Real-time Wide-baseline Place Recognition using Depth Completion

Originally published in:
IEEE Robotics and Automation Letters 4(2), https://doi.org/10.1109/lra.2019.2895826

Funding acknowledgement:
157585 - Collaborative vision-based perception for teams of (aerial) robots (SNF)

Real-time Wide-baseline Place Recognition using Depth Completion

Fabiola Maffra, Lucas Teixeira, Zetao Chen and Margarita Chli

***Abstract*—Place recognition is an essential capability for robotic autonomy. While ground robots observe the world from generally similar viewpoints over repeated visits, other robots, such as small aircraft, experience far more different viewpoints, requiring place recognition for images captured from very wide baselines. While traditional feature-based methods fail dramatically under extreme viewpoint changes, deep learning approaches demand heavy runtime processing. Driven by the need for cheaper alternatives able to run on computationally restricted platforms, such as small aircraft, this work proposes a novel real-time pipeline employing depth-completion on sparse feature maps that are anyway computed during robot localization and mapping, to enable place recognition at extreme viewpoint changes. The proposed approach demonstrates unprecedented precision-recall rates on challenging benchmarking and own synthetic and real datasets with up to 45° difference in viewpoints. In particular, our synthetic datasets are, to the best of our knowledge, the first to isolate the challenge of viewpoint changes for place recognition, addressing a crucial gap in the literature. All of the new datasets are publicly available to aid benchmarking.**

***Index Terms*—Aerial Systems: Perception and Autonomy, Visual-Based Navigation, SLAM, Localization, Recognition**

I. Introduction

SIMULTANEOUS Localization And Mapping (SLAM) refers to the process of building a map of the robot's workspace, while keeping track of its pose within it. In cases where SLAM estimation fails or drifts, it is essential to determine whether the robot has visited the current location in a previous occasion triggering relocalization. While originating from the problem of loop-closure detection, Place Recognition is also essential in multi-robot tasks, informing each robot where the others are. In scenarios, where multiple robots work in collaboration to carry out a given task, the scene is usually observed from very different viewpoints and assessing scene similarity from images captured under such wide baselines (e.g. ground to air) is known to be a very challenging task.

Place Recognition is commonly addressed using visual cues. It was the advent of real-time monocular systems for SLAM that paved the way towards the use of SLAM onboard small UAVs (Unmanned Aerial Vehicles). While many successful strategies for performing Place Recognition using range sensors have been proposed in the literature [1], these sensors are usually heavy and power greedy, severely reducing the endurance of small UAVs or even exceeding their payload capacity. For UAVs restricted to small payloads and as a result, limited computational capabilities, the employment of vision-based approaches comes as a natural choice for automating their navigation.

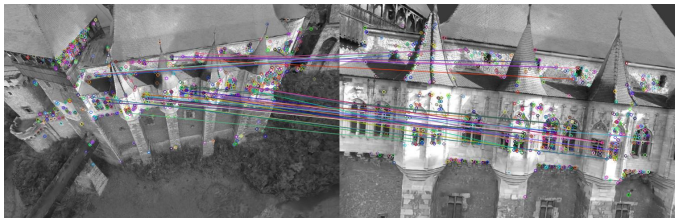


Fig. 1. A loop in the synthetic Corvin dataset correctly detected by the proposed approach, despite the large change in viewpoint (45°).

Motivated by the challenges of place recognition from aerial imagery, in this paper we specifically study the problem of Place Recognition under extreme changes in viewpoint. While still addressing common challenges in Place Recognition, such as illumination and situational changes, here, we push our method to the limits by testing on dramatic changes in viewpoint and showing that feature-based methods can still play a key role, enabling practical use in many common scenarios, such as 3D reconstruction of archaeological sites and collaborative multi-robot SLAM. Fig.1 shows a successful loop-closure detected using the proposed approach designed to address extreme changes in viewpoint.

The main contributions of this paper are:

- a novel real-time pipeline for loop-closure detection that employs depth-completion to enable feature-based matching between images captured from very different viewpoints. As such, this paper advocates and demonstrates that feature-based approaches are still useful for matching images across very wide baselines, while maintaining computation affordable for autonomous UAV navigation.
- new photo-realistic datasets exhibiting dramatic viewpoint changes in simulation, isolating for the first time the problem of viewpoint changes in Place Recognition from other challenges, such as scale variance, dynamicity of the scene, and illumination. In addition to these synthetic datasets, we also release real datasets capturing similarly large viewpoints using aerial and ground footage.

Manuscript received: September, 10, 2018; Revised November, 26, 2018; Accepted January, 9, 2019.

This paper was recommended for publication by Editor Jonathan Roberts upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Swiss National Science Foundation (SNSF, Agreement no. PP00P2_157585) and NCCR Robotics.

Authors are with the Vision for Robotics Lab, ETH Zurich, Zurich 8092, Switzerland (e-mail: fmaffra@mavt.ethz.ch; lteixeira@mavt.ethz.ch; chenze@ethz.ch; chlim@ethz.ch).

This letter has supplementary downloadable material (datasets) available at www.v4rl.ethz.ch/research/datasets-code.html. A video with results of the proposed algorithm is available at https://youtu.be/iwxbxAsCJbM.

Digital Object Identifier (DOI): see top of this page.

