig. 4 for each algorithm, with candidates threshold Top N=20 and varying the inlier threshold. As can be seen in plots (a), (b), and (c), accounting for the indoor datasets, both the *viterbi* and *vitgraph* versions of our algorithm outperform *tfidf* and *gflip* in terms of recall and precision. While *tfidfgraph* only slightly surpasses the performance of *tfidf* in two datasets, it is able to outperform *gflip* for the *MIT-CSAIL-3rd-FLOOR* dataset.

Table III reports statistics for each algorithm at different Top N values. Each row shows the top performance of the algorithm in terms of its F1 score for a fixed Top N while varying the inlier threshold. The best F1 score of each row is highlighted in boldface. The results show that our "weak" geometric assertion allows for a drastic improvement of the recall and precision over the BoW performance and outperforms *gflip* for the three indoor datasets. We attribute this to the fact that the top N returned by the BoW query incorporate "weak" geometrical constraints that are fully leveraged within the purely geometrical consistency check. Adding the BoW database augmentation further improves these results, especially for small numbers of query candidates (lowest values of Top N). However its effect is mitigated when used alone as it decreases the retrieval performance of *tfidf* for the lower Top N values while increasing it for the higher Top N values. Finally, Table III shows that *gflip* requires a higher value of the inlier threshold (*Inl*) to reach its optimal performance, an average increase of two more inliers compared to *tfidf*. Our algorithm reaches it optimal performance for the same *Inl* value as *tfidf* or less.

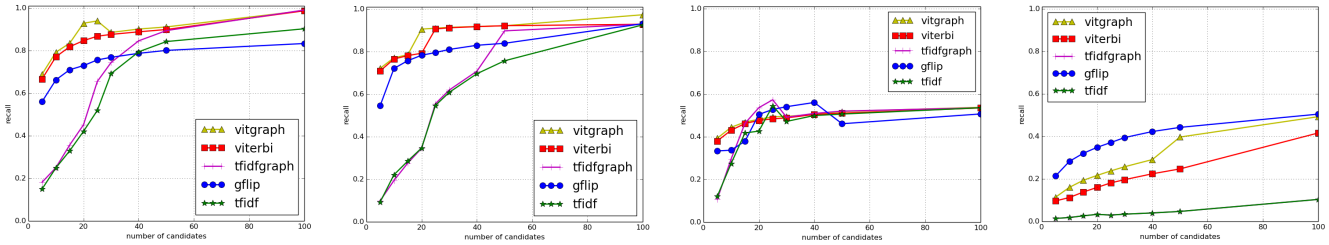
Figure 5 shows the recall over Top N for each algorithm with precision over 99%. Plots (a), (b), and (c) show that both *viterbi* and *vitgraph* outperform *tfidf* and *gflip* recall in the indoor datasets. Moreover, *tfidfgraph* also clearly outperforms *gflip* for the higher Top N cases.

Interestingly, both *viterbi* and *gflip* seem to reach a steady-state in the *FR079* and *INTEL* datasets for the higher Top N cases, while *vitgraph* recall increases. This unveils the limitation of geometrical properties such as the one inferred in [32] or in this paper, and suggests that a closer attention to word-to-word matching should be considered. It also highlights the capability of neighbor documents to empower each other in the



(a) FR079 (b) INTEL (c) MIT-CSAIL-3rd-FLOOR (d) FR-CLINIC

Fig. 4: Recall versus precision for 20 candidates varying the inliers threshold.



(a) FR079 (b) INTEL (c) MIT-CSAIL-3rd-FLOOR (d) FR-CLINIC

Fig. 5: Recall versus number of candidates at 99% precision.

