Loop-Closure Detection in Urban Scenes for Autonomous Robot Navigation

Loop-Closure Detection in Urban Scenes for Autonomous Robot Navigation

Fabiola Maffra, Lucas Teixeira, Zetao Chen and Margarita Chli
Vision for Robotics Lab, ETH Zurich, Switzerland

Abstract

Relocalization is a vital process for autonomous robot navigation, typically running in the background of sequential localization and mapping to detect loops in the robot's trajectory. Such loop-closure detections enable corrections for drift accumulated during the estimation processes and even recovery from complete localization failures. In this work, we present a novel approach loosely integrated with a keyframe-based SLAM system to perform loop-closure detection in urban scenarios for autonomous robot navigation. Generating a mesh of the current robot's surroundings in real-time using monocular and inertial cues, the proposed method estimates the most salient plane in the current view, enabling the creation of the corresponding orthophoto for this plane. Evaluating image similarity on orthophotos forms a much better conditioned problem for relocalization, minimizing effects from viewpoint changes. Employing binary image descriptors and tests on their relative constellation in the image, the proposed approach exhibits robustness also to illumination and situational variations common in real scenes, overall resulting to significant improvement in loop-closure detection performance in urban scenes with respect to the state of the art.

Video – https://www.youtube.com/c/V4RLteam

1. Introduction

The emergence of powerful techniques for robotic egomotion estimation and map building that follow the SLAM (Simultaneous Localization And Mapping) paradigm has been drawing research and industrial interest in recent years, as this is the core ability of spatial understanding for autonomous robot navigation. With the aim of developing general and practical systems, the use of external tracking or unreliable positioning systems (e.g. GPS) is typically avoided, albeit restricting the scalability of approaches for robot navigation as drift inevitably accumulates over time during sequential processing (especially during exploratory trajectories). Detecting when a robot returns to a previously visited place has long been known to offer useful cues for diminishing the effects of drift and similarly, detecting when one robot returns to a place already visited by another robot can also form the basis of any collaboration amongst them [17]. In both cases, it is the problem of Place Recognition that needs to be addressed, *aka* Loop-Closure detection. Following such a loop detection, new pose-to-pose and pose-to-features constraints are established in the SLAM graph, subject to non-linear optimization, such that the loop closure is enforced and the effects of the drift correction are propagated back to the rest of the SLAM graph.

Primarily addressed using visual cues, place recognition is a challenging task, due to the large appearance variations that the same physical place in the world can exhibit. Illumination and situational variations in the scene's appearance become an issue even at different times of the same day and are certainly caused by weather or seasonal changes, while viewpoint changes or dynamic objects add on to the challenge of identifying place similarity. While impressive works exist in the literature addressing some of these variations in isolation, it is still very challenging to simultaneously address them all together, which is key in enabling robust robot navigation in real tasks. In this spirit, this paper proposes a new orthophoto-based approach for loop-closure detection in the presence of viewpoint, illumination and situational variations. With the rationale that comparing orthophotos instead of perspective images poses a far better conditioned query for place recognition, the proposed approach achieves high recall, while filtering out ill posed queries effectively, and thus, minimizing the probability of false positives.

In this paper, we specifically study the problem of urban robot navigation with the outlook of employing such a system for automating the navigation of small Unmanned Aerial Vehicles (UAVs), which are restricted to small payload and limited computational capacity. Moreover, exhibiting great agility, they highlight the need for viewpoint tolerant place recognition techniques. In urban scenarios, we assume the presence of a high density of structures that are largely planar and are common in man-made environments. This assumption allows us to utilize a planarity prior on the scene and harvest the robustness it can bring to place recognition in this scenery. As a result, the main contribution of this work are:

- a new approach to generate orthophotos in real-time from sparse features provided by a visual-inertial SLAM algorithm, and

- a novel loop-closure detection framework for robot navigation in urban scenes, which does not require any previous knowledge of the environment nor does it impose unrealistic assumptions (e.g. Manhattan world).

Evaluated on challenging datasets, the proposed approach achieves higher precision and recall with respect to the state of the art, exhibiting unprecedented robustness to viewpoint, illumination and situational changes.