II. Related Work

Most recent state-of-the-art SLAM systems, such as ORB-SLAM [2], employ image retrieval techniques to enable large-scale place recognition. A Bag-of-Words (BoW) approach combined with an inverted-file-index [3] or its more compact representations, such as Fisher Vectors [4] or Vectors of Locally Aggregated Descriptors (VLAD) [5] are usually applied to efficiently search for loop-closure candidates in a database of images containing all previous experiences of the robot. The widely known BoW approach relies on discretizing the feature-descriptors' space to build a dictionary of visual words that are then used to describe new images by converting locally invariant feature-descriptors into a BoW representation. Although several well-performing feature-based algorithms have been proposed for place recognition [6], [7], the extraction of unique and repeatably recognizable features has proven to be far from trivial [8]. In fact, extreme changes in appearance can pose a significant challenge for feature-based approaches. As a result, approaches using range sensors [1] or structural descriptors [9] have been proposed exploiting the fact that geometry offers better invariance to viewpoint changes when occlusion is not present.

Current feature-based BoW approaches try to circumvent major changes in appearance by using high-quality feature detectors and descriptors, such as SIFT [10] and SURF [11]. However, these features still fail when large changes in viewpoint occur, and are typically too expensive to be employed in real-time applications, for example onboard a small UAV. Affine SIFT features [12] handle large image distortions by generating multiple affine transformations of an image before applying traditional SIFT. However, their increased invariance comes at a prohibitively high computational cost of two orders of magnitude slower than SIFT. By generating a mesh of the current robot's surroundings, the work in [13], makes use of a 3D map provided by SLAM and identifies the most prominent plane in each image computing only one affine transformation, as orthophoto. This enables the creation of a single view of the scene, while using a computationally cheap binary descriptor and avoiding the need for computing multiple transformations of the same image.

While purely 2D image-based approaches can offer the ability to localize images even if local feature matching fails, these methods are usually considered unsuitable for accurate visual localization. 3D structure-based approaches offer more precise pose estimation, becoming a natural choice for visual place recognition methods, which require the recovery of the 6-DoF camera pose. Sattler et al. [14] combine both methods by querying an image database to retrieve a set of related images depicting the same place and performing a small-scale Structure From Motion (SFM) to obtain a local 3D reconstruction around a query image. 3D structure-based techniques assume that the scene is represented by a 3D model, usually obtained from SFM [15] or SLAM [16], and the camera pose can be obtained using a PnP solver [17] in a RANSAC scheme [18]. Another widely used approach is to use LIDAR sensors to obtain the 3D structure of the environment in very fine resolution. SegMatch [1], for example, performs place recognition using 3D laser data using the concept of segment matching. Despite the reduced amount of noise, these maps are usually sparser than maps obtained using vision-based approaches, and as already mentioned, range sensors are still too heavy and often too power-consuming to be carried on a small UAV.

More recently, Convolutional Neural Networks (CNNs) have been successfully demonstrated to extract robust feature descriptors for place recognition [19] or even to regress a 6-DoF pose directly from images [20]. While shown to produce impressive results even under extreme changes in appearance, deep learning techniques, however, usually rely on powerful GPUs, rendering them too computationally expensive to run onboard a small aircraft. Besides this, they also rely on very large, annotated datasets, which are very hard to obtain.

III. Methodology

In the proposed Place Recognition pipeline, illustrated in Fig. 2, we assume that vision-based SLAM running onboard the robot provides, for each image entering the pipeline, a sparse 3D map of the location and, optionally, its 2D features (i.e. keypoints and descriptors). When a new image arrives, a map densification step generates a denser 3D map from the sparse 3D map provided by SLAM using a depth completion approach. New image features can be detected for Place Recognition if the user desires different features from the ones used in SLAM. All features get converted into a BoW representation in order to search for loop-closure candidates that have similar appearance to the query image. A candidate filtering step refines and removes erroneous loop-closure candidates by exploiting covisibility information captured by SLAM. Any remaining loop-closure candidates proceed to a geometric check, where geometric compatibility between the query and each candidate is evaluated by using all their 2D features and their denser 3D maps. If the geometric check succeeds, a loop-closure is deemed as detected and the pipeline returns the loop-closure match with the most keypoints in agreement with the query.

Sections III-A, III-B and III-D describe briefly the main steps of the pipeline already introduced in [21], while Section III-C focuses on the main novelty of this paper, the use of depth-completion to improve the establishment of 3D-3D and 3D-2D correspondences during geometric checks, which is the key component enabling feature-based matching across images of very different viewpoints.

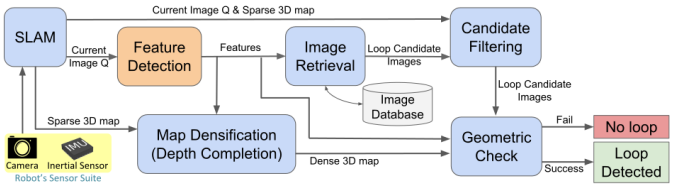


Fig. 2. The proposed pipeline for Place Recognition employing depth completion with appearance and geometric checks to determine whether the current image Q forms a loop closure with an image in the database containing past robot experiences.

