Place Recognition in 3D Scans Using a Combination of Bag of Words and Point Feature based Relative Pose Estimation

Bastian Steder Michael Ruhnke Slawomir Grzonka Wolfram Burgard

***Abstract*— Place recognition, i.e., the ability to recognize previously seen parts of the environment, is one of the fundamental tasks in mobile robotics. The wide range of applications of place recognition includes localization (determine the initial pose), SLAM (detect loop closures), and change detection in dynamic environments. In the past, only relatively little work has been carried out to attack this problem using 3D range data and the majority of approaches focuses on detecting similar structures without estimating relative poses. In this paper, we present an algorithm based on 3D range data that is able to reliably detect previously seen parts of the environment and at the same time calculates an accurate transformation between the corresponding scan-pairs. Our system uses the estimated transformation to evaluate a candidate and in this way to more robustly reject false positives for place recognition. We present an extensive set of experiments using publicly available datasets in which we compare our system to other state-of-the-art approaches.**

***Index Terms*— Place recognition, SLAM, loop closing, point clouds, range images, range sensing**

I. Introduction

Place recognition, meaning the detection that a robot revisited an already known area, is a crucial part in key navigation tasks including localization and SLAM. The majority of state-of-the-art place recognition techniques have been developed for vision- or two dimensional range data. Relatively few approaches work on three-dimensional laser range scans and can efficiently calculate the similarity or the relative transformation between two scans.

In this paper we present a place recognition system operating on 3D range data. Our approach transforms a given 3D range scan into a range image and uses a combination of a bag-of-words approach and a point-feature-based estimation of relative poses that are then individually scored. Figure 1 shows an example application. It visualizes how the calculated relative transformations between scans can be used as edges (loop closures) in a pose graph. This enables us to apply our approach as a front-end for a graph-based SLAM system.

This paper builds on the results of our earlier work in the area of place recognition [16]. This approach had high recognition rates, but had shortcomings regarding the runtime and was not fully invariant to the orientation of the individual scans. Our algorithm described in this paper uses a novel feature type, an improved sensor model, includes a self-similarity analysis, and employs a bag-of-words approach as a preprocessing step to achieve a higher performance.

All authors are members of the University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany. {steder,ruhnke,grzonka,burgard}@informatik.uni-freiburg.de

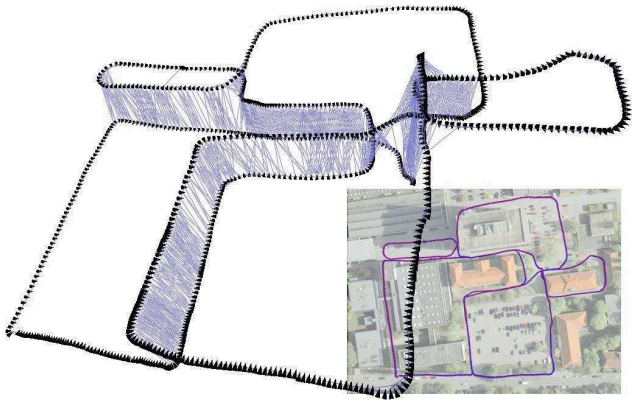


Fig. 1. Results from our place recognition system on the Hanover2 dataset. The image shows the graph of the trajectory (black nodes) and the found loop closures between the scans (blue/gray lines). The z-axis of the trajectory represents the scan index to make the loop closures more easily visible. The image in the bottom right shows an aerial image from Google Earth with the overlaid trajectory of the dataset. The experimental section provides further details about the dataset and our results.

We tested our approach on different kinds of platforms: ground robots and flying vehicles. For the ground robots, we used publicly available datasets to allow comparison with previous methods. For flying vehicles we acquired a new set of 3D range scans. For the sake of repeatability, we will make this data publicly available.

II. Related Work

In the past, the problem of place recognition has been addressed by several researchers and approaches for different types of sensors have been developed. Cameras are often the first choice. Compared to 3D data, vision features are typically very descriptive and unique. However, spacial verification is naturally easier in 3D data. One very successful approach using vision is the Feature Appearance Based MAPping algorithm (FABMAP) proposed by Cummins and Newman [7]. This algorithm uses a Bag-of-Words (BoW) approach based on SURFs [5] extracted from omni-directional camera images and was shown to work reliably even on extremely large-scale datasets. We would like to refer the reader to this paper for a detailed discussion on both vision-based place recognition and BoW approaches.

An approach that is similar to ours regarding the utilization of point features to create candidate transformations is described in the PhD thesis of Huber [11]. His approach extracts Spin Images [12] from 3D scans and uses them to match each scan against a database. Huber reported 1.5 s as the time requirement to match one scan against another. Even considering the advances in computer hardware since 2002,

Fig. 2. Example range image from the FreiburgCampus360_3D dataset. The image pixel positions represent the spherical coordinates of the points. The gray values represent the measured ranges. Blue points are maximum range readings and green points are unknown space.

our approach is substantially faster.