2. Related Work

Place recognition is most often addressed using appearance-based cues and as a result, draws inspiration from Image Retrieval from the Computer Vision literature. Identifying whether a query image is present in the database (i.e. containing all past experiences of the robot in the robot navigation paradigm) can be a very inefficient process, so for this purpose, visual dictionaries have been devised to retrieve matching images with high probability. Inspired by text retrieval techniques, the pioneering work in [18] gave rise to what is widely known as the Bag Of Words (BOW) approach. This technique relies on building a dictionary of visual words by clustering locally invariant feature descriptors, such as SIFT [12], appearing in a set of model images and then representing each image as the set of visual words it contains. The use of this representation, permits the analogous application of many theoretical developments such as TF-IDF (Term Frequency - Inverse Document Frequency) and probabilistic naive Bayes [13] from the fields of text retrieval and classification on images [5, 18]. Such techniques, naturally, apply well to place recognition for mobile robots, and are generally well-established in the field, including extended generative models for location observations [1, 6].

The success of BOW approaches in searching for similar images in a database has led to their wide use, however, it was soon realised that their performance decreases with the size of the vocabulary, not only affecting complexity, but also encouraging misclassification. The FABMAP framework [6], partially alleviating the latter by learning the dependencies between visual words, in a framework that is currently considered one of the highest performing pipelines for loop-closure detection in robot navigation scenarios. Its reliance on computationally expensive image features (i.e. SURF [3]) and intolerance to even small viewpoint changes restricts the applicability of FABMAP to scenarios targeting ground robots with large computational capabilities. As with FABMAP, a common source of error in the vast majority of place recognition systems is that they discard most of the geometric information in the image/scene when comparing feature sets. As a consequence, the discriminative nature of the model is reduced, typically resulting in either perceptual aliasing or reduced recall. Full feature-based comparisons can be computationally expensive, and therefore most of the underlying structure and geometry between features is generally ignored, such as in [5]. Following this realization, a handful of works [8, 14] have investigated ways of incorporating some geometric information into the location models. A common approach is to perform RANSAC [7] to compute a transformation between a query and match candidate images [14].

Probably the most relevant work to this paper is the work in [8], who employ the binary features ORB [16] in a BOW approach and demonstrate its successful applicability to ground robot navigation. As binary features are computationally drastically more efficient than their floating point counterparts (e.g. SURF), they are most commonly used during SLAM [11, 14]. As a result, re-using them for place recognition promises to eliminate unnecessary computational effort, however, the robustness of place recognition systems based on binary features to common scene variations is limited. Inspired by these limitations, in this work we propose to make use of SLAM's 3D estimation to recover a mesh of the local workspace of the robot, which in turn enables the estimation of an orthophoto of the current view. By forming place recognition queries employing binary features in orthophotos, the problem of assessing image similarity using binary descriptors is shown to become more stable and achieve improved performance.

The underlying assumption of largely planar scenery made in this work has also been used to generate orthomosaics from aerial imaging. Orthorectification, essentially facilitates the alignment of images taken from different viewpoints to form a larger mosaic, and as shown in Baatz *et al*. [2] the overlapping part of two orthophotos of the same place is typically very similar resulting to their straightforward alignment. Testing on imagery of buildings facades, [2] factorize the rotation out of the recognition problem by generating gravity-aligned orthophotos outperforming purely 2D-based methods. In a similar spirit, Chen *et al*. [4] demonstrate a gain in place recognition by combining both unmodified perspective images and their corresponding orthophotos. However, both works assume the existence of a prior 3D environment model, which can be unrealistic in some applications.

Inspired by [2], in this paper, instead of searching for a perfect alignment between images, we aim to verify whether the configuration of features shared by two orthophotos presents a consistent layout. This step is known as a geometrical check in loop-closure algorithms. Moreover, in this work, the orthophoto plane is directly extracted from the 3D landmarks used for the robot visual navigation

system without the need of computing lines and extracting vanishing points, as in [2].

3. Methodology

As visual-inertial (VI) SLAM is typical in robot navigation, and UAV navigation in particular, the proposed system is interfaced with a nominal VI SLAM system processing cues from a single camera and an Inertial Measurement Unit (IMU). The pipeline, however, is largely agnostic to the type of vision-based SLAM used, with the only requirements of knowledge of the gravity direction and the metric scale. Generating a mesh in 3D out of the local SLAM landmarks, the predominant plane in the scene is identified and the orthophoto corresponding to the current view (i.e. Q in Figure 1) is generated. Extracting binary features on this orthophoto, the pipeline queries the orthophotos database for an appearance based map identifying possible loop-closure candidates. These are then subjected to a geometric check seeking candidates with matching relative constellation of features in the orthophoto space. Considering the robot navigation paradigm, in the following, we assume that the robotic platform at hand has a monocular-inertial sensor suite onboard.

