Place Recognition Using Keypoint Voting in Large 3D Lidar Datasets

Conference Paper *in* Proceedings - IEEE International Conference on Robotics and Automation · May 2013

DOI: 10.1109/ICRA.2013.6630945

2 authors:

Michael Bosse
ETH Zurich
70 PUBLICATIONS **6,046** CITATIONS
SEE PROFILE

Robert Zlot
Uber
38 PUBLICATIONS **3,788** CITATIONS
SEE PROFILE

Place Recognition using Keypoint Voting in Large 3D Lidar Datasets

Michael Bosse and Robert Zlot

***Abstract*— In developing autonomous solutions for mapping and localization, one problem that often needs to be dealt with is determining when an area is revisited despite having poor or no prior information on the relative alignment error. There are well-formulated approaches for recognizing such matches using the rich information in camera data; however, it is a much more challenging problem using lidar sensors alone. Most existing approaches employ a pairwise place comparison of place descriptors and thus finding matches requires linear time per place. We instead propose the use of a keypoint voting approach to achieve sub-linear matching times. A constant number of nearest neighbor votes per keypoint are queried from a database of local descriptors and aggregated to determine likely place matches. It becomes critical to analyze the distributions of vote scores such that a suitable threshold for matching scores can be determined a priori, so that the system is not overwhelmed by false positives nor starved for true matches. We have empirically determined that the vote scores follow a log-normal distribution, and we are able to fit a parametric model of its hyper-parameters based on the number of neighbors, the number of keypoints in a place, and the total number of keypoints in the database. We demonstrate the performance of our system in a variety of large scale 3D lidar datasets using data collected from a continually scanning handheld lidar sensor, and also on two publicly available lidar datasets.**

I. INTRODUCTION

Place recognition is an important problem in the areas of mapping, localization, and spatial reasoning. It requires the determination of whether a given region has been previously observed, and if so, how the two or more observations of that region are related geometrically. Reliable place recognition solutions have many obvious applications for mobile robotics, in terms of both spatial representation and state estimation. Place recognition also has significant potential to increase the workflow efficiency in the surveying and lidar scanning industry by automating the assimilation of very large datasets.

Generally speaking, an effective place recognition solution provides three fundamental capabilities. The first is global localization: the ability to automatically detect the sensor location within a map in the absence of specialized infrastructure including GPS. For example, first responders at a disaster site might wear or carry a sensing platform that could be used to quickly identify where they are located provided a prior map is available, even in the presence of deteriorated infrastructure. The second capability is the facility to close arbitrarily large loops in Simultaneous Localization and Mapping (SLAM) systems. Registration algorithms typically require a coarse initial alignment in order to converge to a valid solution. This alignment is often performed manually. An autonomous procedure capable of generating a reliable coarse alignment regardless of the initial registration error would enable an increase in the scale of datasets that can be processed in a cost-effective manner. The third capability that can be achieved through place recognition is the ability to merge multiple datasets that contain some degree of overlap. Reliable and efficient dataset merging can increase acquisition efficiency (multiple robots or surveyors can operate simultaneously), flexibility (data can be collected at different times) and robustness (acquisition errors or anomalies can be detected and potentially corrected if at least one dataset contains reliable data in the problem area).

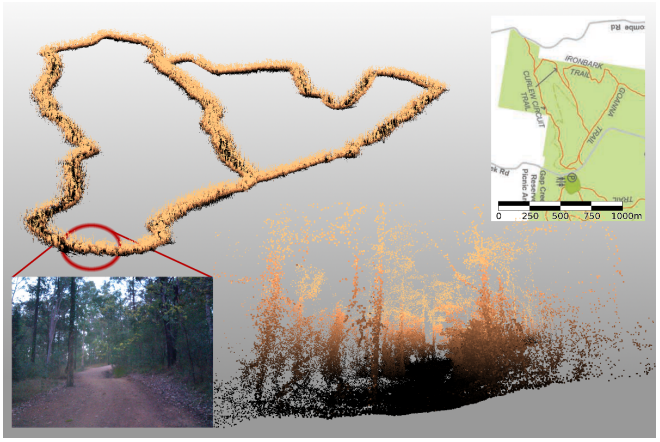

Fig. 1: MT COOT-THA Forest Environment. The dataset was collected from a handheld continuously scanning lidar by two operators walking a total of 2.64 km on different paths through the forest. The points are colored by their height relative to ground, as there are about 60 m of elevation changes. The zoomed in detail is the same place as depicted in the photograph.