A. Loop-Closure Candidates Retrieval

Following the approach suggested by Galvez and Tardos [7], a hierarchical BoW visual vocabulary is formed by discretizing the feature-descriptors' space into a set of visual words. When a new query image arrives, local features, such as BRISK or SURF, are extracted and converted to a BoW representation, used to retrieve a set of database images similar to the current image. The BoW descriptor is scored based on its distance to database entries, using a 'term frequency-inverse document frequency' (tf-idf) weighting scheme [6] to suppress commonly occurring words.

The decision of the feature detector and descriptor to be used is left open in the proposed framework, as this decision affects the trade-off between precision and recall. While SIFT [10] and SURF [11] features can be used in the pipeline, for example, their bigger accuracy comes at the cost of longer run-times, when compared to binary features, such as BRISK [22] and ORB [23]. As BRISK features require low computational cost, being more suitable for UAV navigation, here we use BRISK for our experiments.

B. Candidate Filtering

As geometric checks are usually expensive, here covisibility information captured by SLAM is firstly used to refine and remove erroneous loop-closure candidates suggested by the BoW descriptors when querying the image database. Following the same approach as in [2], the proposed pipeline implements a covisibility graph, where each node is a frame and an edge between two nodes exists if they share enough observations of the same 3D points in the SLAM map. As a simplification, in case of loop-closure detection the covisibility graph is not updated, keeping only covisibility information at the frames' neighbourhood. At first, the minimum score S_{min} between the query and its neighbours in the covisibility graph is recorded, and any candidate which scores lower than 75% of S_{min} is excluded from the list of candidates. While [2] removes all candidates lower than S_{min} avoiding false-positive at all costs, here we employ a more permissive filter in order to recover candidate images taken from more distinct viewpoints subject to strict checks later on. As many overlapping frames exist, when querying the database, many images will exhibit a high score when compared to the query image. These overlapping images are taken into account by summing up the scores of the images that are neighbours in the covisibility graph. Any loop-closure candidate scoring higher than 75% of the best score will proceed to the next step. A candidate loop image is accepted if three consecutive loop candidates are consistent. Two frames are defined to be covisibility-consistent if they share at least one frame among their covisibility neighbours. More details about this approach can be found in [2].

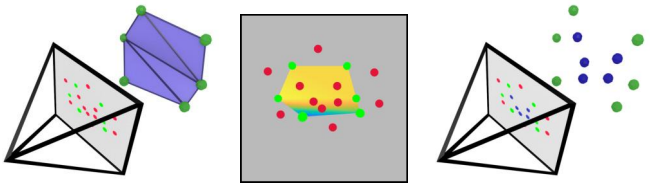
C. Map Densification using Depth Completion

During the geometric check, geometric consistency between the query and the candidate is evaluated by computing the query's pose in the candidate's coordinate frame. This procedure requires the establishment of 3D-3D or 3D-2D correspondences between the query-candidate pair. Assuming that the scene is represented by a 3D map, and each 3D point is associated with one or more local descriptors in the image space, 3D-3D and 3D-2D correspondences are obtained via descriptors matching in the image space. However, under extreme viewpoint changes, feature-based image matching is strongly affected by affine distortions and occlusions, resulting in a reduced number of correspondences between the query's and the candidate's keypoints. Besides this, it must be noted that only keypoints successfully tracked by SLAM have a 3D landmark associated with them. As such, only a small number of keypoints carrying 3D information arrives to the geometric check. By using a depth completion for map densification, interpolated 3D landmarks can be estimated for the 2D keypoints that have no depth-estimates yet, improving the establishment of 3D-3D and 3D-2D correspondences for images captured across very wide baselines.

Fig. 3 illustrates the map densification pipeline, which consists of a depth completion step, shown in Figs. 3a-3b, followed by the creation of the interpolated 3D landmarks, illustrated in Fig. 3c. Our map-densification algorithm, takes as input the camera pose, the 3D landmarks visible by this camera (dark green) and the 2D keypoints (red), for which we want to calculate an interpolated landmark. A dense mesh of the 3D landmarks is first computed (in purple in Fig. 3a) using the open-source mesh-generation pipeline of [24]. A depth image, of the same size as the camera image is obtained by rendering this mesh into the image plane and extracting the depth-buffer of the render engine, as shown in Fig. 3b, illustrating the 2D keypoints in red and the projections of the 3D landmarks in the image in green. Any 2D keypoints lying over a pixel with depth information, have their corresponding 3D landmarks estimated, in camera coordinates, by using the pixel's coordinates and the depth value on that pixel, using Equation (1). Any remaining 2D keypoints cannot have a 3D landmark established. Fig. 3c shows, in blue, the new, interpolated 3D landmarks added to create a denser map of the scene.

$$P_c = (X, Y, Z) = \left(\frac{(u - u_0) * d}{f_u}, \frac{(v - v_0) * d}{f_v}, d \right), \quad (1)$$

where (u, v) is the position of the detected keypoint, u_0 and v_0 are the pixel coordinates of the camera's optical center, f_u and f_v are the focal length in u- and v-direction, respectively, and d is the depth provided by the mesh at the pixel (u, v).



(a) (b) (c)

Fig. 3. The map-densification process: the green 3D landmarks are used to estimate the depth of the red 2D keypoints by creating a mesh (in purple) in (a), and projecting it in a depth image visible in (b). This results to the additional blue 3D landmarks in (c).

