# Placeless Place-Recognition

Simon Lynen
Autonomous Systems Lab, ETH Zurich
simon.lynen@mavt.ethz.ch

Michael Bosse
mike.bosse@mavt.ethz.ch

Paul Furgale
paul.furgale@mavt.ethz.ch

Roland Siegwart
rsiegwart@ethz.ch

## Abstract

*Place recognition is a core competency for any visual simultaneous localization and mapping system. Identifying previously visited places enables the creation of globally accurate maps, robust relocalization, and multi-user mapping. To match one place to another, most state-of-the-art approaches must decide a priori* what constitutes a place*, often in terms of how many consecutive views should overlap, or how many consecutive images should be considered together. Unfortunately, depending on thresholds such as these, limits their generality to different types of scenes. In this paper, we present a* placeless *place recognition algorithm using a novel vote-density estimation technique that avoids heuristically discretizing the space. Instead, our approach considers place recognition as a problem of continuous matching between image streams, automatically discovering regions of high vote density that represent overlapping trajectory segments. The resulting algorithm has a single free parameter and all remaining thresholds are set automatically using well-studied statistical tests. We demonstrate the efficiency and accuracy of our methodology on three outdoor sequences: A comprehensive evaluation against ground-truth from publicly available datasets shows that our approach outperforms several state-of-the-art algorithms for place recognition.*

## 1. Introduction and Related Work

Robust relocalization based on place recognition forms the backbone of many Simultaneous Localization And Mapping (SLAM) frameworks both for robotics and mobile device applications. Image-based place-recognition frameworks process a stream of images capturing the *continuous* change of scene appearance as the user navigates through space. Rather than modeling this appearance change as a continuum, existing algorithms introduce an artificial discretization of the world into places, both inside the database index and on the query side. The process of discretization requires the manual tuning of thresholds for what defines a place w.r.t. different environments (e.g. outdoor vs. indoor). Furthermore using a fixed discretization does not allow for adapting the place creation to the appearance of the environment, which in many cases is key to distinguishing places with strong perceptual aliasing. The main contribution of this paper is a continuous *placeless place-recognition* scheme that uses descriptor-vote density estimation to eliminate the arbitrary discretization and grouping of images into places. Instead we approach place-recognition as a problem of continuous trajectory matching. We find similar pieces of our path by applying a set of transformations and decompositions to the 2D space of matches between the descriptors on the query and the database side. The underlying algorithm works on individual descriptors instead of searching the image as one entity as commonly done in related work. This technique is enabled by applying a dimensionality reduction on descriptors such that they can

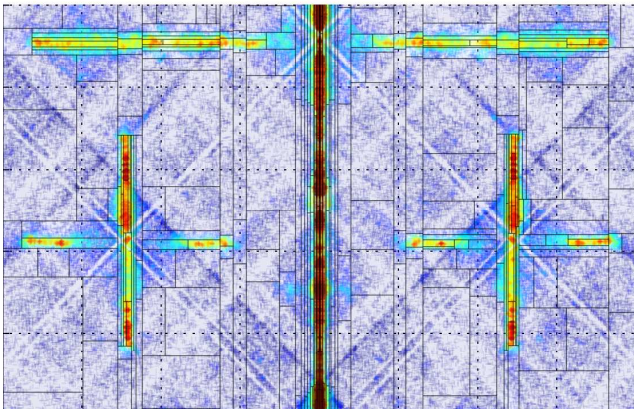

Figure 1: The proposed algorithm creates, transforms and decomposes a 2D space of descriptor votes. Regions with high vote density represent loop-closure candidates.

be searched efficiently at scale.

**Bag of words based retrieval:** Techniques from high-performance image retrieval systems have inspired many of the place recognition algorithms used in mobile phone localization and robotics applications. Most algorithms employ either a Bag-Of-Words (BoW) model or a voting scheme on individual feature descriptors such as SURF [2], BRISK [13] or FREAK [1], to find candidate images from the database.

The BoW model, initially applied to image retrieval by Sivic *et al*. [20], uses a fixed-size vocabulary as a vector quantizer to classify descriptors of both query and database images. The features extracted from an image are transformed into a vector of classes present in the image, and this vector is used to compare two images. The vocabularies are commonly learned using unsupervised density estimation techniques such as k-means, k-medoids or hierarchical k-means [19]. Depending on the properties of the descriptors, the estimated distribution and respective classes might however not be a good representation of the high-dimensional descriptor space, leading to misclassifications. Binary descriptors provide real-time performance and are therefore a prefered choice for BoW based frameworks for mobile phone and robotic applications. Even though the mean of a set of binary descriptors is not well defined, Galvez *et al*. [8] successfully applied k-means++ clustering to binary descriptors which were converted to floating point values during the clustering followed by rounding to form the final centroids. The resulting BoW vectors were then used in a term-frequency-inverse-document-frequency (TFIDF) scheme.