		GFLIP				**TF-IDF**				**TF-IDF+GRAPH**				**VITERBI**				**VITGRAPH**			
Dataset	Top-N	Inl	TP	FP	F1	Inl	TP	FP	F1	Inl	TP	FP	F1	Inl	TP	FP	F1	Inl	TP	FP	F1
FR079	5	9	911	53	772	7	386	178	393	7	324	36	368	7	1008	29	827	7	1033	26	**840**
	10	9	1070	73	843	7	562	36	562	7	565	70	555	7	1153	30	892	7	1185	25	**908**
	20	9	1211	159	884	7	831	60	725	7	877	77	745	7	1278	13	949	7	1300	10	**959**
	50	9	1276	66	932	7	1290	34	947	7	1342	22	971	7	1346	20	973	7	1357	18	**978**
	100	7	1360	78	958	5	1363	16	981	5	1414	28	**994**	7	1382	10	990	7	1383	11	990
INTEL	5	5	1947	72	832	3	897	628	428	3	791	816	370	3	2346	145	910	3	2365	126	**918**
	10	5	2136	73	877	5	1149	124	584	5	1061	139	549	3	2464	95	944	3	2473	88	**947**
	20	5	2292	58	914	5	1637	124	740	5	1666	116	749	3	2518	75	958	3	2525	69	**960**
	50	5	2437	28	950	3	2505	103	950	3	2523	86	957	3	2574	36	976	3	2575	36	**976**
	100	3	2512	54	961	3	2579	39	977	3	2588	29	980	3	2587	31	980	3	2591	26	**981**
MIT-CSAIL-3rd-FLOOR	5	5	555	124	646	5	249	92	361	5	233	95	341	5	644	75	733	5	666	73	**749**
	10	5	679	139	731	5	457	116	567	5	478	126	582	5	747	74	803	5	771	70	**820**
	20	5	773	136	794	5	733	106	781	5	778	90	816	5	823	66	854	5	831	63	**860**
	50	5	858	128	847	5	859	73	872	5	882	62	890	5	865	72	876	3	917	112	**886**
	100	5	905	91	889	5	896	64	896	5	898	61	899	5	894	65	895	5	898	59	**900**
FRCLINIC	5	7	2555	261	**525**	5	361	365	095	5	361	365	095	5	1517	394	344	5	1796	365	396
	10	7	3326	322	**630**	5	629	625	154	5	629	625	154	5	2206	593	454	5	2558	487	514
	20	7	3987	376	**707**	5	1015	922	229	5	1015	922	229	5	3002	765	562	5	3424	559	629
	50	7	4838	426	**795**	5	1774	1260	357	5	1774	1260	357	5	4236	761	712	5	4674	495	774
	100	7	5345	423	843	5	2596	1402	476	5	2596	1402	476	5	5145	688	808	5	5535	405	**861**

TABLE III: First experiment : Algorithms statistics for each dataset at max F1 score. *Top-N*: Number of candidates - *Inl*: Inliers threshold - *TP*: True Positive - *FP*: False Positive - *F1*: F1 score (per mille ‰)

candidate ranking list of the BoW scheme, and thus supports the idea of using a topological augmentation of the BoW database such as the one presented in this paper.

Despite our improvement in recognition over *tfidf*, *gflip* appears to perform better for the outdoor dataset *FRCLINIC*. We conjecture as being due to the fact that the vocabulary tree was trained from scans mostly captured in indoor environments, biasing it's performance towards features encountered indoor.

2) Execution Performance: Table IV shows the average execution time per query with *gflip* and *vitgraph* at 99% precision varying Top N for both the smallest and the largest datasets. We also give the average query time of our *tfidf* implementation as a comparison point for the overhead induced by the weak geometrical check. Experiments were conducted on an Intel Core i7-870 at 2.93 GHz and 8 GiB RAM. Fig. 7 and the accompanying video http://goo.gl/DcCj8q show results of the application of the method during the mapping of a large mall.

These results highlight that the computation overhead of the Viterbi path is linear in the number scans. The complexity of the Viterbi algorithm is about $O(Q \cdot C^2)$ where Q is the number of features of the query scan and C the number of features of the candidate scan. Hence for every query the total computation overhead is $D \cdot O(Q \cdot C^2)$ with D the number of documents in the database. In our experiments, *vitgraph* runs at nearly 2 Hz for the largest dataset and up to 22 Hz for the smallest

one. This suggests that a carefully designed implementation could run at frame rate for the common laser rangefinder — around 10 Hz. However, the target speed should rather be the solver rate (one should allow the solver to finish before issuing a new loop closure event) —around 1 Hz. In such context, the overall impact of improving the precision/recall performance overcomes the penalty in execution time when compared to *e.g.* [32].

Dataset	# Candidates	GFLIP	TF-IDF	VITGRAPH
MIT-CSAIL-3rd-FLOOR	5	0.0038	0.0188	0.0276
	50	0.0125	0.0264	0.0367
	100	0.0233	0.0347	0.0453
FR-CLINIC	5	0.0337	0.1220	0.3369
	50	0.0571	0.1323	0.3545
	100	0.0848	0.1548	0.3686

TABLE IV: Average query time (seconds).

C. Synthetic Obstacles Experiments Setup

The second experiment aims at evaluating the robustness to substantial changes in the environment of the proposed *vitgraph* using synthetic data. For each of the three indoor datasets, an occupancy grid of 0.05 m resolution is generated (Figure 6(a)). First, using the occupancy grid, a set of synthetic scans is generated by the mean of ray-tracing. The set is used to train the BoW databases. Second, virtual obstacles are added (painted) to the occupancy-grid at random position (but on the robot trajectory) and at random scale - 0.05 to 1 m - (Figure 6(b)). Obstacles are of three different types - circles, rectangles and legs (two small circles side-by-side). Then, again, a new set of scans is generated from the new occupancy grid and is then used for querying the BoW database.

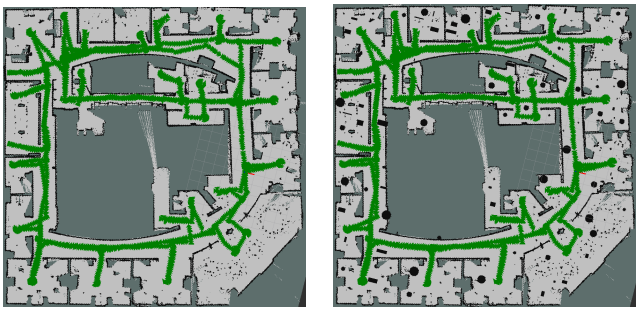
D. Results

The same manner as Table III, Table V reports statistics for both *gflip* and *vitgraph* for a selected subset of Top *N*.

Results show an expected decrease of the recognition performance compared to the previous experiment, but remains good enough to perform loop closures. The magnitude of the descrease is similar for both *vitgraph* and [32]. Since our initial performances are higher than [32], they remain higher in this second experiment. The results show that our proposal is robust too changes in the environment.

		GFLIP				**VITGRAPH**			
Dataset	Top-N	Inl	TP	FP	F1	Inl	TP	FP	F1
FR079	20	9	716	412	570	7	861	369	**659**
	50	9	893	466	651	7	906	412	**670**
INTEL	20	5	1700	482	702	3	2022	503	**780**
	50	5	1935	562	750	3	2062	510	**788**
MIT-CSAIL-	20	5	505	308	555	5	579	166	**662**
3rd-FLOOR	50	5	615	338	628	5	640	168	**690**

TABLE V: Second experiment : Algorithms statistics for each indoor dataset at max F1 score. *Top-N*: Number of candidates - *Inl*: Inliers threshold - *TP*: True Positive - *FP*: False Positive - *F1*: F1 score (per mille ‰)



(a) Intel-lab occupancy grid (b) Intel-lab occupancy grid with painted obstacles

Fig. 6: Intel-lab occupancy grid before and after the addition of virtual obstacles.