While it is possible to extract all the keypoints needed for

Place Recognition during SLAM, only a reduced number of them, represented in bright green in Fig. 3, can be tracked in order to keep its real-time performance. With OKVIS [25] (the SLAM system used in our experiments), for example, we can usually track about 400 landmarks while maintaining real-time performance, however, about 1000 keypoints were used here for Place Recognition. As such, the map-densification approach focuses in estimating the 3D landmarks for the keypoints that were ignored or not successfully tracked by SLAM. However, if the type of keypoints and descriptors used for Place Recognition is different from the one used during SLAM, new features need to be detected. In this case, the map densification will try to estimate a 3D landmark for every newly detected keypoint. One advantage of the latter case is a better decoupling between the SLAM method and Place Recognition.

Another advantage of the proposed map-densification approach is that it can handle arbitrarily sparse maps, which can contain certain amount of noise, while traditional depth-completion algorithms, such as [26], rely on good quality and not very sparse depth images as input in order to create a dense depth image. Here, we opted to use a mesh-based approach to create a dense depth image out of the 3D landmarks provided by SLAM. A higher quality representation of the scene is then obtained using the mesh generation pipeline of [24], which applies a Delaunay triangulation followed by an outlier removal to create a 3D triangle mesh out of the 3D landmarks provided by SLAM. Assuming local planarity among neighbouring vertices of the mesh, outlier removal is performed by comparing the value of a vertex with the centroid of the vertex's neighbourhood. In case of a large disagreement the vertex is eliminated. This approach prioritizes high-quality depth estimations instead of a full representation of the mesh, and holes can exist at points with a high local depth uncertainty. While very efficient in removing outliers, the use of a sparse 3D map together with the local planarity assumption create a smooth mesh of the environment, eliminating details in small areas with a large depth variation. However, as demonstrated in [13] and [24], this approach was already proven to work well in man-made environments, where locally planar structures are usually present. Besides this, this mesh generation approach takes about 7 ms per frame to create a 3D mesh out of the 3D landmarks, rendering it suitable for real-time applications.

D. Geometric Check

The BoW approach does not use any geometric information for image retrieval, accepting two images as a match if they present a similar collection of words. As geometry was shown to play a key role in identifying true loop-closures, here we employ the geometric checks proposed in our previous work [21]. Geometric consistency between a query-candidate pair is evaluated by computing the query's pose in the candidate's coordinate frame. If a pose P_Q^C can be successfully estimated, the candidate is accepted as a loop-closure for the query.

When testing for geometric consistency, we first search for feature correspondences between the query Q and a candidate C using only keypoints with associated 3D landmarks. If enough 3D correspondences are found, we attempt to estimate a similarity transformation (i.e. translation, rotation and scale) between the query and the candidate using Horn's method [25] in a RANSAC scheme [18]. If a transformation that satisfies a minimum threshold on the average reprojection error is found, the candidate is accepted as a loop-closure for the query. In this case, P_Q^C can be easily recovered by multiplying the candidate's pose on his own coordinate system by the similarity transformation. However, if a transformation cannot be estimated or not enough 3D-3D matches can be found, the set of correspondences to be considered is expanded by searching for feature correspondences between the candidate's keypoints with 3D landmark associated and all the keypoints in the query Q. If enough 3D-2D matches are found, we attempt to directly estimate the pose P_Q^C using the 3D-2D matches [17]. If this succeeds, a loop-closure is deemed as detected. We repeat this process to all loop-closure candidates and select the candidate match with biggest number of inliers.

IV. Datasets

In this work, two types of datasets are used to evaluate the proposed method. To isolate the problem of viewpoint changes in place recognition, while keeping full control of the test conditions, we set up a photo-realistic simulation. Finally, tests are conducted also in real conditions, using datasets recorded with hand-held cameras and aerial robots, exhibiting very different viewpoints, such as air-ground matching.

A. Photo-realistic Synthetic Datasets

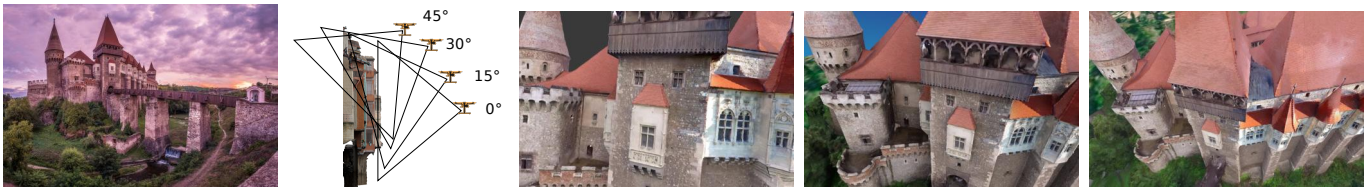
Large scale outdoor experiments using real robots are the best way to validate a place recognition algorithm. However, such data lacks not only the ground-truth of the robot's poses but also the 3D model of the environment. Traditional methods of constructing ground-truth poses, such as with GPS or laser tracking, estimate the robot's position with an accuracy of several centimetres at best, but can also be up to a few meters inaccurate. Even more problematic is the orientation estimation of the camera that is usually unknown or only roughly estimated in post-processing. In order to guarantee good ground-truth for the loop-closures, some datasets are manually annotated, such as in [21]. By making use of synthetic datasets, ground-truth information is easily obtained, allowing quantitative evaluation of the method by automatically estimating the ground-truth.