**Voting and efficient nearest neighbor search:** Voting-based systems, such as our method, directly search the nearest neighbors of query image descriptors to identify potential matches to the database. Jegou *et al*. [11] discussed and demonstrated the performance gain of voting-based systems, which results from the avoidance of artifacts inherent to the quantization step needed by BoW. While voting is more computationally expensive, Stewenius *et al*. [22] recently demonstrated that such algorithms scale to billions of images.

Both methods described above rely on the concept of distance in the space of image descriptors as a fundamental building block. In voting-based systems, this is needed for efficient and precise k-nearest-neighbor (kNN) search. Given the computational constraints on mobile devices we would like to use binary descriptors produced by the BRISK algorithm [13] in our application. However, for binary descriptors, such a kNN algorithm is not straightforward to design due to the high dimensionality and the nature of the binary descriptor space [24]. Consequently, we project binary BRISK description vectors into a dimensionality-reduced real-valued space, which allows accurate and fast kNN search. This is similar to the technique used by Bosse *et al*. [4] for laser keypoints.

**Forming images into places:** Until recently nearly all place-recognition frameworks separated the image-stream from the camera into single frames or small groups of images which are fed to an image retrieval pipeline. For example, the FABMAP algorithm by Cummins *et al*. [5] initially considers places to be individual images. During operation it takes a sequence of *non-overlapping* images and determines if each one belongs to a new or a revisited place. It does so by computing and comparing the probability of a binary BoW vector being generated by either a previously seen place or by a new place represented by a background model. The main limitation of this approach is that, on the next visit through the same environment, the corresponding image might be captured "between" two places generated during the first visit. Both places now attract half of the probability mass and therefore neither reaches the thresholds for successful data association. This situation gets worse from visit to visit as more "new places" from neighboring locations are added to the database. Galvez *et al*. [8] proposed to mitigate this issue by forming "islands" of places by grouping query results that are close in time or have been found to belong to the same place before the voting step. A similar approach was proposed by Mei *et al*. [15] and more recently by Stumm *et al*. [23], where places are formed by grouping query results based on covisibility; images that share more than 50% features are grouped before the final voting step, where the former uses TFIDF and the latter evaluates the FABMAP model. Murphy *et al*. [17] proposes to cluster images based on place topics to narrow the search space. An approach which breaks up the discretization of places is the work by Maddern *et al*. [14] who employ a particle filter for SLAM and the FABMAP model to evaluate the probabilities of observing landmarks based on entire pieces of trajectory using a motion model. The approach closest to a "placeless" representation is the work by Milford *et al*. [16] who perform correlation-based matching on entire sequences of down-sampled images, instead of looking at individual images or descriptors. Given, however, that the images are collected using time-based sampling, the trajectories have to be traversed with similar velocities on both visits. It would be straightforward, however, to apply the distance-based sampling strategy from our proposed algorithm to mitigate this problem in the sequence matching of Milford *et al*. [16].

The proposed work contrasts with the discussed approaches in several ways:

- We propose to sample images by approximate distance along the trajectory, rather than by covisibility or time.
- We do *not compute a BoW representation*, but rather add individual descriptors into our search index.
- We formulate the place recognition problem as a 2D

density estimation in vote space where we take into account the kNN votes of every query descriptor.
- We find loop-closure candidates using statistical tests instead of thresholds. The query evaluation does not rate single images but integrates loop-closures on dynamically defined regions.
- Queries are not done on a single image basis, but consider the votes of all images in parallel as a batch on pieces of trajectory.

Most existing state of the art approaches that aim at breaking up the discretization of places require setting a threshold on the number of images to combine into a place. Values for these thresholds are commonly not valid for all environments or use heuristics which require manual tuning. These thresholds are however not the only ones commonly used in place-recognition pipelines, from which many are implicit, such as: number of classes for K-means, starting values for k-means, depth and width of the hierarchical trees, training data for vocabularies, spacing of images along the trajectory etc. In contrast, the proposed algorithm limits the number of parameters to a minimum where all but two are derived from statistical tests on the data.

## 2. Methodology

During training, our algorithm requires typical image sequences captured from a moving camera in which we track BRISK descriptors. Tracked and non-tracked features are used as exemplars to produce a transformation that reduces the descriptor dimensionality and provides a distance test equivalent to the likelihood ratio test statistic to distinguish true matches from a background distribution (Section 2.1).

During testing, the input to our algorithm is a sequence of images and odometry information. The odometry information is only needed to specify the distance along the path so it can come from high-fidelity visual-inertial-odometry [10] or low-fidelity wheel odometry. We extract BRISK descriptors from the input images and build a KD-tree from the projected descriptors. After construction, we query the k-nearest neighbours of each descriptor and map each match to a 2-dimensional point indexed by the respective distances (query $\times$ database) along the path. Regions of high density in this space represent good candidates for loop closures.