Li and Olson [14] create visual images from LIDAR data, which enables them to use feature extraction methods from the vision sector to create a more universally usable point feature extractor. This feature extraction method is usable in 2D and 3D, although a 2D projection of the points is performed in the 3D case. Therefore relative poses computed from feature correspondences will also just be 2D.

Several approaches have been designed especially for 2D range data. For, example, Bosse and Zlot [6] presented a loop closure solution that builds local maps from consecutive 2D scans for which they compute histogram-based features. The correlation between these features is than used to match the local maps against each other. Tipaldi *et al.* [18] perform place recognition on 2D range scans using the so-called FLIRT-features (Fast Laser Interest Region Transform). The features are used to find correspondences between points in the scans and transformations are extracted using RANSAC. Granström *et al.* [8] proposed a machine learning approach to detect loops in 2D range scans. They extract a combination of rotation-invariant features from the scans and use a binary classifier based on boosting to recognize previously visited locations.

Recently, Granström *et al.* [9] extended their system to 3D range scans. Their system only detects the existence of a loop closure and does not determine the relative transformation between scans. Magnusson *et al.* [15] also proposed a system for place recognition based on 3D data. They utilize the Normal Distribution Transform as features to match scans. These features are global appearance descriptors, which describe the whole 3D scan instead of just small areas as it is the case for our features. While being very fast, their system does also not estimate relative poses. In Section IV, we will compare our algorithm to these two methods. The results indicate that our approach yields substantially higher recall rates.

III. Technical Section

In our former work on place recognition [16] we used point feature correspondences to find candidate transformations between scans and calculated scores for those transformations. The main problem was that the runtime-requirements for this approach were relatively high and that it was not completely rotationally invariant. The algorithm presented here is similar regarding the basic functionalities. However, we introduced several improvements to make the algorithm more efficient and also more robust. In the remainder of this section we will describe the different components of our new algorithm in detail.

A. Overview

Given a database of 3D scans and a scan as input query, our algorithm returns a set of scans which are potential matches with the input. Additionally, it calculates for every returned scan a transformation and a score reflecting how certain the system is that the scans actually match.

More formally, let D denote the database of 3D range measurements and z^* a query scan. The goal of our approach is to calculate a set of candidate pairs, $C(z^*) = (\langle z_1, T_1, s_1\rangle, \ldots, \langle z_n, T_n, s_n\rangle)$. Here, $z_i \in D, i \in \{1, \ldots, n\}$ are the potential measurement candidates from the database which are similar to the current query z^*. Whereas T_i denotes the estimated transformation from z^* to z_i, s_i is a score reflecting the confidence about the match. Our algorithm for calculating $C(z^*)$ mainly consists of the following four steps.

1) Given a database of 3D range measurements D' (training set), calculate a set of features from the 3D scans and build a dictionary for a bag-of-words (BoW) approach.
2) Use the BoW approach to get an initial similarity measure for all scans in the database D with respect to the query scan z^*. Using this measure, order the database scans according to their similarity. Let the resulting ordered set be $\hat{D}(z^*) = \langle \hat{z}_1, \ldots, \hat{z}_{|D|}\rangle$.
3) For each pair $\langle z^*, \hat{z}_k\rangle, \hat{z}_k \in \hat{D}(z^*)$, starting with $k = 1$, calculate a set of possible transformations between z^* and $\hat{z}_k$ by matching point features of the corresponding scans. Note that this set of features is not the same as the one used for the BoW approach, since the parameters for the feature extraction differ.
4) Score each of the possible transformations and get the transformation T_k with the highest score s_k. If this score is above an acceptance threshold then $\langle \hat{z}_k, T_k, s_k\rangle$ is a candidate for a recognized place, i.e., it is added to $C(z^*)$.

The last two steps are repeated until a timeout occurs or $k = |D|$. Note that if there are no time constraints, the first two steps can be skipped so that all scans in D are checked.

Although we work with a database of 3D range scans, we do not use this data directly. We rather represent each three-dimensional range scan by its dual, namely a range image (see Figure 2). If the 3D scan is captured from one point in space, i.e., the sensor does not move while the 3D points are generated, the range image contains the same information as the scan. The advantage of the range image is that it allows us to model unknown areas as well as maximum range readings more efficiently.

We will now describe the individual components of our approach in more detail.

B. Feature Extraction

Our approach applies the so-called NARFs (Normal-Aligned Radial Features) [17] recently developed for robust object recognition based on 3D scans. These point features are used to build a dictionary for the bag of words approach and also to find corresponding regions between two 3D

measurements. Compared to the approach presented in an earlier work [16], NARFs provide more robust key points and the feature descriptor is less prone to noise in the data.