Fig. 4. The Left image shows the result of our simulation and the right is an actual picture taken from the same place with a consumer camera.

In order to create our synthetic datasets, we use 3D models obtained by photogrammetric reconstruction. We create UAV

(a) L'Agout in Castres (b) 0°view (c) 15°view (d) 30°view (e) 45°view

Fig. 5. L'Agout dataset: 3D photometric reconstruction of medieval houses. In (a), a picture of the location shows houses of about 15m height by 100m width in total and a depth variation of 3m among the facades. In (b)-(e) are example images from L'Agout dataset at 0°, 15°, 30° and 45°, respectively.



(a) Corvin Castle (b) angles side view (c) 0°view (d) 30°view (e) 45°view

Fig. 6. Corvin dataset: 3D photometric reconstruction of Corvin Castle (a). In (b), the different viewpoints used to record the synthetic datasets, and in (c), (d) and (e), example images from Corvin dataset at 0°, 30° and 45°, respectively, are shown.

trajectories using the Rotors UAV physical simulator [27] and the RGBD images are produced by the Blender render engine. Fig. 4 shows that our simulation produces images that are very similar to the real ones. This approach on dataset generation produces visual-inertial measurements that reproduce the Skybotix VI-Sensor with resolution of 752×480 pixels, the same resolution as in the outdoor real datasets. Defining as loop a pair of images with more than 50% of overlap and using the ground-truth poses provided by the physical simulator, we were able to easily distinguish (and annotate) the image-pairs that constitute loops.

Namely, we construct the following datasets:

The L'Agout 0° & 15° & 30° & 45° dataset was produced using aerial pictures of "Maisons sur l'Agout" visible in Fig. 5, depicting medieval houses with balconies over the river Agout. We produce 4 sequences of 100 meters with a laterally moving drone carrying a camera facing the houses at 0° (i.e. pointing forwards), 15° from the horizon, 30°, and 45° as shown in Fig. 6b. It is important to highlight that the position of the drone was chosen in a way that the camera frustum is completely filled by the buildings in order to guarantee that the only difference between these sequences happens in the viewpoint, without any changes in scale.

The Corvin 0° & 30° & 45° dataset was produced using aerial footage of the Corvin Castle visible in Figs. 4 and 6. We produced 3 sequences at 0°, 30°, and 45°, while doing a 300-meter circular flight around the castle. These sequences capture a scene composed of a large range of different depths.

B. Outdoor Real Datasets

While we focus our real world experiments on publicly available datasets, we also construct a new air-ground dataset, which we make publicly available together with the new synthetic datasets. All real datasets used in this paper were recorded using a Skybotix VI-Sensor, using only one camera and one IMU in a hand-held setup or mounted on an AscTec Hexacopter Neo for different viewpoints. The datasets are:

Shopping street 1 dataset [21] ⟼ Ground-Ground is a hand-held dataset with the camera revisiting the same location with very similar viewpoints in a busy shopping street in Zurich.

OldCity dataset [13] ⟼ Ground-Ground consists of two walking sequences of 230m in the old city of Zurich, presenting a more complex scenario due to the presence of narrow passages in this area, providing wide range of viewpoints of the same places.

Clausius street dataset [21] ⟼ Air-Air is a dataset recorded along a residential street with the camera mounted on the UAV, facing the buildings of one street side, while performing lateral movements with the UAV in both directions. The two air sequences exhibit large viewpoint changes, perceptual aliasing and strong lighting changes.

Clausius street dataset ⟼ Air-Ground was recorded in the same street, with the air sequence taken from the previous dataset, while a new hand-held sequence was recorded on the same day. This is the most challenging real dataset because of its extreme viewpoint changes.

V. Experimental Results

We benchmark the proposed pipeline against three state of the art place recognition algorithms that are suitable for UAV navigation, referred to here as BoBW [7], ORTHO [13] and VTPR, a modified version of [21] for ease of comparisons. In particular, VTPR here, corresponds to the methodology of [21], albeit using the same feature descriptors (i.e. BRISK instead of BRISK-48-bytes) as used in our method, as well as small modifications in the candidate filtering step. This strategy reveals the true power of map densification, which is also the main contribution of this work. It should be noted, however, that with these modifications VTPR achieves slightly better results than the original method of [21]. The use of BoBW with ORB [23] features in [2], was shown to provide scale and rotation invariance, while keeping real-time capabilities. ORTHO makes use of BRISK [22] features and

minimizes the effect of viewpoint changes by using a mesh-based approach to create orthophotos projecting the image to the most salient plane in the scene.

Although the decision of the feature detector and descriptor to be used is left to open in the proposed pipeline, here we choose to run our experiments using BRISK features, which provide a good matching performance at a very low computational cost. To build a visual vocabulary as in [7], we discretize a BRISK descriptors' space using 6000 images, different from the ones used for testing, depicting indoor and outdoor environments. A vocabulary of 1 million words is generated by building a vocabulary tree with 10 branches and 6 depth levels. The same vocabulary is used throughout all the experiments, demonstrating the robustness of the method.

A. Narrow viewpoint changes

We test the proposed pipeline and the selected algorithms on narrow baselines in order to validate our algorithm on publicly available datasets against the state of the art in conditions that existing algorithms are designed for.