In order to identify re-observed places, most existing approaches define a function that returns a similarity or distance score between a given pair of places, and then perform a pairwise comparison between a query place and all other existing places based on this function. Though the comparisons occur between descriptors at the place-level, these place descriptors are often generated by aggregating keypoint descriptors based on measurements in a smaller-scale local region. One popular approach is the bag-of-words (BoW) technique [4], [16], [7], in which the place descriptors are vectors containing binary values or weighted histograms accumulating occurrences of keypoint descriptors quantized to words in a learned vocabulary. Similarly, the Normal Distribution Transform (NDT) method [10] aggregates the occurrence of local descriptors into a histogram vector for

The authors are with the Autonomous Systems Laboratory, CSIRO ICT Centre, Brisbane, Australia
{mike.bosse,robert.zlot}@csiro.au

the place; however, in this case the local descriptors are shape subclasses arising from 3D points within a Cartesian voxel grid spatial decomposition. The general approach is essentially the same between these two methods, with the fundamental difference being that "words" used in the NDT approach have a geometric interpretation, in contrast to data-defined cluster centers learned from a training set as used by BoW. Granström *et al.* [8] do not define local keypoint descriptors, instead forming place descriptor vectors from place-scale geometric and statistical properties.

In our previously published work, we introduced a keypoint voting-based approach [1] that does not require place descriptors, but instead performs all comparisons at the keypoint descriptor level using nearest neighbors searches. Matches are accumulated into vote scores (rather than descriptors) for their associated places, and therefore queries can be run in sublinear time with respect to the number of places. We demonstrated the effectiveness of this approach in 2D [1], [20], where even low keypoint descriptor match rates still resulted in high place recognition rates in aggregate, even at dataset scales of hundreds of kilometers (while running significantly faster than real-time). We additionally demonstrated the keypoint voting algorithm in 3D [2].

This paper describes our continuing development of a keypoint voting approach and demonstrates results on a variety of larger-scale and otherwise highly challenging environments. Several new contributions are described. Firstly, we introduce a new 3D regional point descriptor based on a generalization of 2D gestalt features [18], [20]. Secondly, we provide a methodology for statistical modeling and analysis of keypoint vote distributions, thus enabling automatic tuning of critical algorithmic parameters. Thirdly, we extensively validate our approach in large-scale and challenging scenarios that we believe go beyond what has been previously demonstrated in the literature (or industry), including multiple kilometer scale mapping of rugged forest trails using a handheld sensor. Finally, we also make several of our unique datasets available to the research community for comparison of place recognition solutions or other algorithm development [21]. It should be noted that though our focus here is primarily on 3D mapping with lidar, many aspects of the keypoint voting-based solution we propose are generally applicable to a variety of sensing modalities.

II. APPROACH

We define a place as a collection of spatially contiguous, *local* measurements of the environment. A dataset is a set of N places $\{\pi_i\}$ which may or may not be registered with respect to one another. The *locality* requirement states that measurements within a place are locally registered to one another; that is, they contain a bounded amount of registration error[1]. For example, in this paper, we consider a place to be a set of temporally contiguous measurements (*i.e.*, we partition the dataset in time order). The place recognition problem is to determine whether a given pair of places (π_i, π_j) contain observations of the same physical region of the environment. Provided a pair of places match, a second objective is to estimate the spatial relationship (*i.e.*, a rigid 6DoF transformation) between the two matching places.

[1]If the amount of drift in an odometry solution is too large, the places become too distorted to recognize, as they may look significantly different on each pass.

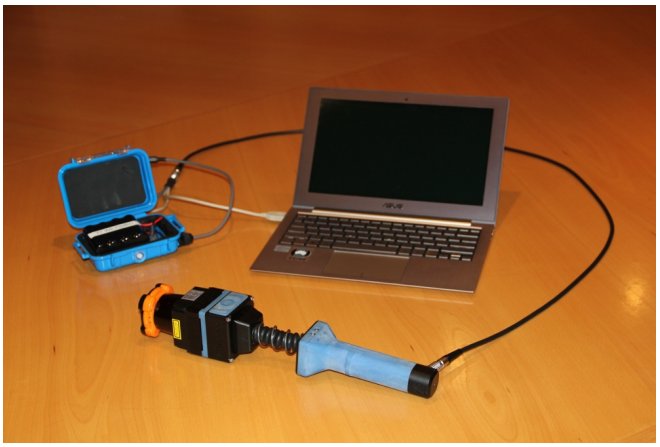

Fig. 2: The mapping hardware used to acquire several of the datasets. *Zebedee* consists of a Hokuyo UTM-30LX lidar scanner and a MicroStrain 3DM-GX3 MEMS IMU mounted on a spring connected to a handle. Also pictured are an acquisition laptop for recording the data via USB, and a lithium-ion battery pack (shown is a 77 Wh battery capable of supplying power to the sensors for over six hours of continuous operation.)