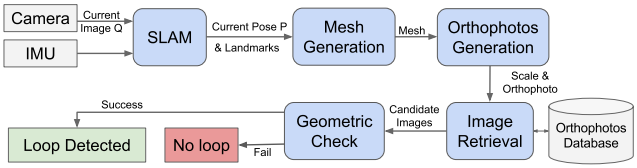

Figure 1. The proposed pipeline for place recognition employing mesh-based orthophoto generation with appearance and geometric checks to determine whether the current image Q forms a loop closure with an image in the database containing past robot experiences.

3.1. Real-Time Visual-Inertial Scene Estimation

In this work, we use the open-source keyframe-based VI SLAM algorithm OKVIS [11], which estimates the trajectory of the robot considering a limited window of past poses, and as a result has no loop-closure detection or correction scheme. OKVIS provides in real-time, the current robot pose P and a 3D map comprising of the estimated locations of 3D visual landmarks extracted from the image feed. These are fed into the open-source mesh generation pipeline of [19], which was demonstrated to robustly compute the 3D mesh of the landmarks visible from P in a computationally very lightweight manner, providing a denser scene representation. Filtering out inconsistent landmark measurements from SLAM, the mesh generation algorithm applies a local Laplace filter, implicitly enforcing local smoothness. This is crucial for robust orthophoto generation, as it has a direct effect on the detection of the most salient plane in the scene.

3.2. Orthophotos Generation

An orthophoto of a largely planar scene is the orthogonal projection of this scene onto the most dominant plane of the scene; so in essence, the orthophoto of a perspective image corrects for the camera tilt and the terrain relief. Figure 4 illustrates an example of an orthophoto generated by the algorithm from the Old City dataset introduced in Section 4.1. Although the environment is usually not planar in general, in urban scenes structures are largely planar and aligned to the gravity direction. In this work, we select the biggest gravity-aligned plane in the image as the orthophoto plane. The rationale behind this is that when viewing the same place at different times, from different viewpoints, most of times, the same orthophoto-plane can be extracted, and as a result, place recognition can be effectively performed.

The generation of orthophotos first requires the estimation of the most predominant plane in the image, that will serve as the orthophoto plane. This estimation is facilitated by the 3D mesh provided by the VI-SLAM and the Mesh generation module. Aligning the mesh's coordinate frame with gravity (OKVIS already provides a gravity aligned map), we project the 3D mesh to the 2D top view of the scene. The longest line in this view corresponds to the largest vertical plane in the 3D scene. In order to recover this line, we use an iterative Huber M-Estimator to fit a line to the 2D SLAM points (i.e. the mesh's vertices) considering any point within a pre-specified distance to the estimated line (here $40cm$) as inliers. Upon discovering the longest line in the top view of the scene, we set the middle of it to correspond to the center of the orthophoto-plane. The normal of this plane is selected as the normal of the line that points to the direction of the camera in the gravity aligned SLAM coordinate frame.

In order to project the current perspective image to the estimated orthophoto plane, we first find where the four corners of the frustum of the camera intersects with this plane (i.e. points P1, P2, P3 and P4 in Figure 2). With this information, we form a homography to transform this plane from image coordinates to metric coordinates and use this to project the perspective image onto the orthophoto plane, forming the orthophoto. In order to restrict the size of this orthophoto, we rescale it to the maximum of twice of the original resolution. We impose this restriction because robot cameras have very low resolution, in our case 752x480. So a higher rescaling factor in addition with the orthogonalization of the image creates a very distorted image.

3.3. Image Retrieval

In order to detect revisited places we make use of a hierarchical Bag of Binary Words (BoBW) visual vocabulary,

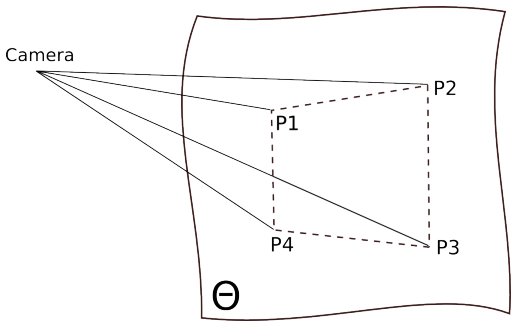

Figure 2. The intersection of the camera frustum with the estimated orthophoto-plane Θ to be used in order to form the homography to be applied on the perspective image for the generation of the corresponding orthophoto.