First, we record the precision-recall curves for all algorithms on the Shopping Street 1 dataset, which depicts a planar scene at small viewpoint changes. All the algorithms perform well in this dataset, with the proposed method presenting the highest recall (0.96) at precision 1, against 0.94 for both BoBW and VTPR, and 0.78 for ORTHO.

Precision-recall curves for the Old City dataset are visible in Fig. 7. This dataset exhibits both small and challenging viewpoint changes. As such, all algorithms can recover correct loops in areas with small changes in viewpoint, while maintaining perfect precision. However, the proposed method can also recover correct loops in areas with challenging viewpoint changes, achieving recall 0.79 at precision 1 and outperforming all others algorithms Example loop-closure detections using the proposed approach in the Shopping Street 1 and Old City datasets are shown in Fig. 8a and 8b, respectively.

The methods were also tested in the Air-Air Clausius Street dataset. The loop-closures detected by our approach and a correct match are illustrated in Fig. 9. While the proposed approach detects one false positive loop, BoBW and ORTHO detect only few correct matches and much more false positives in this dataset. VTPR detects about half of the loops detected by the proposed approach, as can be seen in comparison to the results in [21], however, without any false positive detections.

B. Image Retrieval and Candidate Filtering in wide viewpoint changes

In our exploration towards robust loop-closure detection under large viewpoint changes, the first step was to determine whether our image retrieval algorithm works in these conditions and how many of the top candidates we need in order to guarantee a good chance of having at least one correct candidate in the set passed on to the geometric check. Fig. 10 shows the percentage of queries with at least one correct candidate before and after the candidate-filtering step, while varying the number of images retrieved from the image database. Note that the candidate filtering step not only removes erroneous candidates, but also filters out some correct loop-closure candidates. Empirically, retrieving the top 30 most similar images to a query from the image database is enough to provide enough correct candidates for the next steps of the pipeline. Despite the decrease from 97% (before candidate filtering) to about 88% (after candidate filtering) in the percentage of queries with at least one correct candidate in L'Agout 0°-45° sequence matching, most of the queries can still provide correct candidates for the geometric check, without much compromise in performance.

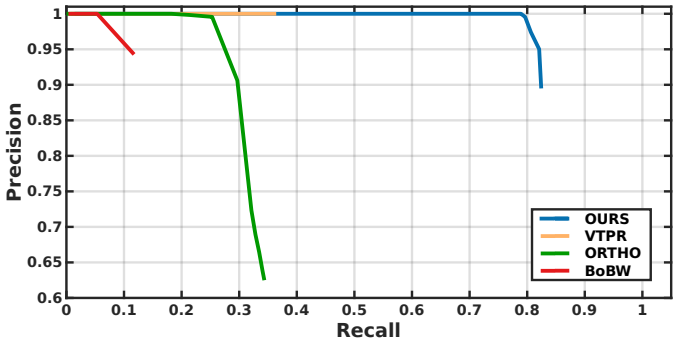


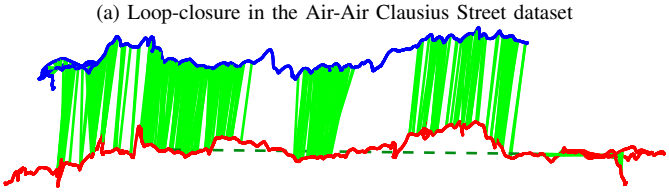
Fig. 7. Precision-Recall Curves for the Old City dataset, showing that the proposed approach outperforms BoBW and ORTHO in scenarios where these algorithms are designed for, planar scenes (in the case of ORTHO) and narrow viewpoint changes.

(a) Shopping Street 1: small viewpoint changes

(b) Old City: more challenging viewpoints

Fig. 8. Example loop-closures from the Shopping Street 1 and Old City datasets.

(a) Loop-closure in the Air-Air Clausius Street dataset



(b) UAV trajectories (in red and blue) and detected loops (in bright green)

Fig. 9. Loop closures in the Air-Air Clausius Street dataset: in (a), is an example loop-closure detected using the proposed method and in (b), the trajectories followed by the UAV, and the loops correctly detected between them. A false loop detection is shown in the dashed green line.

C. Wide viewpoint changes

In order to evaluate how the proposed method performs with increasing changes in viewpoint, we first test for loop-closures within the L'Agout dataset. We test the sequence at 0° against all others (i.e. 15°, 30° or 45°). Except for the neighbours of the current position, that depict the same place and cannot be detected, no self-loops exist along a single sequence. However, as all images entering the pipeline are tested for loop-closures before being inserted into the database of images, false positive detections are still possible inside one sequence. Fig. 11a shows the precision-recall curves for L'Agout for all the algorithms. Although all algorithms perform well at 15° of viewpoint changes (i.e. 0°-15°), both VTPR and the proposed method achieve the highest recall (0.97) for perfect precision. At 30°, both methods achieve a recall of 0.72 for perfect precision against 0.21 for ORTHO, while BoBW fails to detect loop-closures. The robustness of the proposed algorithm in viewpoint changes becomes evident at larger angles. At 45°, the proposed approach achieves recall of 0.54 for perfect precision against 0.38 for VTPR, representing an improvement of 42% with relation to the latter one, while both BoBW and ORTHO fail quickly. Fig. 13 shows a correct loop-closure detected in the L'Agout 0°-45° dataset, using the proposed approach.