While this definition can apply to data from a variety of sensors, map representations, and mapping algorithms, in the remainder of this paper we focus on a specific hardware configuration and in particular 3D lidar data. We first describe the hardware used and the environments in which the datasets were collected, then elaborate on the details of our approach.

A. Data Acquisition

The primary sensor utilized for the experiments is a continuously scanning, handheld 3D lidar system called *Zebedee* [3] (Figure 2). The *Zebedee* mobile mapping system consists of a Hokuyo UTM-30LX 2D lidar scanner and a MicroStrain 3DM-GX3 MEMS IMU mounted on a passive spring. The spring converts the natural motion of the operator into non-deterministic motion of the scanner about the spring, thereby extending the field of view of the 2D lidar to three dimensions. Typical motion about the spring rotates the lidar at a frequency close to 1 Hz through approximately 160° in pitch, though significant components in roll also occur regularly. A 3D incremental SLAM algorithm is capable of producing an accurate 6DoF trajectory from the *Zebedee* lidar and inertial data stream, from which a 3D point cloud can be generated [3]. This open-loop trajectory can then be non-rigidly registered into a globally consistent closed-loop trajectory (and point cloud), provided the open-loop trajectory is within the algorithm's catchment basin for convergence. As the scale of a dataset increases—typical datasets considered here are between 40–60 minutes in acquisition time and up to several kilometers in trajectory length—it becomes more likely that the open-loop drift errors

of the trajectory are too large for the global registration algorithm to be able to converge. Matches identified from place recognition generate additional trajectory constraints thereby providing a coarse alignment to initialize the global registration process (analogous to a coarse alignment step for seeding the first iteration of a rigid point cloud registration method such as ICP [14]).

Data was acquired as the *Zebedee* operators walked through two distinctly different environments at a comfortable walking pace. The first, QCAT, is an environment consisting of indoor and outdoor spaces at a mixed office and industrial site (Figure 3). The trajectory followed an approximately 525 m circuit three times (twice in one direction, once in the reverse direction) in just over one hour; passing through a variety of spaces including office and cubicle areas, indoor and outdoor seating areas of a cafeteria, stairways, hallways, outdoor vegetated areas, and roads between large industrial sheds. The environment observed throughout each circuit in the trajectory contains only a minor degree of self-overlap, and therefore is topologically close to a single loop. The second environment considered is a natural open eucalypt forest with multi-use dirt trails located at Mt Coot-tha Forest in Brisbane, Australia. The forest dataset includes data from two circuit trails (for a total of 4.6 km trajectory length acquired over 93 minutes) traversed simultaneously by two operators, illustrated in Figure 1. The two paths overlap along a common 1.1 km segment, which was traversed in opposite directions by the two operators. On one of the handheld units, the IMU ceased logging partway through acquisition due to a USB disconnection, and therefore the trajectories for which data exist overlap by only about 1 km. The disruption of one of the acquisition systems provides an opportunity to demonstrate one of the use cases for place recognition in which one dataset is used to repair another overlapping dataset. The Mt Coot-tha Forest environment poses a perceptual challenge as it consists almost entirely of natural vegetation and terrain where most areas appear visually and structurally similar to one another. We have made the MT COOT-THA and QCAT datasets available for use by other researchers [21].

B. Place Recognition

Our framework for solving the place recognition problem follows a keypoint nearest neighbor voting approach [1], [2]. For a given query place π_i, the search proceeds as follow. First, a set of n_i keypoints are chosen from the place's local point cloud, and a descriptor is computed for each keypoint. Next, a database containing the descriptors from all other places is queried for the k nearest neighbors to each descriptor in place π_i [2]. Each descriptor match found votes for its associated place π_j, and the places with a sufficient number of votes (*i.e.*, above a threshold) are considered as candidates for place matches. The vote scores can be represented as a vote matrix, in which each entry v_{ij} is the sum of the votes for place π_j from all of the keypoints in query place π_i. For each candidate place match, a 6DoF alignment transformation T_{ij} is sought using a RANSAC-based approach [6] on the individual keypoint correspondences. Provided the RANSAC step finds a sufficiently geometrically consistent solution, its transformation can be refined using a closed form solution to the absolute orientation problem [9]. Additional bad matches can be filtered by identifying a lack of symmetry with respect to the complementary transformation $T_{ji} \approx T_{ij}^{-1}$. If it is necessary to compute a consistent global pose for each place (*e.g.*, in mapping applications), we can optimize a pose graph where the nodes are places and the edges are transformations derived from the open-loop trajectory information or the detected place matches. The pose graph optimization algorithm we use is robust to outlier matches; we do not, therefore, require 100% precision in our place recognition. The resulting solution provides a good initial trajectory for our non-rigid global trajectory optimization algorithm [3].

[2]We use the libnabo [5] implementation of *kd*-tree search available from `https://github.com/ethz-asl/libnabo`.

Fig. 3: The QCAT dataset includes office, industrial, and vegetated areas.

C. Keypoint Selection

As a preprocessing step, the point clouds from the locally registered scans are down sampled using a pair of voxel grids of 0.4 m resolution. The two grids are offset by half the resolution in each dimension. Each grid cell that contains at least three lidar returns within a one minute time window is represented by the centroid of those points.

A random ten percent of the reduced points are selected as keypoints (points at which descriptors are computed). For each keypoint, the orientation and descriptor are computed from points within a fixed radius and time window as follows: To determine the orientation, the 3×3 covariance matrix of the keypoint's neighborhood is computed and decomposed into eigenvalues and eigenvectors. The global up vector is assumed known and used together with the eigenvector corresponding to the smallest eigenvalue, λ_1, to determine the keypoint's local orientation frame. The smallest eigenvector is projected onto the horizontal plane and defines the local x-axis, and the local z-axis is set to

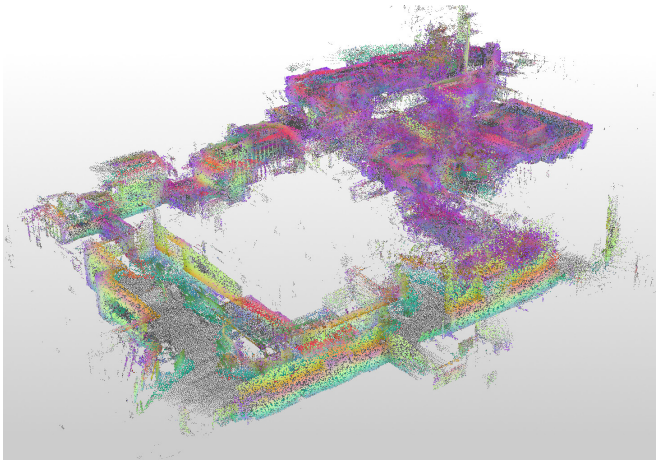

Fig. 4: Point cloud of QCAT with measured points in gray and keypoints in color according to the first three feature dimensions. Note that there are no keypoints in the large planar and horizontal surfaces. The point cloud dimensions are roughly $100 \times 60 \times 23$ m (covering four floors).

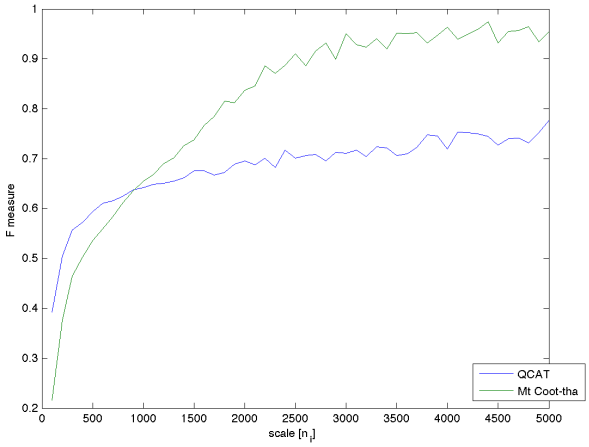

Fig. 5: Maximum F-measure vs. keypoint neighbor scales in different environments. We observe that although the MT COOT-THA forest environment can achieve better F-measures, it needs larger scaled places than the QCAT environment before it plateaus.

the global up-vector. Keypoints with a smallest eigenvector that is within $10\,\mathrm{deg}$ of vertical are discarded since the x-axis direction will not be reliable. The planarity and cylindricality (p, c) of the neighborhood is computed from the eigenvalues $p = 2 * (\lambda_2 - \lambda_1)/\Sigma_i(\lambda_i)$ and $c = (\lambda_3 - \lambda_2)/\Sigma_i(\lambda_i)$. Keypoints with a planarity $p > .9$ are discarded since such a neighborhood has little descriptive value.

D. The Gestalt Descriptor

Once the keypoints have been selected and their orientations determined, we use a 3D adaptation of the gestalt descriptor [18], [1] to encode each keypoint's neighborhood as a vector. As long as the keypoint neighborhoods have sufficient samples, the descriptor is designed to be invariant to the density of the lidar points. This design is necessary for descriptors to match well between environments scanned from slightly different trajectories or different lidar sensor configurations. The descriptor is formed by setting up a polar grid about the keypoint's local z-axis. There are 8 azimuthal divisions and 4 linear spaced radial divisions for a total of 32 bins. The mean and variance of the heights within each bin is computed and included in the descriptor vector. If any bins are underpopulated, then the values are copied from the next valid bin in the direction of the keypoint. There are 66 raw descriptor dimensions consisting of the mean and variance in each bin along with the neighborhood's planarity and cylindricality coefficients.

The raw descriptor vector space is normalized and reduced to a lower-dimensional feature vector (typically around ten dimensions) using the a procedure as outlined in our previous work [1]. The descriptor space normalization first uses a polynomial fit to the sample cumulative distribution of each raw dimension to remap the values such that they will have a Gaussian distribution. Quadratic discriminant analysis is used on a training set of matching and non-matching keypoint descriptors to compute a linear transform on the feature space such that the Euclidean distance between two features is metrically meaningful (in particular, proportional to the negative log likelihood ratio). This transformation enables the use of standard efficient k-nearest neighbor search algorithms which are typically formulated for Euclidean distance-based search. The training data is easily generated from a globally registered dataset where inliers are chosen from keypoints that are spatially close in position and orientation, but temporally distinct, (*i.e.*, they come from different passes of the same environment). The outliers are sampled from randomly chosen keypoint pairs that are spatially distinct. Figure 4 depicts the keypoints colored by the first three normalized feature dimensions as red-green-blue. By trialling various keypoint scales and measuring the precision and recall of k-NN searches for matching keypoints, we have determined empirically that the optimal scale of the keypoint's neighborhood depends on the type of environment. Figure 5 shows the maximum F-measure for various keypoint neighborhood scales in different environments. The F-measure is the harmonic mean of precision p and recall r rates:

$$F = 2\frac{pr}{p+r} \tag{1}$$

We have also determined empirically that linearly spaced radial bins produce better descriptors than logarithmically spaced bins.

In our previous work [2] we compared several 3D regional point descriptors in a place recognition scenario, achieving the best performance with moment grid and shape context descriptors. Having carried out further work in a wider range of environments, however, we have observed better performance with the 3D gestalt descriptors introduced here. Other descriptors for both place and object recognition have been proposed recently (several of which are implemented in the Point Cloud Library [13]); however, many rely on estimates of surface normals for each point [12], [17] which are unreliable in natural environments such as the forest datasets considered here. Still other descriptors [17], [11]

are designed for range image representations, which are not applicable to all sensing configurations (such as *Zebedee* or 2D lidars rotating about the central scan ray while under continuous translational motion). The line image descriptor [11] has some similarity to gestalt. The line image is defined for range image representations, but could potentially be adapted for use with arbitrary point clouds.

E. Place Partitioning

Since our data is collected by continually scanning sensors, there is no natural mechanism to determine where one place starts and another ends. We therefore divide up the data into temporally contiguous chunks. Place boundaries are computed in such a way that each place will have approximately the same number of keypoints. This strategy ensures that each place will be represented by a sufficient number of keypoints and removes some of dependence on the data collection speed, trajectory, and sensor configuration.

F. Modeling and Analysis

The votes that are aggregated for every possible place match give an indication as to how likely different places match; it is not, however, a trivial task to pick suitable algorithmic parameters such as the number of neighbors k for which to search and a threshold on the vote scores v_{ij}. Previously, we have simply chosen a fixed value for k and rather than thresholding the vote scores, have included the best few matches for every place (*i.e.*, the top scores in each row of the vote matrix) [1], [2]. This does not, however, work well when there is uncertainty and variability in the number of passes through each region in the environment. It is important to pick a good threshold to ensure both high recall and high precision for place recognition. In other words, the recall must be high enough so that loop closures are not missed, and the precision must not be so low that the later match verification stages are overwhelmed by false alarms. Note that the algorithm does not require 100% precision for the vote score thresholding, since the geometric verification and the pose graph optimization can filter out false alarms. Too many false alarms, however, will result in increased computation time. By accurately modelling the statistics of the votes scores, an automatic threshold can be set that performs well. A statistical model will also help to automatically choose the number k of neighbors to search for and the size n of each place such that there are sufficient votes to distinguish matching from non-matching places.

We can model the vote scores v_{ij} as a counting process which depends on the number k of nearest neighbor votes for each keypoint, the number n_i of keypoints in each place, the probability $p(m_1|k)$ of finding a matching keypoint at the k^{th} nearest neighbor and the probability $p(m_0|j)$ of finding a non-matching keypoint in place π_j.

$$v_{ij} = \sum^{n_i} \sum_{k} \left(p(m_1|k) + (1 - p(m_1|k))\, p(m_0|j) \right) \qquad (2)$$

The match probability $p(m_1|k)$ depends on the amount of overlap between places π_i and π_j and the saliency of the descriptor. Since we don't know ahead of time the amount of overlap between places, this probability is difficult to model; however, if we have a good model for the non-matching probability $p(m_0|j)$, then we can set the threshold on v_{ij} to a level that is unlikely to be accounted for by non-matches alone. In this paper, we will focus the analysis primarily on $p(m_0|j)$ and leave the analysis of $p(m_1|k)$ for future work.

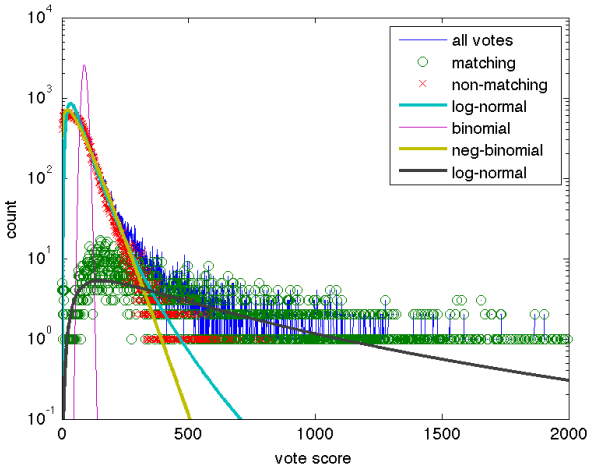

Fig. 6: An example histogram of vote scores from QCAT with various models fitted.

The simplest model for the non matches is one where they are distributed uniformly among all places such that $p(m_0|j) = n_j/N$ and thus v_{ij} is binomial distributed $\mathcal{B}(kn_i, n_j/N)$. Though this model accurately predicts the mean score $\mathrm{E}[v_{ij}] = kn_in_j/N$, the sample variance is significantly higher than as would be predicted $\mathrm{Var}[v_{ij}] >> kn_i(n_j/N)(1-n_j/N)$, since votes tend to clump in similar (but not exactly matching) places. Using a log-normal distribution instead of a binomial distribution seems to provide a much better fit to the data as illustrated in Figure 6.

$$P\left(v_{0ij} = x\right) \approx \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right) \qquad (3)$$

where $v_{0ij} = \sum^{n_i} \sum^{k} p(m_0|j)$ is the vote score when places π_i and π_j do not match.

The intuition behind using a log-normal distribution is that the vote scores are the combination of several underlying counting processes each with different growth rates. As enough of these processes are combined, their distribution in log space approaches a normal distribution. Other over-dispersed counting process models (such as the negative binomial) could also be used, but the log-normal has a heavier tail and is relatively straightforward to fit to data.

The next step of the analysis is to determine how the parameters (μ, σ) of the log-normal distribution vary with k, n_i, and N for particular environments. We have determined empirically that the parameters take on the following

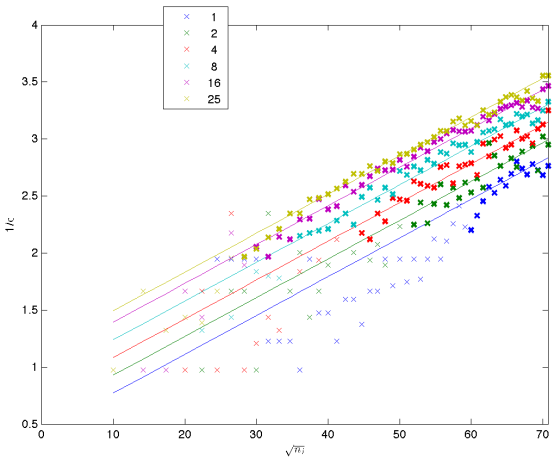

Fig. 7: Model fit to $1/\sigma$ of log-normal distribution. The data points in bold were used to fit a linear model.

form:

$$\mu = a\ln(\frac{k}{N}) + b + \ln(\frac{n_i n_j}{N}) \quad (4)$$

$$\frac{1}{\sigma} = c\ln(\frac{k}{N}) + d\sqrt{n_i} + e \quad (5)$$

where the parameters a, b, c, d, e are fit to the sample means $\bar{\mu}$ and sample variances $\bar{\sigma}^2$ of non-matching vote matrix elements that have been generated with varying trials of input parameters k, n_i and N. To vary N we computed the vote matrices on truncated datasets. When the mean vote score is low, the discrete nature of votes distorts the sample mean and variance of the log vote scores. Therefore, for data fitting, we use the following robust estimate of $\bar{\mu}$ and $\bar{\sigma}$ on trials where $\frac{kn_i n_j}{N} > 20$. (See Figure 7.)

$$\bar{\mu} = Q_2 \quad (6)$$

$$\bar{\sigma} = (Q_3 - Q_2)/0.67 \quad (7)$$

where Q_2 is the median, and Q_3 is the third quartile or 75^{th} percentile of the data. Table I lists the fitted parameters for the two main datasets.

TABLE I: Log-normal model fit parameters.

	a	b	c	d	e
MT COOT-THA	1.01	13.4	0.224	0.0341	3.43
QCAT	1.11	13.1	0.123	0.0085	2.34

With the fitted model, we can automatically set a threshold t_{ij} on the vote score for which is unlikely to come from a non-matching place:

$$t_{ij} = \exp(\mu + 3\sigma) \quad (8)$$

Since the threshold for the vote scores can be determined before computing all the votes, there is no need to store the complete vote matrix. A hash table can be used aggregate the votes, and once a score is large enough, it can be added to a list of candidate matches for further verification. Therefore the voting can run in sub-linear time per place.

The fitted model also provides some insights into the voting process. When the predicted parameters are non-sensical (such as negative variances), then there will be insufficient votes. In some applications, it is also possible to increase the size of the local place n_i such that enough votes are aggregated. If n_i becomes too large, then it will become more likely to miss places that only partially overlap such as path intersections.

III. EXPERIMENTS

For the experiments, we run our place recognition algorithm on various datasets of open-loop trajectories. We use a manually verified closed-loop trajectory as ground truth from which to compute precision-recall curves and make quantitative analyses. To obtain a single metric which evaluates place recognition performance, we use the F-measure, as defined in Equation 1.

In addition to the QCAT and MT COOT-THA datasets described in Section II-A, we consider two publicly available lidar datasets. The HANNOVER2 dataset consists of measurements acquired from two vertically mounted SICK LMS291 scanners rotating about a common axis [19]. The sensors were mounted on a wheeled vehicle following a 1.2 km trajectory containing several loops of various sizes at the Leibniz Universität Hannover. An (unregistered) odometry sequence is provided based on wheel encoders and a 3-axis gyro. The lidar data is segmented into 922 1.2-second scans, each of which is projected into a common frame according to the local odometry.

The NEWCOLLEGE dataset [15] contains lidar data acquired from two trawling SICK LMS291 scanners mounted on a Segway RMP with the scan planes aligned (nominally) vertically and facing towards either side of the platform. The data were acquired while driving 2.2 km around a section of the Oxford University campus and an adjacent park. The path consists of three loops, each of which are traversed multiple times. The trajectory estimate used to generate our initial point cloud is a fusion of the provided platform odometry with the yaw estimate from a visual odometry solution provided on the authors' website (the yaw estimate from the platform odometry drifts too quickly to ensure the locality condition defined in Section II is met).

We further consider two additional datasets for the purpose of independently training the descriptor normalization function. OFFICE2 is a dataset collected in another building containing offices, conference rooms, cubicles, stairways, labs, and highbay space. FOREST2 is a second natural forest dataset collected in Victoria, Australia.

Each dataset can be fully processed (open-loop, place recognition, closed-loop) on a 2012 MacBook Pro in less time than the acquisition time. Figure 8 tabulates the place recognition results for the four datasets. The vote score matrices are depicted with the final optimized place adjacency and candidate matches that did not pass the geometric validation tests. The closed-loop trajectory is also shown with colored links for the candidate place matches. In most of the datasets there are a few false positive candidates that slip through