Figure 3. An example of an orthophoto (on the right) generated automatically by the proposed framework for the corresponding original image shown on the left, by estimating a local mesh illustrated in Figure 4.

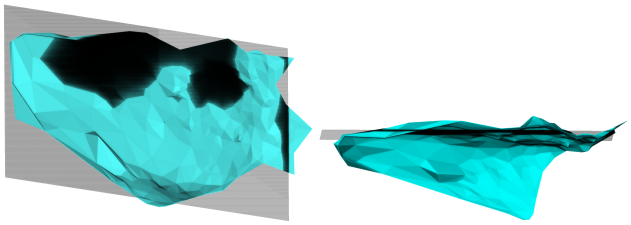

Figure 4. The mesh of the scene of Figure 3 in cyan and the main plane extracted from it in grey. The side view can be seen on the left and the top view on the right.

describing an image as a collection of visual words combined with an inverted file index. In this work, the visual database consists of orthophotos generated from perspective images captured using a traditional perspective camera. Each entry in the database comprises an appearance signature of the corresponding image, namely its BoBW descriptor. Following the approach suggested by Galvez and Tardos [8], we adapt for the binary features used in OKVIS, namely the BRISK features [10]. Namely, we build a visual vocabulary by discretizing the 48-byte BRISK descriptors' space. In order to train this vocabulary, we used about 6000 images comprising both indoor and outdoor environments, different from the ones used for testing. The vocabulary tree built has 10 branches and 6 depth levels, resulting to a vocabulary of one million visual words.

In order to query the orthophoto of the current view Q for appearance matches in the orthophotos database, BRISK features are detected and the BoBW descriptor for Q is formed. The vocabulary tree is used to score the L1-distance of this descriptor against the entries in the orthophotos database using a TF-IDF weighting scheme [6] to suppress commonly occurring words and form the set of matching image candidates.

3.4. Geometric Check

The BOW approach discards all spatial information of visual words by definition, effectively accepting as any match two images having the similar visual features regardless of their relative constellation in the image space. A geometric check based on a RANSAC scheme is usually applied after appearance matching to improve loop closure detection by verifying whether the configuration of features belonging to these two images presents a consistent layout. When matching gravity-compatible orthophotos, Baatz *et al*. [2] reduce the 6 DOF perspective recognition problem to a homothetic problem involving only scale and a translation in a 2D plane. By exploiting the fact that they are solving a homothetic problem, [2] suggests to replace the computationally expensive RANSAC-based geometric check by an efficient 1D voting scheme, where scale as well as a horizontal and a vertical offset are estimated separately.

The proposed approach conducts geometric verification to every query-candidate orthophotos pair that is shortlisted by the image retrieval module. By making use of the metric scale provided for each orthophoto during their creation, we first convert both images to a common scale, which allows us to use the 1D voting scheme for both horizontal and vertical displacement. With both the query and the candidate matching orthophotos in the same scale, we establish BRISK correspondences across features detected in both images.

Following the approach of [2], we estimate the horizontal x and vertical y components of the relative translation between the query Q and the candidate C, independently. Every pair of corresponding points (x_C, y_C) and (x_Q, y_Q) contributes with one vote for the x-displacement $(x(i) = x_C(i) - x_Q(i))$ and one vote for the y-displacement $(y(i) = y_C(i) - y_Q(i))$. The global displacement of the x-coordinate is determined by fitting a probabilistic density function to all the votes computed along the axis x. To this end we use a Kernel density estimation (KDE) supported by a Gaussian kernel, where each offset in x contributes with a Gaussian probability density function with mean centered at $x(i)$ and a standard deviation defined by a translation tolerance in meters. The probability density function is then computed by summing up all these contributions and the global maximum of this distribution is used as the global displacement in this direction. The corresponding points whose coordinate differences are within a certain distance from the global displacement in x are considered inliers.

Since all the coordinates are expressed in meters it is easy to define a distance tolerance to compute the inliers set. The same procedure is then applied to compute the displacement in y. The intersection of the two resulting inlier sets constitutes the final inliers of the geometric check. The number of inliers is then used as a metric to decide whether a candidate should be accepted as a loop closure to match the query image. The different thresholds applied are analysed in section 4 by means of precision-recall curves.