There are three parameters needed for the extraction of NARFs. First, the size of the feature descriptor, second the maximum number of calculated features, and finally the support size, which is the size of the area around the feature point that is used to calculate the descriptor. We chose 36 as the descriptor size. For the BoW approach a high number of features describing small parts of the environment is most useful. Therefore we extract 2000 features with a support size of $1/10$ of the average range in database D. However, when matching a new query z^* against D, a smaller number of more distinctive features is needed. Here, we extract 200 features with a support size of $1/5$ of the average range in D. Intuitively, a small support size makes the features susceptible to noise and less distinctive, whereas a large support size makes them more expensive to compute and less robust to partial occlusion and missing data. However, we found the values above to provide reasonable trade-offs between those properties.

The descriptors of the features can be compared using standard norms like the Manhattan distance. The resulting measure (the *descriptor distance*) describes the similarity between the described regions. Here, a high value reflects a low similarity. Furthermore, NARFs can either be used in a rotationally invariant version or without invariance regarding the rotation around the normal. For example, in the rotationally variant case the features distinguish between a top right corner and a top left corner of a square, whereas they do not in the rotationally invariant case. This is a useful distinction, since wheeled robots capturing 3D scans often move with very little change in their roll and pitch angle. Accordingly, they do not need the rotational invariance around the normal vector for the features. The same is the case if the robot is equipped with an IMU. This can reduce the computational complexity of the problem since the feature matching with one degree of freedom less is more robust. A comparison between the two modes can be found in Section IV-B.

C. Bag of Words

We use a BoW approach as a fast initial method to pre-order the scans according to their similarity to the given query scan z^*. BoW approaches are based on the idea that similar structures in an environment will create similar distributions of features. The goal is to obtain a general representation for those feature distributions. We want to encode each scan in terms of a small set of words (the *dictionary*). To learn this set, we use a training database D' of 3D scans and calculate 2000 NARFs for each scan. For a database of size n, this leads to $n \cdot 2000$ feature descriptors (each of size 36). We then apply k-means clustering to obtain a total of 200 clusters. Our dictionary is now made of 200 words, each being the averaged descriptor of its cluster. We found that this size provides a reasonable trade-off between being able to generalize and being descriptive enough. Given this dictionary, we can now express each scan $z_i \in D$ in terms of the words of the dictionary by selecting the closest word for every feature descriptor (regarding the Euclidean distance). For each z_i, we obtain a histogram H_i having 200 bins. The number in each bin reflects how often the corresponding word is present in z_i. Given the histogram of the query scan H^* (obtained in the same way), we calculate $\|H^* - H_i\|_2$ as the distance between the histograms. This distance is then used as an initial similarity measure to create the ordered set $\hat{D}(z^*)$, as described in Section III-A

In the next step we calculate a set of candidate transformations between the scan pairs.

D. Determining Candidate Transformations

Each NARF encodes a full 3D transformation. Therefore, the knowledge about a single feature correspondence between two scans enables us to retrieve all six degrees of freedom of the relative transformation between them (i.e., by calculating the difference between the two poses). To obtain the candidate transformations for each scan pair, we order the feature pairs according to increasing descriptor distance (see Section III-B) and evaluate the transformations in this order. In other words, we calculate a score for each of these transformations (see Section III-E). In our experiments we stop after a maximum number of 2000 evaluated transformations when using the rotationally variant version of the NARFs. In the rotationally invariant case however, we evaluate up to 5000 transformations due to the bigger search space introduced by the additional degree of freedom.

E. Scoring of Candidate Transformations

The result of the feature matching is a list of relative poses $\hat{T}_k = \{\hat{T}_{k_1}, \ldots, \hat{T}_{k_n}\}$ for the candidate pair $\langle z^*, \hat{z}_k \rangle, \hat{z}_k \in \hat{D}(z^*)$. Our goal now is to evaluate those candidate transformations and calculate a score (likelihood) for each $\hat{T} \in \hat{T}_k$ reflecting the confidence of the transformation given a model of our sensor. Recall that we use 3D range data, i.e., each measurement z is a set of 3D points. This enables us to evaluate the candidate transformation $\hat{T}$ on a point-by-point basis (i.e., we assume the points are mutually independent).

Let P be a set of *validation points* from the query scan z^*. This set P could contain all points from z^* but we will only use a representative subset of z^* as described at the end of this section. Given a candidate transformation $\hat{T} \in \hat{T}_k$, we first transform each $p \in P$ in the reference frame of z^* into a point p' in the reference frame of z_k. Since we represent our scans as range images, we can calculate the pixel position $p'(x, y)$ in the range image of $\hat{z}_k$ in which the point p' would fall into as well as the range value r' the point should have. Let $p_k(x, y) \in \hat{z}_k$ be the point that is already at this pixel position (in the range image of $\hat{z}_k$) having the range value $r_k(x, y)$. For each $p \in P$ we will now calculate a score $s_{\hat{T}}(p) \in [0, 1]$ and a weight $w_{\hat{T}}(p) > 0$ reflecting how good the prediction r' is explained by the observation $r_k(x, y)$. The point scores will then be used to calculate the overall likelihood $s(\hat{T})$ for the transformation $\hat{T}$ by:

$$s(\hat{T}) = \frac{\sum_{\forall p \in P} w_{\hat{T}}(p) \cdot s_{\hat{T}}(p)}{\sum_{\forall p \in P} w_{\hat{T}}(p)}. \quad (1)$$

Let $\Delta r = r_k(x,y) - r'$ be the difference between the observed and the predicted range. To evaluate Δr, we have to consider the model of the sensor. In the case of a laser scanner, a pulse of light moves from the sensor's origin along a line to the measured point (each range image pixel represents one such beam). There are several different cases needed to be considered regarding the interpretation of Δr:

1) The observation is within a confidence interval $\Delta r_{\max} > 0$ of the prediction, i.e., $|\Delta r| < \Delta r_{\max}$. In other words, what we expected to see mostly fits with what we measured. In this case, we calculate the score as $s_{\hat{T}}(p) = 1 - \frac{|\Delta r|}{\Delta r_{\max}}$ and weight it by $w_{\hat{T}}(p) = 1$, which represents a triangular distribution. While a Gaussian would be a more realistic representation, we chose a triangular distribution, since it is less expensive to compute.
   All the other cases will receive a score $s_{\hat{T}}(p) = 0$. Thus, the associated weight $w_{\hat{T}}(p)$ reflects the confidence about how *wrong* the transformation $\hat{T}$ is.
2) The observed range is larger than the predicted one, i.e., $\Delta r > \Delta r_{\max}$. This means that we actually observed something behind p' and basically looked through it. This could be the evidence for a dynamic or partially transparent obstacle, but in general it is a strong indicator for a wrong transformation. We therefore penalize the overall likelihood by a high weight $w_T(p) = w_{\text{seeThrough}} \geq 1$.
3) The observed range is smaller than the predicted range, i.e., $\Delta r < -\Delta r_{\max}$. In this case, there are two more situations to distinguish:
   a) $\hat{T}^{-1} \cdot p_k(x,y)$ exists in z^*. This means that we could not see p' in $\hat{z}_k$ because of an already known obstacle. In this case we give a low weight $w_{\hat{T}}(p) = w_{\text{knownObstacle}} \leq 1$ in order to enable us to receive relatively high scores even if the overlap between scans is low.
   b) $\hat{T}^{-1} \cdot p_k(x,y)$ does not exist in z^*. This could be evidence for a formerly unseen or dynamic obstacle, but in general it is a strong indicator for a wrong transformation. Similar to case 2, we penalize this by a high weight $w_{\hat{T}}(p) = w_{\text{unknownObstacle}} \geq 1$.
4) $p_k(x,y)$ is an unobserved point in the range image of $\hat{z}_k$. This means that p' could not be observed because it is outside of the scan. We treat this the same as 3a.
5) $r_k(x,y)$ is a far range reading (i.e., exceeding the max range of the sensor) in the range image of $\hat{z}_k$. There are two more situations to distinguish, for which we need to consider the original range r of p in z^*:
   a) The point should actually be closer to the sensor in $\hat{z}_k$, i.e., $r' \leq r$. In this case it is improbable that p' is out of range and therefore we treat this the same as case 2.
   b) The point moved further away from the sensor in $\hat{z}_k$, i.e., $r' > r$. In this case it is possible that p moved out of range and we give a medium high weight $w_{\hat{T}}(p) = w_{\text{farRange}} \geq 1$.

To avoid that slight errors in the estimate of a correct transformation lead to a very small score, e.g., if the point lies on an obstacle border and we hit the much further away neighbor instead, we actually consider not only $p'(x,y)$ as a correspondence for p', but also its neighbors in a small pixel radius $e \in \mathbb{N}$ (3 in our experiments) around it and select the point with the least negative influence on the complete score.

Until now we did not say, how the set of validation points P, from which we select p, is obtained. In principle it could contain all the points from z^*. However, this would lead to a high number of points to be tested and thus would be computationally expensive. We therefore use only a subset of z^*. A random subset of a fixed size could be used, but it is better to select points that have some significance in the scene, or two scans could get a high score, just because the floor or a big wall is well aligned. Furthermore, the points should be evenly distributed over the scan in 3D space to be invariant regarding the non-uniform resolution of 3D scans. To achieve this, we use the set of key-points $\hat{P}$ (i.e., the points where the NARF's are) that we calculated in the feature extraction process as a base to create the set of validation points. We add a random point from $\hat{P}$ to P and then iteratively add the point $\hat{p}_i \in \hat{P}$ that has the highest 3D distance to all points already in P, until a maximum size is reached (200 points in our current implementation). This has the interesting property that each ordered subset $\langle p_0, \cdots, p_j \rangle$ of the ordered set $P = \langle p_0, \cdots, p_{|P|} \rangle$ is a subsampled version of $\hat{P}$ with mostly equidistant points for every j. This also means that one can stop the calculation of $s(\hat{T})$ (see Eq. 1) before handling each point in P if the score is already to low after a certain minimum of handled points (30 points in our experiments), since this subset already represents the whole set quite well.

Since the score $s(\hat{T})$ for the transformation $\hat{T}$ is not necessarily the same as for $\hat{T}^{-1}$ (by switching the role of z^* with $\hat{z}_k$), we adapt the scoring to $s'(\hat{T}) = \min(s(\hat{T}), s(\hat{T}^{-1}))$ as the score for the pair $\langle z^*, \hat{z}_k \rangle$ with transformation $\hat{T}$.

F. Self-Similarity

There are scans that qualify only poorly for the pose estimation process because of a high self-similarity, e.g., corridors with very few distinctive structure. To prevent false positives (false transformations getting a high score) in those areas, we calculate a self-similarity score for every scan. We do this by matching the scan z against itself, using the procedure described above and consider only transformations that are not close to the identity matrix. We call the highest score in this set $self(z)$ and consider it as a measure for self similarity. We then adapt the scoring and obtain the final score for a transformation between z^* and $\hat{z}_k$ in the following way: $s^*(\hat{T}) = (1 - (self(z^*) + self(\hat{z}_k))/2) \cdot s'(\hat{T})$. Recall that we perform the steps described so far for each candidate transformation $\hat{T} \in \hat{T}_k$. If the best score out of all candidates is above a threshold, $\hat{z}_k$ represents a potential loop closure, i.e., $C(z^*) := C(z^*) \cup \langle \hat{z}_k, T_k, s_k \rangle$ with $T_k = \text{argmax}_{\hat{T} \in \hat{T}_k} s^*(\hat{T})$ and $s_k = s^*(T_k)$.

G. Implementation details

We perform some additional steps to improve the results. After an initial scoring of the candidate transformations for a scan pair $\langle z^*, \hat{z}_k \rangle$ we first remove transformations with a very low score. We then cluster the transformations and identify those describing very similar relative poses, keeping only the best ones in the candidate list. Next, we perform ICP to improve the transformation estimate, using only the set of validation points to speed up this step. Finally, we update the scores given the corrected transformations and return the transformation associated with the highest score as the result.

IV. Experiments

In this section, we present the real-world experiments carried out to evaluate our approach. We used four publicly available datasets of 3D scans, namely two outdoor datasets and two indoor dataset. In the following we will give an overview over these datasets and their specific challenges.

A. Datasets

The following datasets were used in our experiments:

- For the first indoor dataset we chose **AASS-loop**[1] [1]. This dataset was also used in the related work [8], [15], which makes a comparison easier. Its main challenge is that it contains some highly ambiguous areas in long corridors.
- For the second indoor dataset we captured 3D scans with a flying **Quadrotor** robot, equipped with a 2D laser scanner [10]. This dataset [4] is challenging because of a higher noise level and the existence of highly similar scans from different poses in a corridor environment.
- For the first outdoor dataset we chose **FreiburgCampus360_3D** [2]. We already used this dataset in our prior work on place recognition [16]. It contains high resolution, 360° scans and its main challenge is the large distance between consecutive scans, stressing the system's ability for translational invariance.
- For the second outdoor dataset we chose **Hanover2**[2] [3]. This dataset was also used in previous work [8], [15], [16], which makes a comparison easier. This is a challenging dataset, since it contains a high number of very sparse scans and the robot traverses different areas with very similar structure.

All datasets apart from the Quadrotor dataset were recorded with 2D laser scanners mounted on pan/tilt units. We acquired SLAM trajectories using the provided odometry and manually verified scan matching as edges in the graph SLAM system g^2o [13]. These trajectories were used to evaluate false/true positives and false/true negatives in our system. In the Quadrotor dataset the helicopter occasionally captured a 3D scan by flying downwards and upwards again while hovering around the same spot. Here, the trajectory was estimated using the quadrotor's navigation system as described in [10]. Please refer to Figure 3 for more information about the datasets.

In all experiments we used $w_{\text{seeThrough}} = 25$, $w_{\text{knownObstacle}} = 0.5$, $w_{\text{unknownObstacle}} = 15$, and $w_{\text{farRange}} = 5$ as the parameters for the scoring function (as defined in Section III-E).

B. Confusion Matrices

We calculated the confusion matrices for the datasets (see Figure 4(a)) by matching each scan with every scan in the database and returning the score of the best found transformation. The dark areas that are not close to the main diagonal mark loop closures. Here the system was able to match scans from different points in time where the robot visited a previously visited area (see also Figure 3).

To evaluate if a match is a false positive, we compared the ground truth transformation between the scans with our found transformation and check if it exceeds an error value. Please note that this is a harder condition than used in related work [15], [9], where no relative pose is estimated and only the distance between the scans is considered.

Figure 4(b) gives an overview over the number of true positives and false negatives and the resulting recall rate as a function of the distance between scans, using the minimum acceptance threshold for which no false positive was found.

Figure 4(c) plots the number of false positives as a function of the acceptance threshold. The recall rate for a manually set maximum distance between scans is also shown.

For AASS-loop we used $1.0\,\text{m}$ as the distance to consider two scans a match (this is the same as in previous work [15], [9]). The minimum acceptance threshold for which we received no false positive is 0.09. Above this value we have a recall rate of 0.938. The equivalent values for the Quadrotor dataset are $2.0\,\text{m}$ / 0.25 / 0.75, for FreiburgCampus360_3D $10.0\,\text{m}$ / 0.05 / 0.958, and for Hanover2 $3.0\,\text{m}$ / 0.19 / 0.925 respectively.

Our evaluations do not include the diagonal elements of the confusion matrices (where the scans are matched against themselves). Since Granström and Schön [9] used a machine learning algorithm based on boosting they had to split their dataset into learning and test sets for the cross validation and therefore did not evaluate the complete confusion matrix at once. They reported rates of 0.53 ± 0.14 (min 0, max 0.88) for the AASS-loop dataset and 0.63 ± 0.6 (min 0.28, max 0.76) for the Hanover2 dataset.

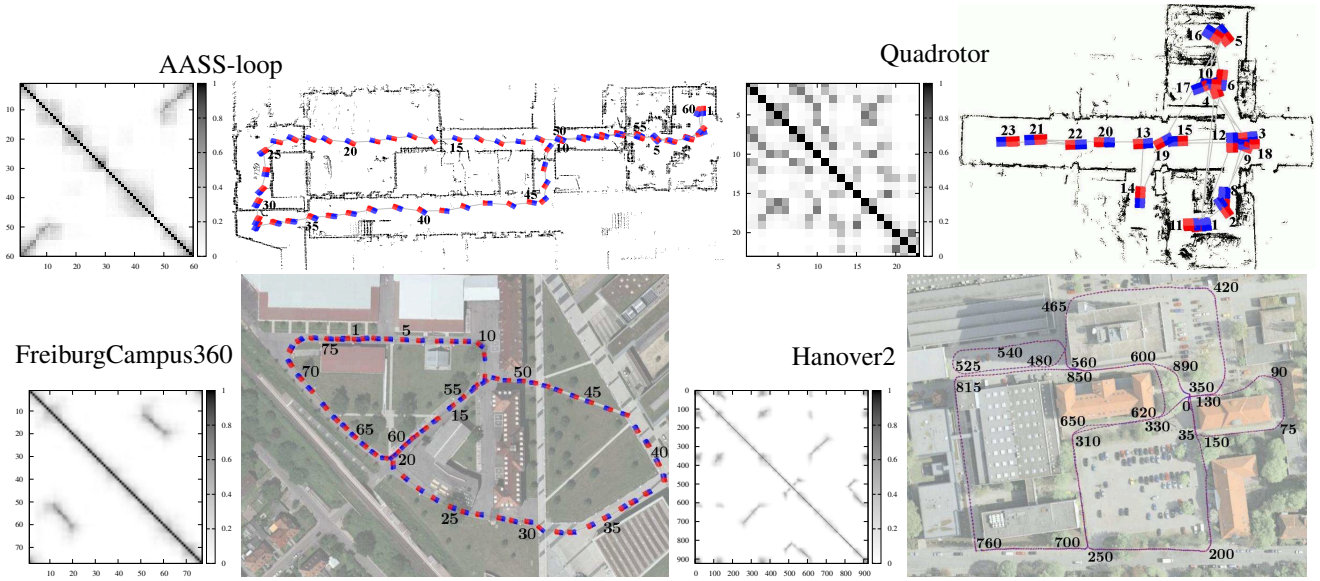
Magnusson *et al.* [15] evaluated their system in a SLAM scenario, where only scans that are at least 30 scans apart are evaluated. In this scenario they got 0.7 as the recall rate for AASS and 0.47 for Hanover2, respectively at 100% precision. With the same setting we got 1 (0.08 acceptance threshold) for AASS and 0.911 (0.19 acceptance threshold) for Hanover2.

C. Timings and Influence of the BoW approach

The values given so far are the results we receive, when we do not restrict the time requirements of our system.

For the AASS-loop dataset it takes us $881\,\text{ms}$ to extract interest points, features and validation points per scan and

[1]Courtesy of Martin Magnusson, AASS, Örebro University, Sweden
[2]Courtesy of Oliver Wulf, Leibniz University, Germany



	stop&go	#scans	#points	#farRanges	res	traj	dist	$\mu-$range	maxRange
AASS	yes	60	80873	n/a	0.7°	111.4 m	1.89 m	2.81 m	67.1 m
Quadrotor	no	23	171608	95515	1.0°	79.1 m	3.6 m	1.9 m	6.1 m
Freiburg	yes	77	155562	56451	0.45°	723 m	9.51 m	8.9 m	50 m
Hanover	no	923	12959	3622	1.3°	1247.8 m	1.35 m	7.18 m	29 m

Fig. 3. **Top:** SLAM trajectories and ground truth confusion matrices for the used datasets. For the indoor datasets the trajectory is plotted on a 2D projected laser map and for the outdoor datasets it is overlaid on a Google Earth aerial image. The gray values in the confusion matrices represent the amount of overlap between the scans given the true relative pose. **Bottom:** Overview over the properties of the used datasets: **stop&go**=scans captured in stop and go fashion, **#scans**=number of scans, **#points**=average number of points per scan, **#farRanges**=average number of far range readings per scan, **res**=usable angular resolution for range images, **traj**=trajectory length, **dist**=average distance between consecutive scans, $\mu-$**range**=average measured range value, **maxRange**=maximum range value

585 ms to match a scan against the database, meaning 10 ms for each scan pair. The equivalent values for the Quadrotor dataset are 305 ms, 102 ms, and 4 ms, for the FreiburgCampus360_3D dataset 1107 ms, 838 ms, and 11 ms, and for the Hanover2 dataset 316 ms, 4132 ms, and 4 ms respectively. All experiments were performed using an Intel I7 quad-core PC.

When using the BoW approach, there is an additional overhead for the creation of the histograms (including feature extraction), which is 894 ms for AASS-loop, 276 ms for Quadrotor, 730 ms for FreiburgCampus360_360, and 246 ms for Hanover2.

Using the BoW pre-ordering of the potential corresponding scans, we can define a timeout for the database query. Please refer to Figure 4(d) for an overview, how the recall rates (for the respective minimum acceptance threshold and maximum scan distance) evolve for increasing timeout values. It can be seen that the additional overhead for the histogram calculation is only justifiable for the biggest dataset, namely Hanover2. Here, a recall rate of close to 80% can already be reached after one second per database query. In the same plots there is also a comparison between the rotationally invariant and non-invariant version of the NARFs. It can be seen that the additional degree of freedom introduced by the rotational invariance increases the typical runtime to achieve a certain recall rate and that the maximum achievable recall rate is lowered. But overall, the recall rates are still above the values of the other state-of-the-art systems in the related work.

Please note that we used the Freiburg dataset to learn the dictionary for the BoW approach. Therefore this result (see Figure 4(d)) might be overconfident.

V. Conclusions

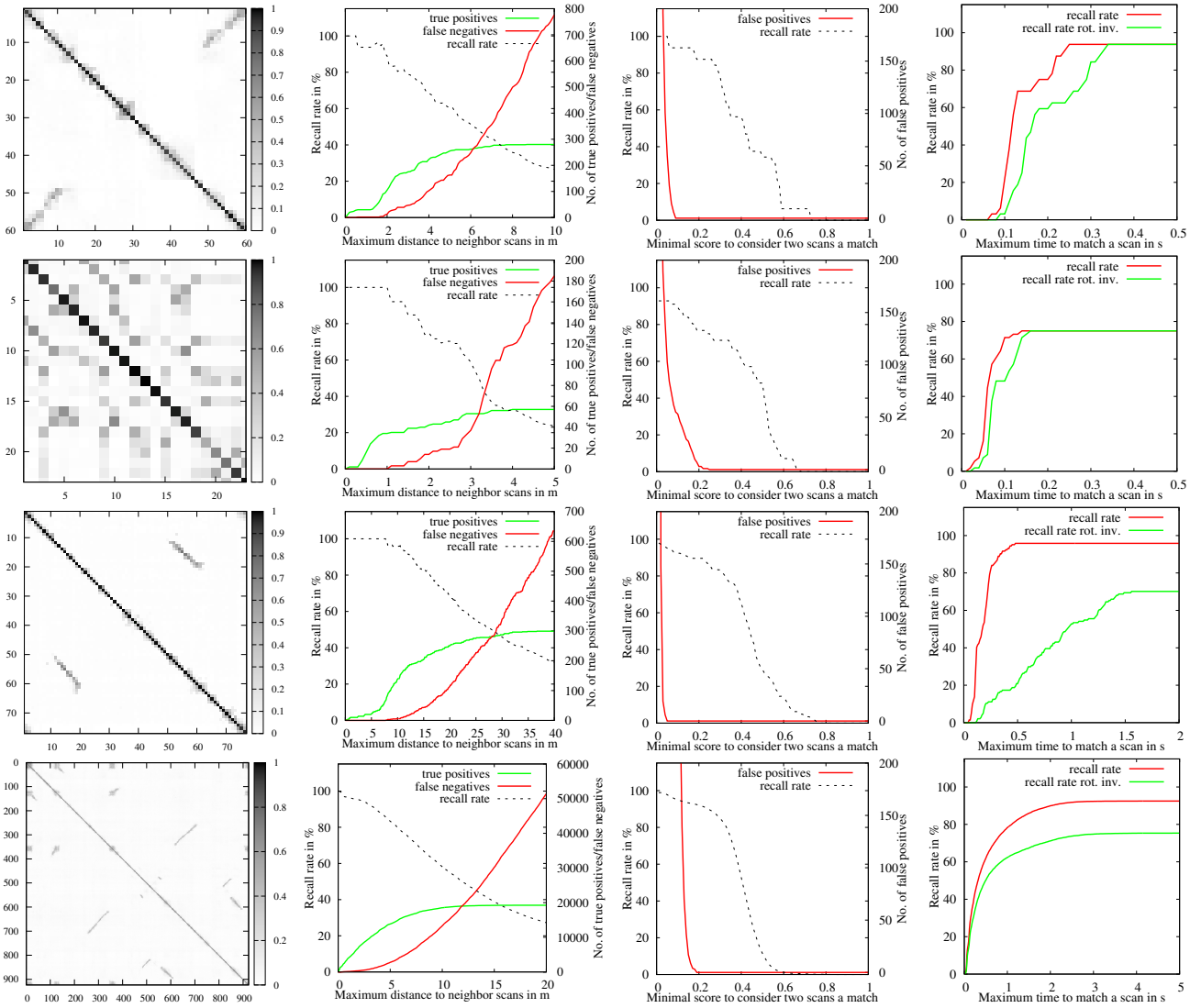
In this paper we presented a robust approach to 3D place recognition that simultaneously computes relative pose estimates between the involved 3D range scans. Our approach is computationally more efficient compared to our previous work while still receiving recall rates that compare favorably to alternative approaches. Additionally, the application of the recently developed normal-aligned radial features enabled us to overcome the limitations regarding rotational invariance of our former approach. We also presented a novel sensor model. A carefully carried out evaluation revealed that our new approach yields a more robust scoring of relative pose estimates.

Acknowledgments

This work has partly been supported by the European Commission under contract number FP7-231888-EUROPA. We wish to thank Giorgio Grisetti for his valuable input and Rainer Kümmerle for his support regarding obtaining the SLAM trajectories and his helpful advice.

References

[1] Aass-loop dataset. http://kos.informatik.uni-osnabrueck.de/3Dscans.
[2] Dataset of 360° 3D scans of the Faculty of Engineering, University of Freiburg, Germany. http://ais.informatik.uni-freiburg.de/projects/datasets/fr360.



(a) **(b)** **(c)** **(d)**

Fig. 4. **First row**: Results AASS-loop **Second row** Results Quadrotor **Third row**: Results FreiburgCampus360_3D **Fourth row**: Results Hanover2 **(a)**: Confusion matrices created by our system. **(b)**: The number of true positives, false negatives, and the resulting recall rate for different maximum distances between scans to consider them overlapping. Respectively for the minimum acceptance threshold that did not return any false positives. **(c)**: Number of false positives and the recall rate for different minimum scores. The recall rate is determined regarding a maximum distance of 1.0 m / 2.0 m / 10.0 m / 3.0 m (from top to bottom) between the scans. **(d)**: The recall rate dependent on the maximum time the system has to match a scan against the database, using the BoW approach. The two graphs represent the recall rate with and without the rotational invariance in the NARFs.

[3] Hanover2 dataset. http://kos.informatik.uni-osnabrueck.de/3Dscans.

[4] Quadrotor dataset. http://ais.informatik.uni-freiburg.de/projects/datasets/quadrotor079.

[5] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*, 2006.

[6] M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. *International Journal of Robotics Research*, 27(6):667–691, 2008.

[7] M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *Int. Journal of Robotics Research*, Nov 2010.

[8] K. Granström, J. Callmer, F. Ramos, and J. Nieto. Learning to detect loop closure from range data. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[9] K. Granström and T. Schön. Learning to close the loop from 3D point clouds. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[10] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[11] D. Huber. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2002.

[12] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999.

[13] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[14] Y. Li and E. Olson. A general purpose feature extractor for light detection and ranging data. *Sensors*, 10(11):10356–10375, 2010.

[15] M. Magnusson, H. Andreasson, A. Nüchter, and A. J. Lilienthal. Automatic appearance-based loop detection from 3D laser data using the normal distributions transform. *Journal of Field Robotics*, 26(11–12):892–914, November 2009.

[16] B. Steder, G. Grisetti, and W. Burgard. Robust place recognition for 3D range data based on point features. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.

[17] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard. Point feature extraction on 3D range scans taking into account object boundaries. In *Proc. of the IEEE Int. Conf. on Rob. & Automation (ICRA)*, 2011.

[18] G. D. Tipaldi, M Braun, and K. O. Arras. Flirt: Interest regions for 2D range data with applications to robot navigation. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2010.