We repeat the same experiment for the Corvin dataset, which captures a scene with strong depth variations. We record precision-recall curves for the sequence at 0° against the one at 30° and at 45°. As evident in Fig. 11b, these datasets present great challenges for all algorithms, with BoBW and ORTHO failing quickly. While VTPR achieves, a recall of 0.5 at 30° and 0.04 at 45° viewpoint changes for perfect precision, the proposed method achieves a recall of 0.71 at 30° and 0.14 at 45° for the same precision. This represents an improvement of 40% at 30° and 250% at 45° of viewpoint changes, when compared to VTPR. Fig. 14 depicts correct loop-closure detections, in the Corvin dataset, using the proposed approach.

The methods were also tested in the Air-Ground Clausius Street dataset. While our approach detects one false positive loop and many correct loops, as shown in Fig. 12, BoBW, ORTHO and VTPR detect only very few correct matches (less than 5) and few more false positives.

VI. TIMINGS

As consecutive frames are usually very similar, loop-closure detection does not need to be attempted at every frame, so in practice, runtime in the range of 1-5 Hz is enough for real-life applications. In the worst-case scenario, where all candidates entering the geometric check are tested for loop-closure, the proposed algorithm runs at 5Hz on average on a single core Intel i7 2.8GHz, allowing real-time place recognition within a SLAM system. Setting a maximum of 50 image-candidates at the end of the image-retrieval step, we avoid compromising the timings in cases of longer robot trajectories (resulting to larger image databases). In reality, even faster performance is expected as the geometric check can abort as soon the first suitable candidate is found.

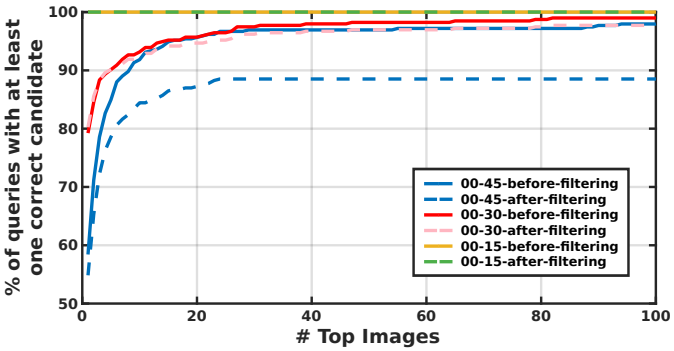
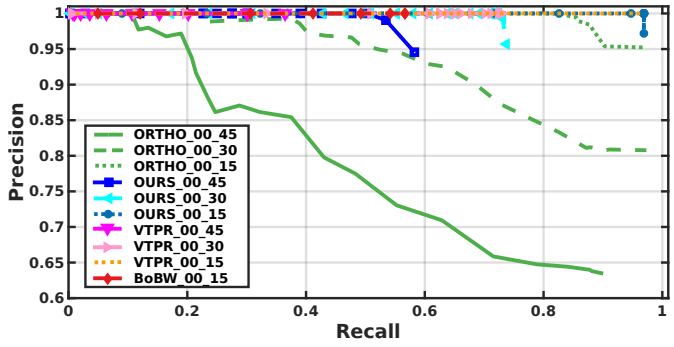
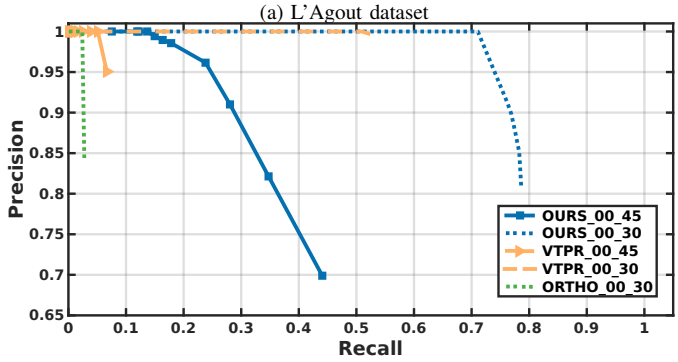


Fig. 10. The percentage of queries with at least one correct loop-closure candidate to be passed on to the geometric check for different viewpoint changes. We provide curves both before and after the candidate-filtering step used for efficiency, while varying the number of top images retrieved from the image database. The higher the percentage achieved, the better the chance of discovering the correct loop-closure after the geometric check.

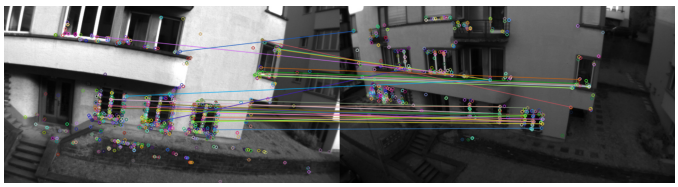


(a) L'Agout dataset

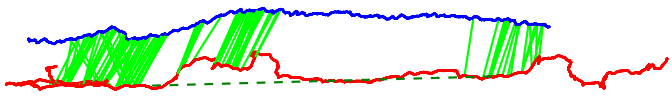


(b) Corvin dataset

Fig. 11. Precision-Recal Curves in the L'Agout dataset in (a) using different viewpoint variations (from 0° to 15°, 30° and 45°), and in the Corvin dataset in (b) while varying the viewpoints from 0° to 30° and to 45°.