We rotate the match space into "placial" indices such that path-aligned match regions correspond to vertically or horizontally aligned regions of high density. These regions can then be easily segmented by recursive axis-aligned splits using statistical tests on the vote densities. This decomposition is data driven and performed at query time. Each segmented region of sufficient density indicates a correspondence between two places along the path.

### 2.1. Descriptor dimensionality reduction

We learn a linear projection that transforms the binary descriptors $\mathbf{x}_i$ to a lower dimensional space, while maximizing the separability between matching and non-matching descriptor pairs. We take the technique which was used by Bosse *et al.* [4] for laser keypoints and apply it to binary BRISK [13] descriptors. The projection is designed such that the $L_2$ distance between descriptors in the projected space matches the Likelihood Ratio Test (LRT) statistic. The LRT is the hypothesis test with the maximum power for a given maximum false-positive rate [18].

Under the assumption that the descriptors are i.i.d. the central limit theorem suggests that a sufficiently large number of samples will be uniformly distributed. Therefore the difference between descriptors can be modeled with a multivariate Gaussian distribution, which in return leads to the conclusion that the likelihood ratio test statistic, $\Lambda(\cdot)$, can be computed as follows:

$$\Lambda(\mathbf{x}_a - \mathbf{x}_b) := \frac{|\mathbf{\Sigma}_U|^{1/2}}{|\mathbf{\Sigma}_M|^{1/2}} \frac{e^{\{-0.5(\mathbf{x}_a-\mathbf{x}_b)^T \mathbf{\Sigma}_M^{-1}(\mathbf{x}_a-\mathbf{x}_b)\}}}{e^{\{-0.5(\mathbf{x}_a-\mathbf{x}_b)^T \mathbf{\Sigma}_U^{-1}(\mathbf{x}_a-\mathbf{x}_b)\}}} \quad (1)$$

where $\mathbf{\Sigma}_M^{-1}$ and $\mathbf{\Sigma}_U^{-1}$ are the respective covariances computed from a training set of matched ($M$) and unmatched ($U$) descriptor differences and $\mathbf{x}_a$ and $\mathbf{x}_b$ are individual descriptor vectors. Taking the logarithm on both sides allows us to drop the scaling constants and simplify the computation. The resulting distance, $D$, satisfies the LRT [4] since:

$$\begin{aligned} -log(\Lambda(\mathbf{x}_a, \mathbf{x}_b)) &\propto (\mathbf{x}_a - \mathbf{x}_b)^T (\mathbf{\Sigma}_M^{-1} - \mathbf{\Sigma}_U^{-1})(\mathbf{x}_a - \mathbf{x}_b) \quad (2) \\ &= (\mathbf{x}_a - \mathbf{x}_b)^T (\mathbf{A}^T \mathbf{A})(\mathbf{x}_a - \mathbf{x}_b) \quad (3) \\ &= \underbrace{\|\mathbf{A}\mathbf{x}_a - \mathbf{A}\mathbf{x}_b\|_{L_2}}_{=:D} \quad (4) \end{aligned}$$

The matrix $\mathbf{A}$[1] is the linear transformation of the descriptors to a subspace where the LRT is a simple threshold on Euclidean distances. After applying this linear descriptor transformation we perform dimensionality reduction by removing the dimensions with the lowest signal-to-noise ratio. This increases the efficiency of the kNN search without significant loss of precision as shown in Figure 2. We obtain the required training data of matching and non-matching descriptors from feature tracking in a visual SLAM framework.

### 2.2. Placeless descriptor vote integration

To allow a placeless representation of the environment, we do not follow the standard BoW model for image retrieval. Instead we use a descriptor based voting scheme in which we perform kNN search in the low dimensional descriptor space we obtained from the descriptor transformation and dimensionality reduction. For the moderate sized

[1] $\mathbf{A}$ is computed using the singular value decomposition.