4. Experiments and Results

While there do not exist directly comparable methods for place recognition using orthophotos, as a baseline algorithm, we form a variant of the proposed pipeline adapted to use perspective images as done traditionally in robot navigation scenarios, as the monocular-based ORB-SLAM [14]. This enables fairness of comparisons as we ensure that all tests use the same features and are subject to the same quality of SLAM estimation. As the geometric verification voting scheme is not suitable when using perspective images for the variant pipeline that we refer to as $Persp_{FM}$, we implemented the strategy used in the BoBW approach in [8]. This consists in computing a spatial transformation between the matched images by estimating the fundamental matrix using RANSAC for the variant algorithm. The proposed method from here onwards is referred to as $Ortho_{TR}$ to denote the use of Orthophotos with the voting scheme used to estimate translation.

4.1. Datasets

Existing place recognition datasets normally only contain visual information, however, in order to put our proposed approach to the test, we need visual and inertial sensing information, as well as ground truth. Outdoor visual-inertial datasets, such as KITTI [9] are designed for motion estimation and are not well suited for testing place recognition as they exhibit mainly forward camera motion with a front-looking camera, rendering it very difficult to label the images for ground truth in loop closures.

All the datasets used in this paper were recorded using a visual-inertial sensor [15] providing grayscale global-shutter images at 20 Hz synchronized with inertial measurements. For our experiments, we use information from only one of the two cameras of the sensor to conduct monocular-inertial estimation. The datasets were recorded using a hand-held setup with the camera facing perpendicular to the direction of motion (i.e. side-looking). All imagery was labelled for ground-truth loop closures, by first using any priors on GPS information whenever available to suggest potential loops and then manually correcting these suggestions. Below, we describe in detail all the datasets used in this paper.

4.1.1 Shopping Street sequences 1 & 2

Two datasets were recorded when walking down a busy shopping street with many pedestrians. Examples are shown in Figures 9 and 10. Shopping Street 1 was recorded with the sensor held at eye-level height and exhibits loops with small viewpoint changes, perceptual aliasing and changes in the scene appearance. Shopping Street 2 was recorded along the same street a few months later with the sensor mounted at the top of a $4m$-long rod held vertically in order to capture the scenery captured in Shopping Street 1, at least partially, but from different viewpoints. By combining these two sequences, a very challenging place recognition dataset is created, where the scene is not only revisited from very different viewpoints, but due to the large time interval between recordings, strong appearance variations can also be observed with most of the restaurants and shop windows in different configurations; e.g. shutters closed, window displays and even store logos changed. Moreover, parts of Shopping Street 2 exhibit large variance in illumination conditions, making it hard even for humans to detect whether it is the same place visited in the first sequence. These sequences have a total of approximately 1200 meters and 26 mins.

4.1.2 Old City sequences

Two sequences were recorded at the end of the day in an old city area, exhibiting similar characteristics as the Shopping Street datasets, albeit with more challenging viewpoint variations. This dataset comprises two traverses along the same route, each one covering a distance of approximately 230 meters. In total, 10 minutes of data were recorded for this dataset. Example images are shown in Figure 5.

4.2. Orthophotos versus Perspective Images

Aiming to verify whether using the orthophotos generated by $Ortho_{TR}$ can perform better than their perspective counterparts in a place recognition scenario, we test for loop closures within Shopping Street 1, which comprises two different traverses one the same day, along the same route. Images from the first traverse are used to populate the database of images, and using images from the second traverse this database is queried for loop closures. Parts of these trajectories does not overlap and in that case loop closures should not be detected.

Each of $Ortho_{TR}$ and $Persp_{FM}$ builds their own, separate database of images for retrieval; $Ortho_{TR}$ builds a database of orthophotos, while $Persp_{FM}$'s database comprises of the perspective images. It is important to note that only the perspective images with more than 30% of inliers have their corresponding orthophotos computed. If an image does not meet this requirement, it is is not considered neither $Ortho_{TR}$ nor $Persp_{FM}$ during this test. For both

Figure 5. Example loop-closuring pairs from the Old City dataset identified using with the proposed approach. Each group of four images shows the original perspective views in the top row and the respective orthophotos in the bottom row.

pipelines the image retrieval step considers the top 10 best images recovered from the database, while the corresponding geometric check is run for every pair of query-candidate images. Their performances are illustrated by the dashed lines plotted in Figure 6, demonstrating that $Ortho_{TR}$ performs consistently better than $Persp_{FM}$ in this scenario. This attests to our earlier claim that using orthophotos, place recognition can be more robust and accurate, compared to employing perspective images for the same tests. The performance of the two systems in a more general scenario is recorded, in which all the images in the Shopping Street 1 sequence images are considered, shown in the solid lines in Figure 6. As expected, in this case the recall for $Ortho_{TR}$ decreases, but still performs systematically better than $Persp_{FM}$.

We also compute precision-recall curves for Shopping Street 1 + 2 and Old City using both $Ortho_{TR}$ and $Persp_{FM}$ algorithms as can be seen in Figure 7 and Figure 8, respectively. As previously done, the first sequence of each dataset is inserted into the database of images, while the second one is used to form image queries. For Shopping Street 1 + 2, we insert the first loop of Shopping Street 1 into the database and form queries from the Shopping Street 2 sequence. $Ortho_{TR}$ is evidently able to maintain higher precision than $Persp_{FM}$, essentially attesting to better consistency of performance, rendering it more trustworthy in

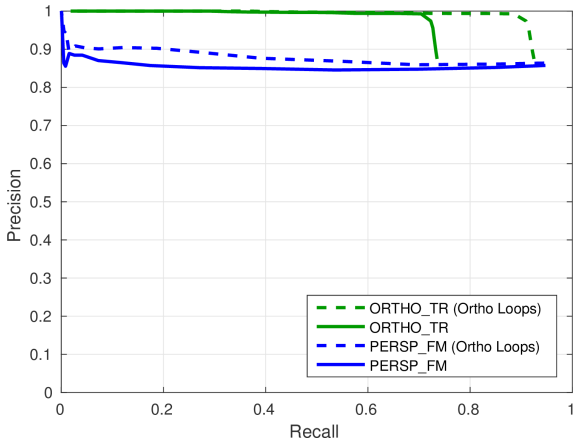

Figure 6. Precision-recall curves on Shopping Street 1 comparing the performance of using orthophoto (green) and perspective (blue) images for place recognition. Dashed lines indicate the respective performances when considering only the images with enough inliers to generate orthophotos, while solid lines illustrate performances when all the images of the sequence are considered.

closing loops during autonomous robot navigation. Example loop closures for Shopping Street 1 + 2 and Old City are shown in Figures 5 and 9, respectively. As evident in Figures 6-8, the proposed method achieves superior precision as long as orthophotos can be successfully generated, but exhibits a sharp decline in precision when no dominant plane can be detected (e.g. due to a tree occupying most of the image space) or no corresponding orthophotos can be generated. In the latter case, loop closures cannot be detected. However, in urban scenes planes are ubiquitous and as shown in Figure 10 even if there are multiple planes present it is still possible to generate corresponding orthophotos.

4.3. Viewpoint Changes and System Scalability

In order to test different extents of viewpoint variations using both perspective images and orthophotos, we implemented three different spacing policies between consecutive images when populating our database. In the first setting, all the keyframes used by OKVIS are inserted into our database of images, resulting in a big overlap between consecutive images in our sequence. In the second setting, an image is only inserted if it is at least 2 meters away from the previously inserted image. In the last setting, a distance of 5 meters between consecutive images is considered, leading to a much more challenging place recognition scenario as illustrated in Figure 12.

Both pipelines, $Ortho_{TR}$ and $Persp_{FM}$, were tested using these three different policies. Using the same strategy as before, in a first step all relevant images from the first traverse are used to populate the corresponding image database and then all the images in the second traverse are

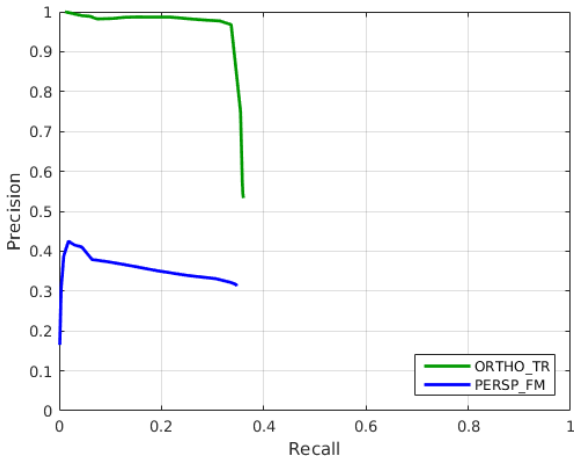

Figure 7. Precision-recall curves on the combined Shopping Street 1 + 2 dataset, comparing performances when using orthophoto (green) and perspective (blue) images for place recognition. While the reference traverse is the same as the one used in Figure 6, the test traverse here is collected at the same route after four months, thus exhibiting much stronger condition variations.