(a) Loop-Closure in the Air-Ground Clausius Street dataset



(b) UAV trajectory (in red and blue) and detected loops (in bright green)

Fig. 12. Air-Ground Clausius Street dataset: In (a), example loop-closure detected using the proposed method and in (b) the trajectory followed by the UAV and by the hand-held setup, and the loops correctly detected between them. A false loop detection in the dashed green line.

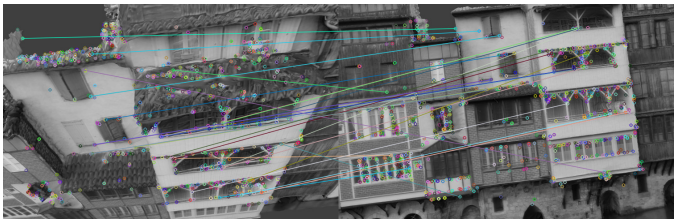


Fig. 13. An example loop-closure detection in the L'Agout dataset using the proposed approach for a change in viewpoint from 0° to 45°.

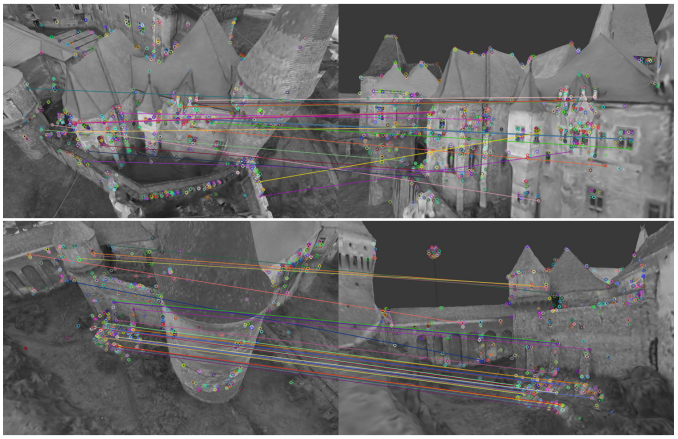


Fig. 14. Example loop-closure detections in the Corvin dataset using the proposed approach. A viewpoint change from 0° to 45° illustrates the extent of the challenge in this dataset.

VII. Conclusion

This paper proposes a new place recognition pipeline capable of addressing dramatic changes in viewpoint (of up to 45°), while maintaining robustness at smaller angles, from narrow baselines. It relies on a depth-completion approach to improve the establishment of 3D correspondences during geometric checks, enabling feature-based matching across images captured from very wide baselines.

Evaluation on synthetic and real datasets with both hand-held and aerial footage, reveals that the proposed method achieves significant improvement in precision and recall in comparison to the state of the art, while keeping onboard computation affordable for autonomous UAV navigation, demonstrating that feature-based techniques still have a lot to offer in place recognition at extreme viewpoint changes.

To the best of our knowledge, the new synthetic datasets presented here are the first to completely isolate the problem of viewpoint changes for place recognition, closing a crucial gap in the literature. To facilitate further research on this topic, our datasets are publicly available.

Acknowledgment

The authors would like to thank Kimbo and CNRS UMR Traces, for the 3D models used in the synthetic datasets.

References

[1] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics (T-RO)*, 2015.

[3] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2003.

[4] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, 2012.

[5] R. Arandjelovic and A. Zisserman, "All about vlad," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[6] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *Int. Journal of Robotics Research (IJRR)*, 2011.

[7] D. Galvez-Lopez and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics (T-RO)*, 2012.

[8] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab, "Benchmarking template-based tracking algorithms," *Virtual Reality*, 2011.

[9] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, "Point cloud descriptors for place recognition using sparse visual information," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, 2004.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[12] J.-M. Morel and G. Yu, "Asift: A new framework for fully affine invariant image comparison," *SIAM journal on imaging sciences*, 2009.

[13] F. Maffra, L. Teixeira, Z. Chen, and M. Chli, "Loop-closure detection in urban scenes for autonomous robot navigation," in *3D Vision (3DV), 2017 International Conference on*, 2017.

[14] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla, "Are large-scale 3d models really necessary for accurate visual localization?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3d point clouds," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[16] R. Dubé, M. G. Gollub, H. Sommer, I. Gilitschenski, R. Siegwart, C. Cadena, and J. Nieto, "Incremental-segment-based localization in 3-d point clouds," *IEEE Robotics and Automation Letters*, 2018.

[17] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," 2011.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.

[19] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *CVPR*, 2016.

[20] A. Kendall, R. Cipolla, *et al.*, "Geometric loss functions for camera pose regression with deep learning," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] F. Maffra, Z. Chen, and M. Chli, "Tolerant place recognition combining 2d and 3d information for uav navigation," in *Proceedings of the IEEE International Conf. on Robotics and Automation (ICRA)*, 2018.

[22] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT and SURF," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[24] L. Teixeira and M. Chli, "Real-time mesh-based scene estimation for aerial inspection," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2016.

[25] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, and R. Siegwart, "Keyframe-based Visual-Inertial SLAM using Nonlinear Optimization," in *Proceedings of Robotics: Science and Systems (RSS)*, 2013.

[26] F. Mal and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[27] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Vol.1)*, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework.