Point cloud descriptors for place recognition using sparse visual information

Conference Paper · May 2016

DOI: 10.1109/ICRA.2016.7487687

Titus Cieslewski
University of Zurich

Elena Stumm
ETH Zurich

Simon Lynen
ETH Zurich

Roland Siegwart
ETH Zurich

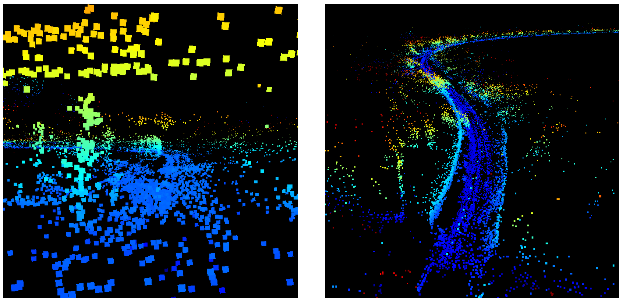
Point Cloud Descriptors for Place Recognition using Sparse Visual Information

Titus Cieslewski*, Elena Stumm*†, Abel Gawel*, Mike Bosse*,
Simon Lynen*‡ and Roland Siegwart*
*Autonomous Systems Lab ETH Zurich, †Robotics&Interactions, LAAS-CNRS, ‡Google Inc.

***Abstract*— Place recognition is a core component in simultaneous localization and mapping (SLAM), limiting positional drift over space and time to unlock precise robot navigation. Determining which previously visited places belong together continues to be a highly active area of research as robotic applications demand increasingly higher accuracies. A large number of place recognition algorithms have been proposed, capable of consuming a variety of sensor data including laser, sonar and depth readings. The best performing solutions, however, have utilized visual information by either matching entire images or parts thereof. Most commonly, vision based approaches are inspired by information retrieval and utilize 3D-geometry information about the observed scene as a post-verification step. In this paper we propose to use the 3D-scene information from sparse-visual feature maps directly at the core of the place recognition pipeline. We propose a novel structural descriptor which aggregates sparse triangulated landmarks from SLAM into a compact signature. The resulting 3D-features provide a discriminative fingerprint to recognize places over seasonal and viewpoint changes which are particularly challenging for approaches based on sparse visual descriptors. We evaluate our system on publicly available datasets and show how its complementary nature can provide an improvement over visual place recognition.**

I. Introduction and Related Work

Simultaneous localization and mapping has evolved as a central paradigm to provide a solution for robotic navigation without relying on external sensors. Given noise in the sensor signals, modeling inaccuracies and errors due to linearization of the inherently non-linear system models, even the most accurate state estimators [1, 2] are subject to drift over distance traveled. In recent years the community has shown substantial advances in camera as well as camera-imu based SLAM approaches allowing visual-inertial online mapping [3, 4] over large distances. However, place recognition still remains to play a vital role in eliminating positional error accumulated over the course of a robotic operation by providing loop closures. As a result, appearance based approaches to place recognition have been developed in parallel to provide the required data associations. Many visual place recognition techniques utilize bag-of-words algorithms borrowed from the text and image retrieval communities [5, 6], along with adaptations to probabilistic frameworks which deal with perceptual aliasing [7]. In these approaches, local feature descriptors from an image are vector quantized and aggregated in a sparse histogram which can subsequently be used to represent an image in a compact and invariant manner. Extensions to such frameworks include working with binary descriptors for improved computational efficiency [8], aggregating local features through a Fischer vector for improved performance and efficiency [9], and grouping features over covisible viewpoints for improved context [10]. Again taking inspiration from image-search approaches [11], recent methods which utilize voting based frameworks have shown to perform well even at a very large scale [12]. Here the database contains individual descriptors from all previously observed images and the search focuses on finding the n-nearest neighbors to the query descriptors. Every nearest neighbor votes for a particular database image; with most highly voted images being forwarded to geometric verification and pose recovery [13, 14, 15]. For all the aforementioned approaches, geometric constraints play a crucial role as a post-verification step prior to performing data association. However they only implicitly use the underlying structural information instead of explicitly utilizing it to guide the place recognition query. Instead minimal geometric solvers such as perspective 3, 5 and N point algorithms embedded in a RANSAC loop are used to filter outliers from the loop closure frontend. While there exist approaches which include geometric verification directly during the query of the index [12], only a subset of the work [16, 17, 18] loosely encode 3D-structure explicitly for scoring candidate images. On the other hand, some methods therefore rather rely on holistic image descriptors which implicitly require geometric consistency and have been shown to be robust to appearance changes such as weather variations, but lack viewpoint invariance as a result [19, 20].



(a) An overhead structure and an approaching train.

(b) A curve along a textured wall and some vegetation.

Fig. 1: Landmarks extracted from a visual SLAM pipeline form the basis of the Neighbour-binary landmark density descriptor (NBLD) – a novel structure descriptor useful for place recognition. The proposed local signatures provide a complementary source of information to visual feature descriptors used in related work. We show that in particular for scenes with challenging conditions due to appearance changes, the proposed method can provide an advantage over using visual cues only.

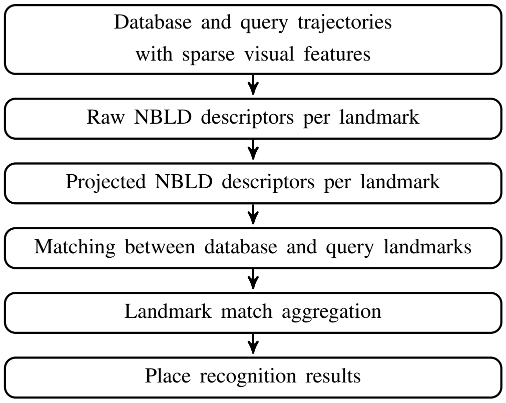

Database and query trajectories
with sparse visual features
Raw NBLD descriptors per landmark
Projected NBLD descriptors per landmark
Matching between database and query landmarks
Landmark match aggregation
Place recognition results

Fig. 2: The place recognition pipeline first assigns raw and then projected descriptors to sparse visual feature landmarks, matches database and query landmarks according to those descriptors and finally aggregates the matches to determine if a place has been recognized.

Given the rich information contained in 3D structure we propose to use the information as a signature for place recognition to complement vision based approaches. In particular we adopt approaches from laser based place recognition, where structural descriptors such as the Gestalt descriptor [21] enable recognizing places solely based on 3D points. Due to the fact that a single point is not discriminative, structural descriptors rely on the extraction of higher-level features from point clouds, such as surface normals, planes and histograms over densities. As an example, *Point Feature Histograms* (PFH) [22] use surface normals to derive characteristics of all pairs of points within a neighbourhood and use histograms of these characteristics to describe the neighbourhood. The *Signature of Histograms of Orientations* (SHOT) [23] descriptor creates cells in a normalized sphere, and derives the descriptor from histograms of normals in each cell. Sparse visual features however do not provide the required point density to estimate surface normals, such that neither of those descriptors can be directly applied to our approach.

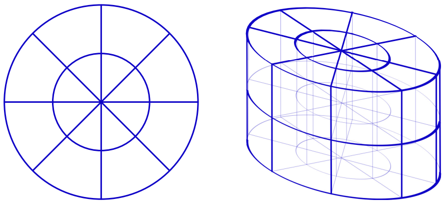
The 3D *Gestalt* descriptor [21], however, can be evaluated without any surface normals, and thus serves as a starting point for the work presented in this paper. Taking inspiration from Gestalt we propose a novel 3D point-descriptor which is better suited for place recognition from sparse point clouds such as those from Structure from Motion (SfM) or SLAM. To demonstrate the complementary nature of the proposed algorithm, we show how our approach outperforms vision-only place recognition in particular on datasets which are challenging due to strong appearance changes.

II. Methodology

In order to perform place recognition, a pipeline as outlined in Figure 2 is implemented. The following sections describe the individual parts of the pipeline in detail.

A. Sparse visual features

The 3D landmarks that we use for our descriptors can be obtained from Structure from Motion algorithms with sparse data, such as Visual-Inertial Odometry [4]. Typically, such landmarks are obtained by tracking salient visual features, such as corners, across a sequence of images, then triangulating their position based on the estimated positions from which the images were taken.



(a) Gestalt bins (b) NBLD bins

Fig. 3: Bins used in the Gestalt and in the NBLD descriptor. The amounts of bins in each direction (azimuthal, radial, vertical) are parameters of the descriptors.

B. Landmark description

With dense 3D point clouds, keypoint detection can be used to select relevant areas of interest within a point cloud, which then serve as representative features to summarize the environment (analogously to keypoint detection in 2D image analysis). For example, the work of Hänsch et. al [24] describes and compares two methods of 3D keypoint detection, namely *normal aligned radial feature* (NARF) keypoints and 3D-SIFT keypoints, which analyze curvature properties of the points to select relevant and stable locations within the point cloud. However, due to the already sparse nature of sparse visual feature landmarks, no such keypoint detection is necessary in our case. We hence evaluate descriptors for all points in our point cloud. We then seek to calculate representative and discriminative structural descriptors for each individual landmark l_i, based on the distribution of landmarks surrounding it, which is summarized as follows:

1) Determine all landmarks $L_{i,\Omega}$ that are within a neighbourhood volume Ω_i around the position $\vec{p}_i$ of landmark l_i.
2) Express the neighbouring landmark positions $\vec{p}_L = \{\vec{p}_j | j \in L_{i,\Omega}\}$, in a normalized frame of reference T_N.
3) Compute the descriptor in the normalized frame, using the positions of the neighbouring points $\vec{p}_L$.

Since our descriptors will not be invariant to transformations of $\vec{p}_L$ (such as rotation), unlike for instance the PFH descriptor [22], we need to perform a normalization transformation T_N on $\vec{p}_L$. Here, we can make use of two assumptions: Firstly, since the position of the described landmark $\vec{p}_i$ with respect to $\vec{p}_L$ is known, we can center $\vec{p}_L$ around $\vec{p}_i$. Secondly, point clouds can be aligned according to the gravitational axis, such that only one rotational degree of freedom is unknown (yaw). To select a yaw rotation for normalization, we use the method of Bosse and Zlot [21]. First, the sample covariance C_L of $\vec{p}_L$ is evaluated. Let $\vec{e}_{C,0}$ be the eigenvector of lowest eigenvalue of C_L. Then, T_N is chosen such that the projection of $\vec{e}_{C,0}$ onto the horizontal plane is collinear with the x-axis. Since this can be achieved in two ways, N is further chosen such that the bearing vector from $\vec{p}_i$ to the position of the first observer of l_i has a positive x-coefficient.

Similar to the Gestalt descriptor proposed by Bosse and Zlot [21], we draw landmarks from a cylindrical neighbourhood Ω_i aligned with the gravitational axis. The Gestalt descriptor then uses bins in the horizontal plane as shown in figure 3a, and uses mean height and height variation within each bin to describe l_i. In contrast, we try to incorporate more information about the vertical landmark distribution. To that end, we choose a vertical extent of Ω_i, and introduce additional vertical subdivision, as shown in figure 3b. We then calculate the density of points within each bin, and combine the information in the form of a descriptor. In particular, we take inspiration from BRISK [25] and FREAK [26], that compare relative quantities between pairs of bins instead of using absolute values. This results in increased robustness, especially to changes in illumination. We aim for similar effects with respect to change in landmark density for our descriptor. Landmark density can change due to different illumination or different distances to structure. Instead of considering all $\binom{n}{2}$ possible bin relationships, only nearby bins are considered. In the end, our descriptor is obtained by subdividing the neighbourhood Ω_i of a landmark l_i with the bins shown in Figure 3b. Then, for each bin, the density of landmarks is calculated. Finally, for each bin the density of landmarks is compared to each adjacent bin in azimuthal, radial and vertical direction, and the results are then concatenated into a binary descriptor. Accordingly, we name our descriptor *Neighbour-binary landmark density descriptor*, or *NBLD* descriptor.

C. Place recognition: Definition

Place recognition can be defined as a function

$$x \in Q \rightarrow f(x) \in D \tag{1}$$

where Q denotes a query set and D a database set. We distinguish between two kinds of place recognition: On one hand there is localization, which is done between two trajectories, A and B. The places of one trajectory, say A, constitute the database D, while places of the other trajectory B constitute the query Q. On the other hand, there is loop closure, which is done within a single trajectory. Here, the query and database are defined by a given time t and location at that time $x(t)$,

$$Q(t) = \{x(t)\} \tag{2a}$$

$$D(t) = \{x(t')|t' < t - \Delta t\} \tag{2b}$$

where $\Delta t > 0$ is a minimal time difference between query and match to avoid self-queries.

D. Place recognition: Landmark voting framework

Since the focus of this work is on descriptors, place recognition will be evaluated using the recognition of landmarks based on descriptor matching. To infer places from matching landmark descriptors, we for now ignore recent advances in efficient place recognition pipelines and instead focus on voting-based methods that match query frames to database frames. In particular, we simply define Q and D to be composed of visual frames. Doing this allows better understanding of the key mechanisms in our pipeline and gives a more direct feedback on the performance of our descriptors specifically. The remainder of this section provides more details about each component in the pipeline.

1) Voting scheme: The work of Jégou et al. [11] compiled a survey of frame matching methods, and for our work, we have selected the described descriptor voting framework. In voting, each frame in the database, $d \in D$, is assigned a vote count $v(d)$ for each query frame x, and $f(x)$ is selected based on the number of votes each database frame d receives. At first, each landmark l_x observed as feature track by x casts votes for a set of candidate landmarks of the database, l_d. Then, each database landmark l_d gives as many votes as it has received from l_x to each of the database frames d it is observed from, resulting in each frame d in the database accumulating the final number of votes $v(d)$.

2) Landmark matching: In order for each query landmark l_x to cast its votes, a set of candidate matching landmarks needs to be retrieved from the database. These are the landmarks whose descriptors are the k nearest neighbours of l_x in Euclidian descriptor space. We use the *libnabo* KD-Tree library [27] for matching. As it is most efficient when executing multiple queries at the same time, we consolidate the loop-closure domains from (2b) such that

$$Q_i = \{x(t)|t \in \{i\Delta t, (i+1)\Delta t\}\} \tag{3a}$$

and

$$D_i = \{x(t)|t \in \{0, (i-1)\Delta t\}\} \tag{3b}$$

The result of this consolidated loop closure is equivalent to (2b) where the new Δt varies between Δt and $2\Delta t$.

Finally, to reject outliers, we introduce a threshold γ_d, such that landmark matches d_i are rejected if $\Delta d_i > \gamma_d \cdot \Delta d_k$, where Δd_i is the distance from query descriptor to d_i, and Δd_k is the distance to the k-closest descriptor.

3) Descriptor projection: To speed up the nearest-neighbour search, we project the raw descriptors into a lower-dimensional space. A principal component analysis (PCA) is applied, as it weighs the original dimensions according to their information content and orders the resulting dimensions in a way that expresses most variations with the least amount of dimensions. A more sophisticated approach would be to evaluate a projection, such that the distances between projected descriptors satisfy the likelihood ratio test (LRT), as introduced by Bosse et al. [28] and also used by Lynen et al. [14]. Instead of just normalizing the distribution of descriptors, this projection minimizes distances between matches and maximizes distances between non-matches, according to a provided ground truth. Due to the need for a ground truth, it can't be evaluated ad-hoc, and can be prone to over-fitting. Since it furthermore precludes the variation of descriptor parameters such as Ω_i extents and bin counts, we refrain from using it for the time being.

E. Place recognition: Decision making

Once descriptors are projected, matched, and votes are aggregated for database frames, the remaining task is to

	tp	fp	tn	fn
If database place d is assigned to query place q.				
If $\|\vec{p}_q - \vec{p}_d\| \leq r_e$	1	0	$\|D\| - 1$	0
Else, if $\exists d' \in D, \|\vec{p}_q - \vec{p}_{d'}\| < r_e$	0	1	$\|D\| - 2$	1
Otherwise	0	1	$\|D\| - 1$	0
If no database place is assigned to query place q.				
If $\exists d' \in D, \|\vec{p}_q - \vec{p}_{d'}\| < r_e$	0	0	$\|D\| - 1$	1
Otherwise	0	0	$\|D\|$	0

TABLE I: True positives, false positives, true negatives and false negatives accumulated for each query place q. Here, $\vec{p}_i$ are ground truth positions.

determine which database frame, if any, is considered recognized in a query frame. Before making this decision, two normalizations are performed to remove biases.

A first bias is that voting treats database vertices with many landmarks preferentially. To illustrate, assume perfect landmark matching with $k = 1$. Then, if one vertex observes a subset of landmarks that another landmark observes, and if the database is queried with the vertex with fewer landmarks, both database vertices receive the same amount of votes. As a remedy, we normalize the vote count of each database landmark by the amount of landmarks that it observes.

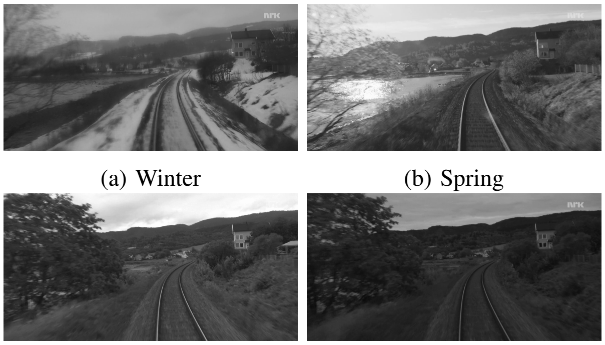
A second bias that only affects loop closure is that the database size grows throughout the trajectory. However, as we are always using k best matches for all landmarks, this means that database vertices matched at the beginning of the trajectory generally receive more votes than database vertices towards the end of the trajectory. If strong outliers are present at the beginning of the trajectory, these can have more votes than inliers at the end of the trajectory. To prevent this, we multiply all votes with the database size at the given moment.

After applying these normalizations, the database frame with most votes is selected for each query. However, it is not necessarily the case that a matching database frame exists for each query. In order to cope with this, a threshold on the vote count is used to reject matches.

III. Experiments

A. Quality measure

The quality of the place recognition is evaluated using precision-recall (PR) curves and the Matthews Correlation Coefficient (MCC), obtained by varying a threshold over the vote scores of the best match for each query frame. The evaluation is equivalent to an evaluation of a binary classification where the collection of items is $Q \times D$. Here, at most one matching location from the database is assigned to the query, $d \in D$ to $q \in Q$. We consider (q, d) to be retrieved and $(q, d') \forall d' \neq d$ not to be retrieved. Then the rules outlined in Table I are assigned to each query q. The resulting counts of true positives, false positives, true negatives and false negatives are then aggregated over all $q \in Q$. Note that the size of the database $|D|$ can vary over q in the case of loop closure applications. We use the same time-based consolidation here that we used for place recognition. Different ground truth tolerance radii r_e are hand-selected for different evaluation datasets, according to their spatial extent.



(a) Winter (b) Spring

(c) Summer (d) Fall

Fig. 4: Frame 9873 for each of the four seasons in the Norldandbanen datasets.

B. Evaluation datasets

Loop closure capabilities are evaluated using sequences from the KITTI visual odometry dataset [29]. For localization, and to show results on visually challenging datasets, the Nordlandsbanen videos [30], [31] are used. Here, we give a brief account on how we used these datasets.

1) KITTI datasets: From the KITTI suite, we use the 2012 Odometry/SLAM evaluation sequences 00 (3.7km in 7 minutes 50 seconds), 05 (2.2km in 4 minutes 47 seconds), and 06 (1.2km in 1 minute 54 seconds) due to the presence of loop closures. Since state-of-the-art visual odometry algorithms are shown to perform with down to sub-percent drift on the KITTI datasets [29], we assume that we have perfect visual odometry available. Thus, to obtain landmarks, we use the pose ground truth and triangulate 3D-landmarks from feature tracks.

2) Nordlandsbanen videos: The Nordlandsbanen videos have been previously used by Sünderhauf et al. [31] for evaluating the image-based SeqSLAM [20], and therefore did not require a precise pose estimate nor camera calibration parameters. However, in order to properly triangulate landmarks from feature tracks, this information is necessary, but not provided by NRKbeta [30]. What NRKbeta provides, however, are four location-synchronized videos and a corresponding GPS track. We extract the individual video frames and downsample them to a resolution of 960×480 pixels and grayscale. The GPS track by itself is not useful due to noise and the lack of attitude information. However, since the dataset was recorded on a train, we make assumptions on acceleration and orientation. We perform the pose and calibration estimates with the following steps (all numeric values used are explicited in Table II):

Time synchronization We plot trajectory updates concurrently with corresponding images for different offset hypotheses. We then manually pick a time offset $t_{v,g}$ such that the video time $t_v = t_{v,g} + t_g$, where t_g is the corresponding GPS measurement time, given that the GPS measurement timestamps in the provided summer dataset start at 0.

Position interpolation While camera measurements are available at 25Hz, GPS measurements are available at 1Hz. Thus, we interpolate the GPS measurements using cubic splines.

Video time of the first GPS measurement $t_{v,g}$	168s
Acceleration error term for consecutive frame position triplets $\{(\vec{p}_{i-1}, \vec{p}_i, \vec{p}_{i+1})\}$	$\frac{\vec{p}_{i-1} - 2\vec{p}_i + \vec{p}_{i+1}}{\frac{0.125m}{\Delta t}}$, $\Delta t = \frac{1}{25}s$
GPS error term for pose $\{\vec{p}_i\}$ and GPS measurement $\{\vec{m}_i\}$	$\frac{\vec{p}_i - \vec{m}_i}{3m}$
Focal length at 960×480	495
Principal point at 960×480	$(460, 202)$

TABLE II: A summary of parameters that can be used to extract sparse visual features from the Nordlandsbanen [30] videos.

Ω_i	Descriptor domain extents (incl. temporal). Includes the radius r, the height h and the temporal radius t.
$\|B\|$	Number of bins in all dimensions. Consists of number of azimuthal, radial and vertical bins (n_a, n_r, n_v).
s	Number of dimensions after projection
k	Number of matches for landmark matching
γ_d	Relative rejection threshold for matches
	vote count threshold
r_e	Ground truth radius

TABLE III: Parameters that can be varied in our system, in order of application.

	KITTI	Nordlandsbanen
Ω_i	$(r, h, t) = (9m, 18m, 60s)$	$(r, h) = (16, 18)m$
$\|B\|$	$(n_a, n_r, n_v) = (16, 4, 8)$	$(n_a, n_r, n_v) = (12, 2, 8)$
s	60	18
k	15	4
γ_d	0.8	1
r_e	$5m$	$20m$

TABLE IV: NBLD parameters used in the evaluation. The vote count threshold is varied over the PR curves.

Position smoothing We then smooth the trajectory using a non-linear optimization [32] with error terms penalizing accelerations and deviations from the GPS measurements.

Orientation recovery Next, we recalculate all pose attitudes such that the camera is always aligned with the velocity vector, and such that roll is 0.

Camera calibration estimation Finally, with the smoothed trajectory, we again use non-linear optimization with visual error terms to optimize the camera calibration. Since the used camera is a television camera, we assume no distortion.

While we don't claim that these steps result in the most accurate estimation of the environment, the resulting point clouds sufficiently represent a realistic underlying structure, as can be seen in Figure 1. Our current reverse-engineered parameters for the Nordlandsbanen dataset are summarized in Table II. We use only the first eight minutes ($\sim$ 10km) of the Nordlandsbanen trajectory for our evaluation.

C. *Visual-feature based place recognition*

As a comparison to the proposed algorithm, we put the BRISK descriptors from the same landmarks that we used for NBLD through an analogous place recognition pipeline. Instead of using a PCA projection, a projection matrix proposed by Lynen et al. [14] is used. The projection matrix has been trained using LRT training: It maximizes the descriptor distances from different tracks while minimizing the descriptor distances within the same track. We project to 10 dimensions, as this has been shown to reach best performance. Since each landmark is observed several times, where each 2D observation has its own descriptor, we match specific 2D observations of landmarks, based on their descriptors. For k, the number of nearest neighbours retrieved by each 2D observation, we use 5 neighbours, as this yields the best results. Now, since specific 2D observations, and not landmarks, are matched, the vote counting is adapted. The query vertex first votes for its 2D observations of landmarks. These 2D observations in the query then vote for 2D observations in the database that were also matched using nearest neighbour search. Finally, the 2D observations in the database do not vote for their single observer vertices directly, but instead for their corresponding landmarks first, who then vote for all their observers. This ensures that the vote counting is more robust to viewpoint changes.

D. *Parameter Configuration*

A list of free parameters in our system is given in Table III. For all of our experiments, we set the number of keypoints extracted per frame to 500. The remaining parameters are tuned independently for the KITTI and Nordlandsbanen datasets (see Table IV). These values were found by manually tuning individual parameters until satisfactory results were obtained, while limiting some parameters, such as the radius of Ω_i, to values that realistically represent information that is available within a couple of frames in a given dataset. Choosing a prior method for parameter selection in different scenarios, such as finding keypoint scales, remains a task for future work.

For the Gestalt evaluation that we perform on the Nordlandsbanen datasets we use the same parameters as for NBLD.

IV. Results

The results for KITTI 00, 05 and 06 can be seen in figure 5. We can conclude that with the given parameters, NBLD performs similarly to BRISK.

Since NBLD operates on 3D structure and ignores visual descriptors, it is interesting to see how it performs on data that has been recorded across seasons, since such data is subject to changes in the visual information. As can be seen in Figure 6, NBLD performs better than both Gestalt

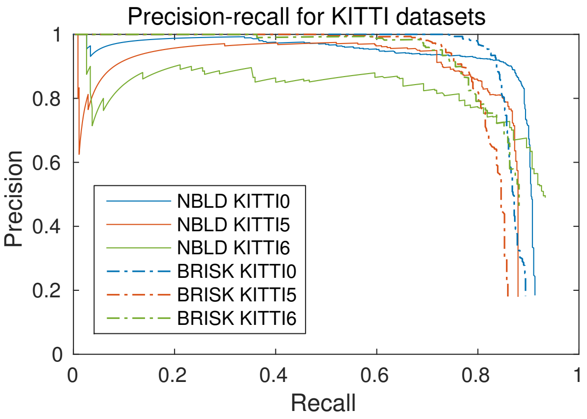


Fig. 5: Precision-recall curves for the three KITTI datasets that contain a discernible loop closure. The used NBLD radius is 9 meters.

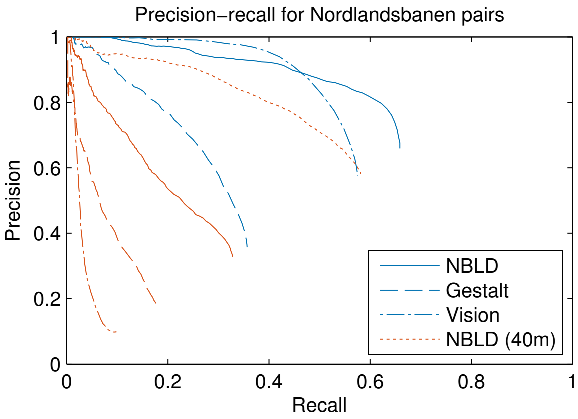


Fig. 6: Precision-recall curves for summer-fall (blue) and spring-winter (red) of the Nordlandsbanen datasets (first 10km). Gestalt and NBLD use a Ω_i radius of 16 meters. Additionally, NBLD has been evaluated with a radius of 40m on spring-winter.

and BRISK across the challenging spring-winter pair from the Nordlandsbanen datasets. Still, the performance for this dataset pair is several times worse than the performance across the easier summer-fall combination or the performance with the KITTI datasets. We presume that this is due to the fact that our structural descriptors rely on what landmarks are detected by the structure from motion algorithm. Indeed, the results for other season pairs are even worse than for spring-winter, for all evaluated descriptors. Which landmarks are seen still varies with lighting. For instance, textured areas that receive more light can potentially give rise to more landmarks than if they are dark. As can be seen in Figure 4, the winter frames have illumination that is darker and completely different than in the spring frames. Inspecting the Nordlandsbanen point clouds indeed reveals that the spring dataset contains significantly more points than the winter dataset. Additionally, the tracking is more difficult as a window wiper regularly passes in front of the camera. In spite of these challenging conditions, the difference in structure still gives rise to less confusion than the difference in visual information in the images. Moreover, the performance of NBLD can be improved by increasing the radius of the descriptor domain Ω_i. Figure 7 shows how the MCC in the spring-winter pair increases as that radius is increased. Around a radius of $40m$ a plateau is reached. The PR curve at this radius is included in Figure 6. With the Nordlandsbanen dataset, Ω_i becomes emptier as the radius increases, since a bigger portion of the descriptor covers space where no landmarks are present. This explains why performance stops increasing at a certain radius. Furthermore, there is a caveat with increasing the radius: At a certain point, the descriptor describes the shape of the trajectory instead of the landmark structure. It would be worth investigating in future work at which radius this point is reached.

The timing of the spring-winter Nordlandsbanen evaluation ($\sim 300'000$ landmarks combined) is shown in Table V. It is not possible to give an exact estimate of the runtime in an online setting based on offline timing, especially since we employ CPU data parallelism in many places. However, the results encourage us to believe that our implementation could easily be adopted in a real-time online localization application, especially since descriptor evaluation, PCA and projection for D could be pre-processed.

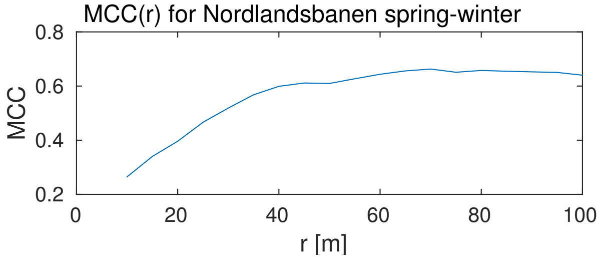


Fig. 7: Increasing the radius of the descriptor domain Ω_i increases the place recognition performance up to a plateau.

Descriptor evaluation	12s
Principal component analysis	63s
Descriptor projection	34s
Landmark matching	106s
Vote aggregation	2s
Total	217s

TABLE V: Timing of the spring-winter Nordlandsbanen evaluation ($\sim 300'000$ landmarks combined). Each Nordlandsbanen trajectory has been recorded over 480 seconds.

V. Conclusion

In this work we have shown that sparse 3D points estimated by visual SLAM can be aggregated into a structural descriptor, which is useful for large-scale place recognition. The proposed descriptor, dubbed *Neighbour-binary landmark density descriptor* (NBLD) can perform better than visual descriptors if the scene undergoes appearance changes, such as in the Nordlandsbanen datasets recorded across different seasons. However, since the points used by NBLD still originate from visual cues, the appearance changes cannot be arbitrarily strong. While for instance SeqSLAM [20] presents better results for stronger appearance changes, it is very sensitive to viewpoint changes [31]. Since NBLD does not depend directly on the image sequence, it would hence be very interesting to analyze the viewpoint dependence of NBLD-based place recognition.

Other than providing robustness to lighting changes, we believe that the presented work lays out how place recognition can be achieved in applications where availability of visual data is limited, such as in future DVS SLAM pipelines [33].

In future work we propose to reduce the amount of free parameters, for instance by deriving a keypoint scale from structural properties. Another very interesting subject is direct fusion of point cloud descriptors with visual descriptors in order to obtain a descriptor that incorporates both visual and structural information, potentially combining the advantages of both approaches.

VI. Acknowledgments

The research leading to these results has received funding from Google's project Tango and the Swiss National Science Foundation through the National Center of Competence in Research Robotics.

References

[1] M. Li, B. H. Kim, and A. I. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," in *IEEE Int. Conf. on Robotics and Automation*, 2013.

[2] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Towards consistent vision-aided inertial navigation," in *Workshop on the Algorithmic Foundations of Robotics*, 2013.

[3] E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis, "C-KLAM: Constrained Keyframe-Based Localization and Mapping," in *IEEE Int. Conf. on Robotics and Automation*, 2014.

[4] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization," *The Int. Journal of Robotics Research*, vol. 34, no. 3, 2015.

[5] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Int. Conf. on Computer Vision*, 2003.

[6] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[7] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The Int. Journal of Robotics Research*, vol. 27, no. 6, 2008.

[8] D. Galvez-Lopez and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, 2012.

[9] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, 2012.

[10] E. S. Stumm, C. Mei, and S. Lacroix, "Building location models for visual place recognition," *The Int. Journal of Robotics Research*, 2015.

[11] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conf. on Computer Vision*, 2008.

[12] H. Stewénius, S. Gunderson, and J. Pilet, "Size Matters: Exhaustive Geometric Verification for Image Retrieval," in *European Conf. on Computer Vision*, 2012.

[13] T. Sattler, B. Leibe, and L. Kobbelt, "Fast Image-Based Localization using Direct 2D-to-3D Matching," in *Int. Conf. on Computer Vision*, 2011.

[14] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart, "Placeless place-recognition," in *Int. Conf. on 3D Vision*, 2014.

[15] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," 2015.

[16] E. Johns and G.-Z. Yang, "Generative methods for long-term place recognition in dynamic scenes," *Int. Journal of Computer Vision*, vol. 106, no. 3, 2014.

[17] R. Paul and P. Newman, "FAB-MAP 3D: Topological mapping with spatial and visual appearance," in *IEEE Int. Conf. on Robotics and Automation*, 2010.

[18] E. Stumm, C. Mei, S. Lacroix, and M. Chli, "Location graphs for visual place recognition," in *IEEE Int. Conf. on Robotics and Automation*, 2015.

[19] N. Sünderhauf and P. Protzel, "Brief-gist-closing the loop by simple means," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011.

[20] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE Int. Conf. on Robotics and Automation*, 2012.

[21] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3d lidar datasets," in *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2013.

[22] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008.

[23] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conf. on Computer Vision*, 2010.

[24] R. Hänsch, T. Webera, and O. Hellwicha, "Comparison of 3D interest point detectors and descriptors for point cloud fusion," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 3, 2014.

[25] S. Leutenegger, M. Chli, and R. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Int. Conf. on Computer Vision*, 2011.

[26] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[27] 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*, 2012.

[28] M. Bosse and R. Zlot, "Keypoint design and evaluation for place recognition in 2D lidar maps," *Robotics and Autonomous Systems*, vol. 57, no. 12, 2009.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.

[30] T. Hellum, M. Steinholt, S. Skrede, *et al.*, "Nordlandsbanen: minute by minute, season by season," 2013.

[31] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging SeqSLAM on a 3000 km journey across all four seasons," in *IEEE Int. Conf. on Robotics and Automation*, 2013.

[32] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[33] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt, "Event-based 3D SLAM with a depth-augmented dynamic vision sensor," in *IEEE Int. Conf. on Robotics and Automation*, 2014.