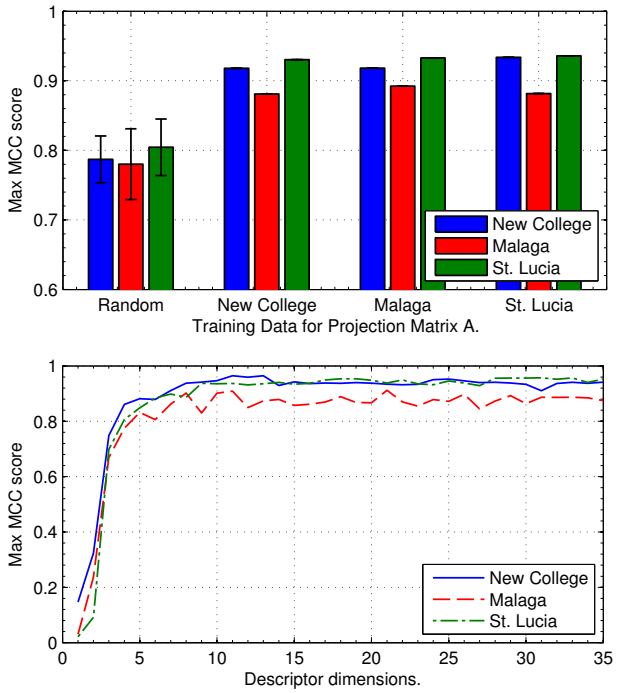


Figure 2: The maximum Matthews correlation coefficient (MCC) score as a function of different projection matrices (top) and the dimension of the projected space (bottom). Our method for learning the projection matrix outperforms a random projection matrix (error bars show 1 $\sigma$ of 10 different random matrices), while the difference between matrices learned from unrelated datasets has little influence. Most of the descriptor space can be captured in a 10 dimensional subspace. Higher dimensions of the projected descriptor entail little gain in performance, while drastically increasing the complexity of the kNN search.

databases (≈10,000 key-frames) a direct search for nearest neighbors using the *libnabo* [7] KD-Tree library performs well. Therefore we do not need to apply any approximations using hashing or vocabulary-based quantization for finding nearest neighbors. The algorithm processes a dataset as batch in which all descriptors are queried against the database. The kNN from this search form samples in a 2D space of votes (spanned by the query-descriptors and their matches). An online approach is feasible, but left for future work. In the following sections of this paper, we describe the transformations and decompositions of this space which allow us to find regions in which the vote density is high and thus signals a closeness in appearance between different trajectory segments.

### 2.3. Trajectory aligned vote space segmentation

To evaluate the place-recognition problem in batch form, we retrieve the $k$-nearest neighbors from the database for each descriptor[2]. Initially, the votes for each feature match can be represented by points in a 2D space indexed by the times (or frame number) of the corresponding images. However, since the individual votes are quite sparse and noisy, we would like to aggregate nearby votes in this space. If the places were predefined, then the vote space could be represented as a matrix where each element contains the sum of all the votes contributing to the corresponding place match. The places would need to be big enough such that suffiently many votes are captured and matching places could be distinguished from non-matching places that receive outlier votes. If the places, however, are too large, then the vote score would be unduly influenced by non matching parts of the places. (See Figure 3 for an illustration of the process). Since we do not predefine any places, we treat the votes as observations from a 2D continuous probability density in the space of place matches. The two dimensions of this space are spanned by the database and the query descriptors as the path is traversed. Matching places should have a higher density of votes whereas non-matching regions should have a low density of randomly matching votes. To mitigate artifacts to varying velocities while traversing the path, we remap the coordinates from time to distance along the path, $d$, and assign each feature a weight, $w$, such that the sum of weights over a unit distance is constant. Each vote is weighted by the product of the weights from the matched descriptors. Aggregations of votes in this space correspond to integrals over rectangular regions. One can see in Figure 3b that large integration windows will include a significant fraction of background or false positive votes unless the windows would be aligned to capture the density of the underlying distribution. However finding the parameters of this alignment without employing ad-hoc thresholds is hard. Therefore we define the *"placial"* coordinates, $(x, y)$ such that axis aligned integration windows can model large sections of matches from places traveling in the same direction (vertically aligned) or traveling in opposite directions (horizontally aligned) (Figure 3c). In summary, when query descriptor $a$ votes for database descriptor $b$, we compute the weight, $w_{ab}$, and *placial* coordinates, $(x_{ab}, y_{ab})$, as

$$w_{ab} = w_a * w_b, \quad x_{ab} = d_a + d_b, \quad y_{ab} = |d_a - d_b|. \quad (5)$$

Given this distribution of weighted votes expressed in *placial* coordinates we employ a decomposition of the space that allows matching corresponding pieces of the trajectory as described in the following section.

### 2.4. Determination of place sizes at query time

In order to find an appropriate window size for vote integration which is both dynamically determined at query time and takes into account the local density of the background process, we suggest a recursive tree-based segmentation algorithm driven by statistical tests. Employing statistical

[2]See experimental section 2.4 for the impact of $k$.