VI. Conclusion

We proposed in this paper two contributions to the BoW-based place recognition using 2D laser rangefinder only. First, a "weak" geometrical check that emphasizes BoW candidates which share a static reliable sequence of features with a given query scan. Second, a topological augmentation of the BoW database which permits topological neighbors to empower each other in the BoW candidates ranking list. By using the graph provided by the SLAM algorithm such augmentation comes at no extra computation cost. The addition of both contributions to the classical *tf-idf* scheme outperforms the state-of-the-art loop closure detection methods in terms of recall and precision for three indoor datasets, while drastically improving *tf-idf* results for all datasets. The central idea behind the use of Viterbi - emphasizing candidates by asserting co-occurrent sequences of ordered words - is very generic and could be applied (in a different modality) to image-based BoW. In the same manner, augmenting the BoW database using the graph provided by the SLAM framework is directly usable for image-based BoW. The developped laser-based loop-closure detection framework has been successfully integrated and tested on an industrial robotics plateform as depicted in Figure. 7.

Our ongoing work is to investigate the use of soft clustering for feature quantization, whose probability of belonging to a cluster center can be directly injected in the output probability of the HMM. We expect in this way to reduce further the quantization error. Moreover, the reliability of word matches can be weighted by inferring a score from soft word probabilities. We also plan to extend the developed place-recognition framework to other sensor types with a first aim at cameras.

References

[1] J. Callmer, K. Granström, J. Nieto, and F. Ramos, "Tree of words for visual loop closure detection in urban SLAM," in *Proc. Australasian Conf. Robotics Autom.*, Canberra, 2008, pp. 1–8.

[2] J. Collier, S. Se, and V. Kotamraju, "Multi-sensor appearance-based place recognition," in *Proc. Int. Conf. Comput. and Robot Vis.*, Regina, May 2013, pp. 128–135.

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV Workshop Stat. Learn. Comput. Vis.*, Prague, 2004, pp. 1–22.

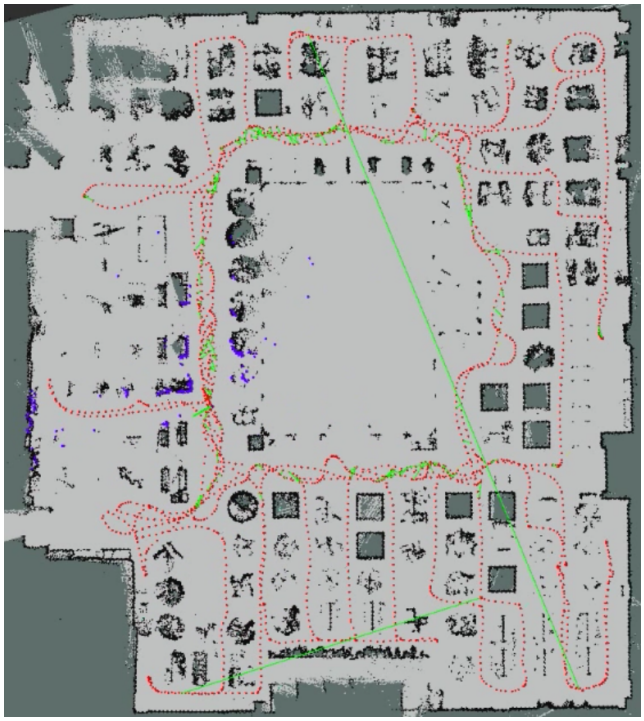


Fig. 7: Occupancy grid of a large mall floor (approx. 2900m^2). Red dots represent robot key frames and green edges loop-closures between them. During mapping, 415 loop closures were detected with only 2 false positives, easily eliminated using distance constraints.

[4] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. 13th Eur. Conf. Comput. Vis*, ser. Lect. Notes Comput. Sci., vol. 8689, Zurich, 2014, pp. 834–849.

[5] H. Friedrich, D. Dederscheck, K. Krajsek, and R. Mester, "View-based robot localization using spherical harmonics: Concept and first experimental results," in *Pattern Recognition*, ser. Lect. Notes Comput. Sci., vol. 4713, Heidelberg, Sep. 2007, pp. 21–31.

[6] D. Galvez-Lopez and J. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[7] A. Gersho and R. M. Gray, *Vector quantization and signal compression*, ser. The Springer International Series in Engineering and Computer Science. Springer, 1992, vol. 159.

[8] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth, "OpenFABMAP: An open source toolbox for appearance-based loop closure detection," in *Proc. IEEE Int. Conf. Robotics Autom.*, Saint Paul, May 2012, pp. 4730–4735.

[9] O. Grasa, E. Bernal, S. Casado, I. Gil, and J. Montiel, "Visual SLAM for handheld monocular endoscope," *IEEE Trans. Med. Imag.*, vol. 33, no. 1, pp. 135–146, 2014.

[10] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proc. IEEE Int. Conf. Robotics Autom.*, Anchorage, May 2010, pp. 273–287.

[11] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb 2007.

[12] P. Hansen and B. Browning, "Visual place recognition using HMM sequence matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, Sep. 2014, pp. 4549–4555.

[13] M. Himstedt, J. Frost, S. Hellbach, H. J. Bohme, and E. Maehle, "Large scale place recognition in 2D LIDAR scans using geometrical landmark relations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, Sep. 2014, pp. 5030–5035.

[14] M. Himstedt and E. Maehle, "Geometry matters: Place recognition in 2D range scans using geometrical surface relations," in *Proc. Eur. Conf. Mobile Robots*, Lincoln, Sep. 2015.

[15] A. Howard and N. Roy, "The robotics data set repository (Radish)," http://radish.sourceforge.net, 2003.

[16] F. Kallasi, D. Lodi Rizzini, and S. Caselli, "Fast keypoint features from laser scanner for robot localization and mapping," *IEEE Robotics Autom. Lett.*, vol. 1, no. 1, pp. 176–183, jan 2016.

[17] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE/ACM Int. Sym. Mixed and Augmented Reality*, Nara, Nov. 2007, pp. 1–10.

[18] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1066–1077, Oct 2008.

[19] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," *Int. J. Robotics Res.*, vol. 29, no. 8, pp. 941–957, 2010.

[20] C. McManus, B. Upcroft, and P. Newmann, "Scene signatures: Localised and point-less features for localisation," in *Robotics: Science and Systems*, Berkeley, Jul. 2014.

[21] R. Mur-Artal, J. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[22] P. Newman and K. L. Ho, "SLAM loop closing with visually salient features," in *Proc. IEEE Int. Conf. Robotics Autom.*, Barcelona, Apr. 2005, pp. 635–642.

[23] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. 20th IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, Jun. 2006, pp. 2161–2168.

[24] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[25] J. Philbin and A. Zisserman, "Object mining using a matching graph on very large image collections," in *Proc. 6th Indian Conf. Comput. Vis. Graphics Image Process.*, Bhubanswar, Dec. 2008, pp. 738–745.

[26] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[27] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. Robotics Res.*, vol. 21, pp. 735–758, 2002.

[28] R. Shinghal and G. T. Toussaint, "Experiments in text recognition with the modified Viterbi algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 2, pp. 184–193, Apr. 1979.

[29] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nice, Oct. 2003, pp. 1470–1477 vol.2.

[30] N. Sünderhauf and P. Protzel, "BRIEF-Gist - closing the loop by simple means," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, Sep. 2011, pp. 1234–1241.

[31] G. Tipaldi and K. Arras, "FLIRT - interest regions for 2D range data," in *Proc. IEEE Int. Conf. Robotics Autom.*, Anchorage, May 2010, pp. 3616–3622.

[32] G. D. Tipaldi, L. Spinello, and W. Burgard, "Geometrical FLIRT phrases for large scale place recognition in 2D range data," in *Proc. IEEE Int. Conf. Robotics Autom.*, Karlsruhe, May 2013, pp. 2693–2698.

[33] P. Turcot and D. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problems," in *Proc. ICCV Workshop Emergent Issues Large Amount. Vis. Data*, Kyoto, Oct. 2009, pp. 2109–2116.

[34] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.