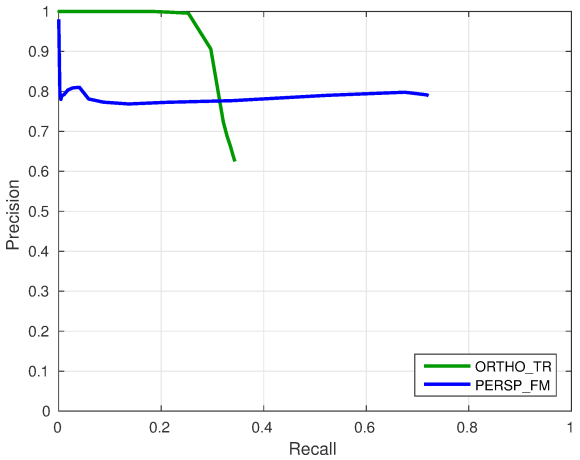

Figure 8. Precision-recall curves on the Old City dataset comparing performances of using orthophoto (green) or perspective (blue) images for place recognition. It is clear that orthophoto-based place recognition achieves much higher precision at the same recall rate.

used to query that database. Figure 11 shows the respective precision-recall curves for each case. $Persp_{FM}$ presents a sharp drop in precision-recall rates when the gap between images increases, while $Ortho_{TR}$ is still able to maintain much better recall for perfect precision. This illustrates that the orthophotos generated automatically are more robust against viewpoint variations than the perspective images, as expected. Based on these findings, it would be possible to augment the pipeline to select non-overlapping images in order to build a less confusing database of images (i.e. places), while making the place recognition problem more scalable.

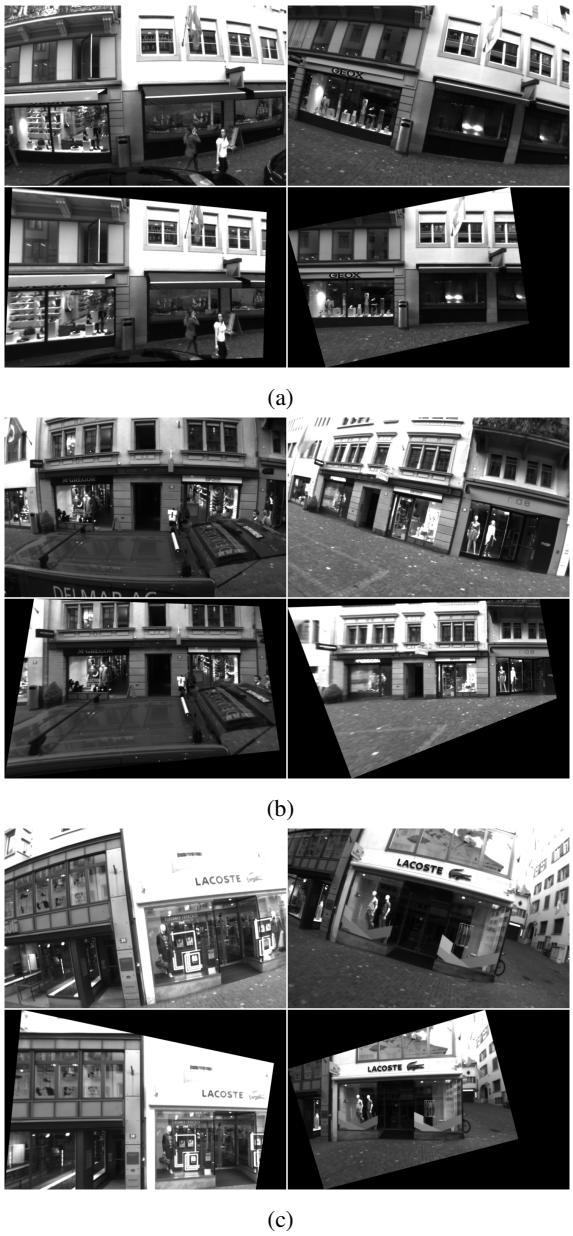

Figure 9. Example loop-closures from the combined Shopping Street 1 + 2 dataset shown in each row, using the proposed approach. In each group of images, the top row illustrates to the original perspective images, while the respective orthophotos are in the bottom row. In (a) and (b) we can observe large viewpoint and situational changes, with pedestrians and a major occlusion by a car, while (b) and (c) show difficult lighting conditions.

5. Timings

Table 1 shows timings of each individual component in the proposed pipeline averaged over all the runs in the experiments. As evident, the proposed approach is about twice real-time, with the bottleneck on the feature detec-

Figure 10. Example loop-closures from Shopping Street 1 tested with the proposed approach and their respective orthophotos. The images show that it is possible to compute corresponding orthophotos even if more than one plane is present in the scene.

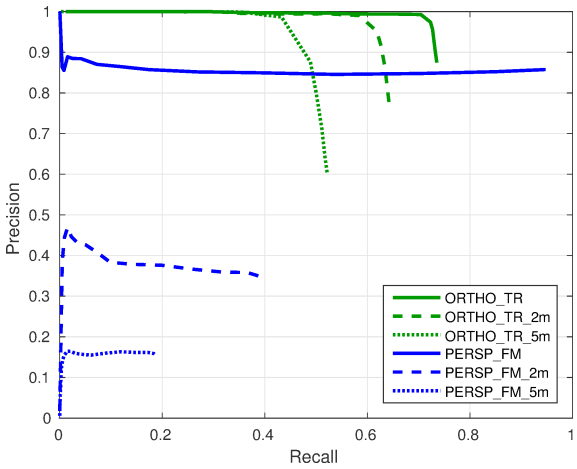

Figure 11. Precision-recall curves on the Shopping Street 1 dataset, comparing performances of using orthophoto (green) and perspective (blue) images for place recognition with different sampling spacings between consecutive images in the reference traverse (solid lines: original spacing, dashed: $2m$ spacing, dotted: $5m$ spacing across the camera's trajectory).

tion and matching. As loop-closure detection and correction usually runs on a background thread in most SLAM systems, real-time is not a requirement. It is worth noting that an adaptation to vote for the scale as in [2], the processing time can be reduced, as it eliminates the need to rescale the image and recompute features in it.

Figure 12. Example illustration of two consecutive images for different spacing strategies. From top to bottom the figure depicts no gap, 2 meters and 5 meters between the images.

Step	Average time per image
Image Rescaling	5 ms
Features Detection	40 ms
Features Matching	21 ms
KDE	0.2 ms
Total	66.2 ms

Table 1. Average timings for the online component of $Ortho_{TR}$.

6. Conclusion

This paper presents an efficient and precise algorithm to tackle the loop-closure detection problem based on orthophotos automatically generated online. Evaluation against a baseline approach employing perspective images and combined appearance and geometric checks, shows that the proposed approach achieves consistently better precision-recall characteristics in challenging datasets exhibiting viewpoint, illumination and situation changes simultaneously. Tailored for robot navigation in urban scenarios and aiming for low computational complexity, this approach makes the most of a SLAM system that is typically already running in the background in such scenarios.

Further directions include interfacing this pipeline with a global mapping algorithm to enable loop-closure correction within SLAM and harvest the benefits of a robust loop-closure detection pipeline in robot navigation.

Acknowledgement: This research was supported by the Swiss National Science Foundation (SNSF, Agreement no. PP00P2_157585) and EC's Horizon 2020 Programme under grant agreement n. 644128 (AEROWORKS).

References

[1] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics (T-RO)*, 2008. 2

[2] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Handling urban location recognition as a 2D homothetic problem. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. 2, 4, 8

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008. 2

[4] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al. City-scale landmark identification on mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[5] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research (IJRR)*, 2008. 2

[6] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *International Journal of Robotics Research (IJRR)*, 2011. 2, 4

[7] 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*, 24(6):381–395, 1981. 2

[8] D. Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics (T-RO)*, 28(5):1188–1197, 2012. 2, 4, 5

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013. 5

[10] 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. 4

[11] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *International Journal of Robotics Research (IJRR)*, 2015. 2, 3

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2004. 2

[13] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. 2

[14] 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)*, 31(5):1147–1163, 2015. 2, 5

[15] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart. A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 5

[16] 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. 2

[17] P. Schmuck and M. Chli. Multi-UAV Collaborative Monocular SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 1

[18] 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. 2

[19] 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. 3