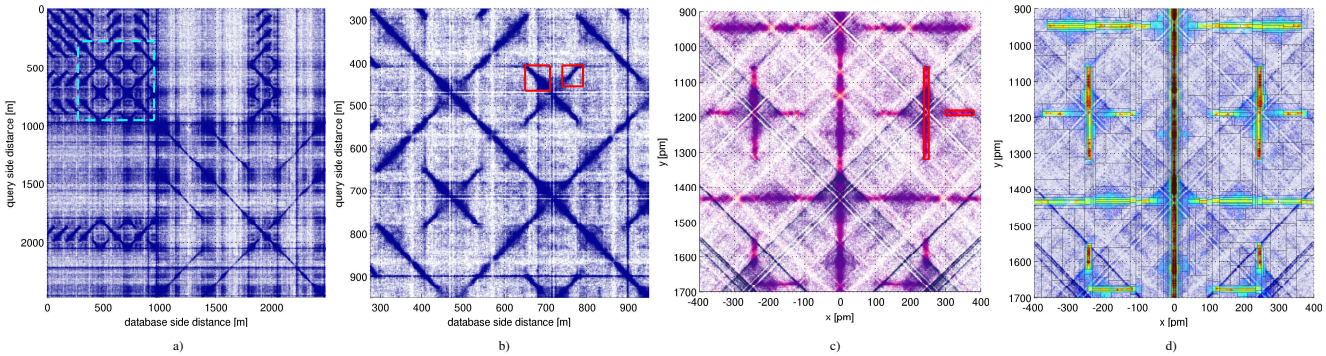


Figure 3: The vote space at different stages in our algorithm: We start with the raw landmark votes a) full, b) callout. Axis aligned integration windows (red) don't capture the density well. The vote space is transformed to allow axis aligned integration (c), the coloring denotes the number of descriptors per image. After the tree decomposition using statistical tests, the loop-closure candidates are separated and scored by their weighted vote density d).

tests that take into account the local density has the advantage that we do not introduce a dependency on the number of detected features or the scene appearance. Thereby regions with richer structure (resulting in stronger visual features) are equally weighted to other parts of the trajectory, avoiding bias in the loop candidates towards scenes with rich texture which many other approaches suffer from.

We aim to test the null hypothesis, $H_0$, that the landmark votes within a particular region of the query-result belong entirely to one of the two classes (foreground or background). If the test fails, we split the region into sub-regions. Natural non-parametric test statistics are the Kolmogorov-Smirnov, Cramer von Mises, and Kuiper's statistic types [6, 12] which are all based on the deviation of the empirical distribution from the theoretical distribution.

We found the Kolmogorov-Smirnov (KS) statistic is well suited in identifying the best place to split the integration window, but the very similar Kuiper's (K) statistic is better for determining whether or not to split the window. For a given integration region within the rotated space, we evaluate the statistics for the $x$ placial coordinates and $y$ coordinates separately to determine the optimal axis and location to split the region. Let $V_x = \{x_i | i = 1 \dots n\}$ be the set of $x$ coordinates of all votes inside the region. We assign a weight, $w_i$, as in Eqn. (5) to each vote to account for the appearance of the scene. We then compute $F_n(x)$, the empirically weighted cumulative distribution function (CDF) for the $x$-axis as

$$F_n(x) = \frac{1}{\sum_i w_i} \sum_i^n w_i I_{x_i \leq x}, \tag{6}$$

where $I_{x_i \leq x}$ is the indicator function which is equal to 1 if $x_i \leq x$ or 0 otherwise. A similar derivation yields the CDF of the $y$ axis. The Kolmogorov-Smirnov statistic for a given CDF $F(x)$ is

$$T_x = \sup_x |F_n(x) - F(x)|, \tag{7}$$

where sup is the supremum of the set of differences, and we compare to the uniform distribution

$$F(x) = (x - x_{\min})/(x_{\max} - x_{\min}). \tag{8}$$

As can be seen from Figure 4 the value of $T_x$ is maximal at the boundary of background and foreground distributions (indicated by the black vertical line), and indicates the optimal place to split the space into subregions. Apart from the splitting location, we would also like to determine which axis to optimally split next. Since the value of the KS-statistic is somewhat dependent on the location of the boundary between distributions, we instead use Kuiper's statistic [12] to find the best dimension to split. Kuiper's statistic is related to the KS-statistic, but is invariant to cyclic shifts of the data:

$$K_x = \sup_x \left(F_n(x) - F(x)\right) - \inf_x \left(F_n(x) - F(x)\right). \tag{9}$$

Where inf is the infinimum of the set of differences. The axis with the larger $K$ is split at the location of the maximum $T$. See Figure 4 for a depiction[3] of this process. We continue recursively splitting the space at the dimension given by the Kuiper's statistic (and location given by the KS-statistic) until the following condition is met:

$$\sqrt{N} \max(K_x, K_y) < K_s, \tag{10}$$

where $K_x$, $K_y$ denote the Kuiper statistics for the $x$ and $y$ axis respectively and $N$ denotes the number of samples in the current region. Figure 5 shows that the parameter has a wide area of values with high performance over different datasets.

After the K-statistic suggests no further splitting of the distribution, the resulting regions in the vote space represent candidates for overlapping trajectory pieces. The mean vote

[3]The votes in this plot are down sampled by a factor of 10 for viewing convenience.

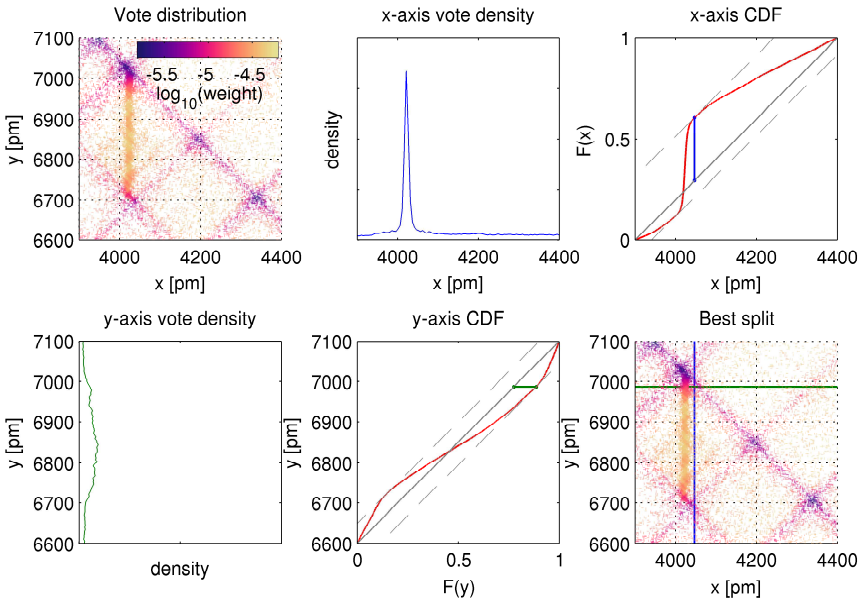


Figure 4: The Kolmogorov-Smirnov (KS) and Kuiper statistics per axis for a region of the vote distribution. The KS statistic identifies the location of the optimal split, where as Kuiper's statistic is used to chose the dimension to split on. The colors in the vote distribution correspond to the vote weights (logarithmically scaled), which we use to compute weighted empirical CDFs.

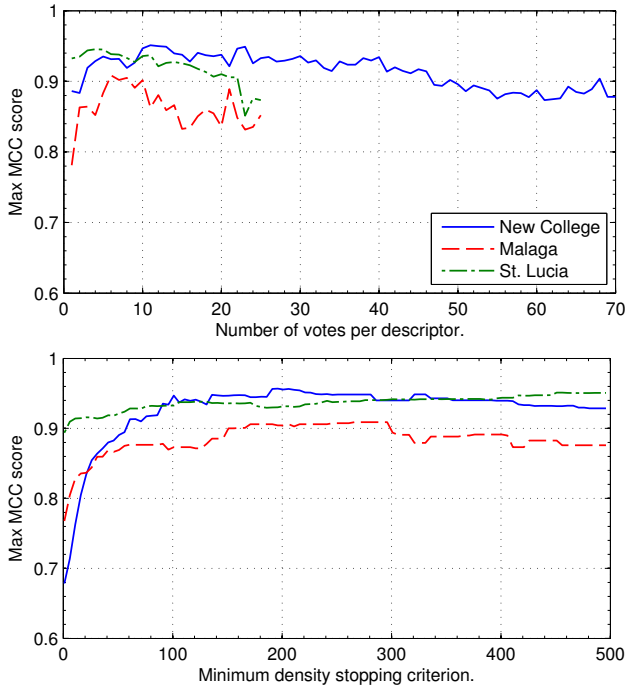


Figure 5: Influence of two main parameters on the maximum MCC score. The number of votes per query descriptor (top): Additional votes boost noise and subsequently cause false alarms. The density stopping threshold $K_s$ of the vote decomposition process (bottom): Lower density thresholds correspond to over segmentation of loop-candidate regions.

density in these regions can be used as a score to determine which candidates should be kept for further consideration, e.g. passed to a subsequent geometric-verification step.

## 3. Experiments

### 3.1. Experimental setup

We compare our algorithm against a BoW pipeline using BRISK [13] descriptors and TFIDF scoring similar to the implementation of [8] and against several voting-based algorithms using projected descriptors[4] where we aggregate the votes on a per image basis [22, 11], on places formed by images close in time [8] and covisibility [23]. We do not compare to the FABMAP 2.0 [5] algorithm itself or apply its probabilistic weighting to the raw votes such as done for instance by Stumm *et al*. [23] or Glover *et al*. [9]. We see this as an additional step which can be applied to the votes independent of the way the votes got accumulated into places. While FABMAP can perform data-association to merge observations from multiple visits into a place, we see this output different than the raw retrieval evaluated in this paper, and therefore not directly comparable. Instead, we point readers to the FABMAP evaluation done by Glover *et al*. [9] who used two of the datasets that are also evaluated in this paper. Similarly we also do not use geometric verification in the performance evaluation for any of the algorithms as it would make the experiments be less informative w.r.t. the proposed method. Clearly any actual implementation

[4] All voting-based algorithms use the same number of votes $k$ per query descriptor.

should make use of the improved performance offered by a geometric verification step.

We use three standard public outdoor datasets with moderate trajectory lengths namely: "New College" (panoramic, Ladybug, 2.4 km) [21], "Malaga Urban Extract 10" (stereo forward facing, 5.8 km) [3] and "St. Lucia" (mono forward facing, webcam, 17.6 km) [9]. To the best of our knowledge we chose parameters for all algorithms for them to give the best performance and then fixed them for all datasets.

To generate ground truth loop-closure candidates we use GPS and, if available, odometry information to determine when the same part of the trajectory is revisited. For the *Malaga* and *St. Lucia* datasets we directly use the GPS measurements as they are mostly accurate. For the *New College* dataset however the GPS has many outliers and is often interrupted. We therefore apply a robust least-squares optimization to fuse the GPS measurements with the available wheel-odometry to get a more accurate and complete trajectory.

### 3.2. Precision recall

To mitigate the difficulty of deciding which parts of the ground truth trajectory should definitely be loop closures and which should not, we employ two evaluation thresholds: We identify true positives using pairs of points on the ground-truth trajectories that are closer than a threshold $gt_{near}$. Parts of the trajectory that are farther apart than a second threshold $gt_{far}$ are marked as true negatives. The pairwise parts of the trajectory whose distance falls between the two thresholds are "don't care" regions and not used in the evaluation since we cannot be certain whether those sections will induce reliable loop closures or not. Given that the Malaga dataset has a front-facing camera, we additionally mask out all regions where the same trajectory is traversed in the opposite direction.

Besides the proposed "Placeless" algorithm, the precision-recall (PR) plots (Figure 6) show different methods for aggregating images to places. For the algorithm "Single image place" we evaluate every image as an individual place. This is similar to the approaches found in image retrieval [22, 11]. The performance of an algorithm that forms places from images close in time [8] is denoted by "Time based place". Forming places by accumulating votes from images connected in the covisibility graph [23] corresponds to the curve "Covisibility based place".

Due to the dynamic determination of place sizes, the placeless approach consistently reaches the highest values with very high recall even at $90\%$ precision. We would like to point out the remarkable $93\%$ recall at $90\%$ precision on the 'St. Lucia' dataset.

## 4. Conclusion and Future Work

We have presented a method for batch *placeless place-recognition* using projected binary descriptors and a kNN voting scheme with a loop-candidate segmentation using statistical-tests. Instead of scoring individual images spaced by time, we formulate place recognition as a continuous 2D probability density estimate in the space of matches along path distance. We apply voting and scoring based on all query descriptors jointly and drop the widely used keyframe-based discretization of places. This allows us to handle different sizes of places, indoor and outdoor environments as well as perceptual aliasing in a *continuous* and *placeless* way. Our approach contrasts with many existing works that aim at a "placeless" representation where covisibility and a set of thresholds are used to control the number of images to combine for modeling the appearance of a scene. Instead we rotate the vote space to obtain a *"placial"* index, in which we can segment and integrate places without considering boundaries between individual images. This way we can build a truly *placeless place-recognition* which at query time determines the number of descriptors to combine from both query and database for best representation of the scenes' appearance. Statistical tests control the segmentation of the vote space into loop-candidates. In fact these tests take into account the local background vote distribution and hereby capture the local structure of the environment in a better way than what a global threshold could deliver.

While this work focused on a batch method for place-recognition, we will focus on the development of an online-method for future work.

## 5. Acknowledgments

The research leading to these results has received funding from Google's project Tango.

## References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. 2

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer Vision (ECCV), 2006 IEEE European Conference on*, pages 404–417. 2

[3] J.-L. Blanco, F.-A. Moreno, and J. González-Jiménez. The málaga urban dataset: High-rate stereo and lidars in a realistic urban scenario. *International Journal of Robotics Research*, 33(2), 2014. 7

[4] M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2d lidar maps. *Robotics and Autonomous Systems*, 57(12):1211–1224, 2009. 2, 3

[5] M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123. 2, 6

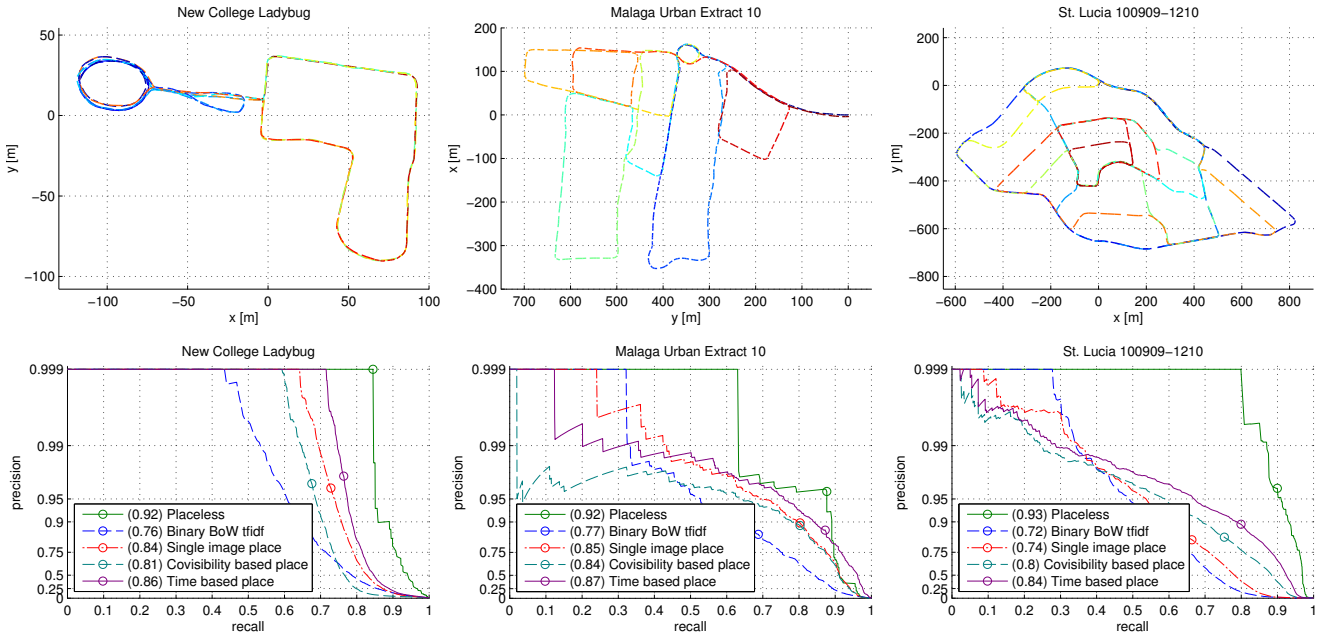


Figure 6: Comparison against related work on the 'New College', 'Malaga' and 'St. Lucia' datasets for precision $p \in [0, 1]$ on a log scale. The maximum MCC is given inside parentheses in the legend and its location denoted by a circle.


[6] D. A. Darling. The kolmogorov-smirnov, cramer-von mises tests. *The Annals of Mathematical Statistics*, pages 823–838, 1957. 5

[7] J. Elseberg, S. Magnenat, R. Siegwart, and A. Nüchter. Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration. *Journal of Software Engineering for Robotics*, 3(1):2–12, 2012. 4

[8] D. Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *Robotics, IEEE Transactions on*, 28(5):1188–1197, October 2012. 2, 6, 7

[9] A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth. Fab-map+ ratslam: appearance-based slam for multiple times of day. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3507–3512. 6, 7

[10] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014. 3

[11] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Computer Vision (ECCV), 2008 IEEE European Conference on*, pages 304–317. 2, 6, 7

[12] N. Kuiper. Tests concerning random points on a circle. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappe*, Series A(63):38–47, 1960. 5

[13] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. 2, 3, 6

[14] W. Maddern, M. Milford, and G. Wyeth. Cat-slam: probabilistic localisation and mapping using a continuous appearance-based trajectory. *The International Journal of Robotics Research*, 31(4):429–451, 2012. 2

[15] C. Mei, G. Sibley, and P. Newman. Closing loops without places. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3738–3744. 2

[16] M. J. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. 2

[17] L. Murphy and G. Sibley. Incremental unsupervised topological place discovery. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, page to appear. 2

[18] J. Neyman and E. S. Pearson. *On the problem of the most efficient tests of statistical hypotheses*. Springer, 1992. 3

[19] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*, volume 2, pages 2161–2168. 2

[20] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision (ICCV), 2003 IEEE International Conference on*, pages 1470–1477. 2

[21] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *The International Journal of Robotics Research*, 28(5):595–599, 2009. 7

[22] H. Stewénius, S. H. Gunderson, and J. Pilet. Size matters: Exhaustive geometric verification for image retrieval. In *Computer Vision (ECCV), 2012 IEEE European Conference on*, pages 674–687. 2, 6, 7

[23] E. Stumm, C. Mei, and S. Lacroix. Probabilistic place recognition with covisibility maps. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4158–4163. 2, 6, 7

[24] T. Trzcinski, V. Lepetit, and P. Fua. Thick boundaries in binary space and their influence on nearest-neighbor search. *Pattern Recognition Letters*, 33(16):2173–2180, 2012. 2