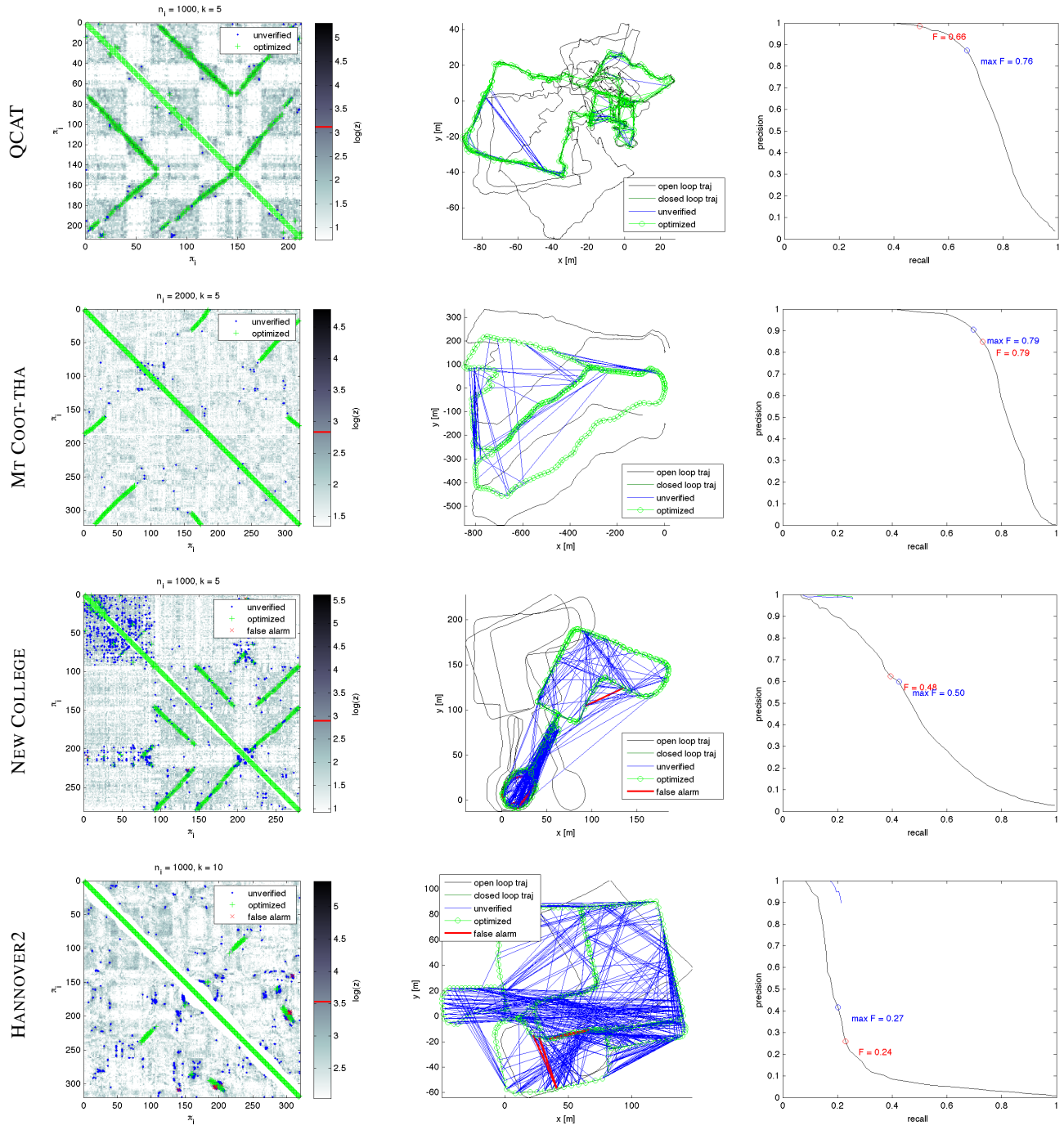

Fig. 8: Place recognition results on various datasets. For visualization purposes we depict the logarithm of the vote scores normalized by the mean votes per keypoint $z_{ij} = \log(\frac{N}{n_i n_j} v_{ij})$. The place adjacency for the various processing stages is superimposed on the vote matrix. The vote score threshold is indicated by the red line on the colorbar. The blue links are matches that are larger than the vote threshold but do not pass the geometric verification. The red links are false alarms that pass the geometric verification and are indicative of perceptual aliasing, but the pose graph optimization is robust to their presence. In all cases, an independent training set was used for training the descriptor normalization function.

the geometric validation; however, they are easily identified in the pose graph optimization step. Table III lists provides sample computation times for various stages of the algorithm.

Table II shows the maximum F-measure for various datasets when another dataset is used to train the descriptor transform. We also evaluate the effect of using a random ten-dimensional subspace on which to project the keypoint descriptors, rather than using a trained transformation. As expected, the tests with a trained transformation outperform the random transformation. This experiment also indicates

TABLE II: Maximum F-measure for each dataset with descriptor transform trained using another dataset. In the last column, a random ten-dimensional subspace is used to transform the raw descriptors. All descriptors were generated using a radius of 4 m.

DATASET training → ↓ testing	QCAT	OFFICE2	MT COOT-THA	FOREST2	NEWCOLLEGE	RAND
QCAT	0.69	0.68	0.71	0.67	0.69	0.53
OFFICE2	0.51	0.52	0.53	0.51	0.53	0.40
MT COOT-THA	0.71	0.68	0.78	0.64	0.71	0.64
FOREST2	0.50	0.48	0.50	0.53	0.52	0.38
NEWCOLLEGE	0.74	0.75	0.73	0.72	0.74	0.69
HANNOVER2	0.18	0.17	0.20	0.19	0.19	0.09

TABLE III: The breakdown of running times for the various stages of the algorithm as descriptor generation, k-NN lookups and descriptor voting, geometric match verification, and pose graph optimization, (t_d, t_n, t_g, t_o). Note that the descriptor generation and lookups are implemented in optimized C++ code, whereas the verification and pose graph optimization are currently unoptimized Matlab code.

DATASET	time (dist)	N_k (N_p)	t_d	t_n	t_g	t_o
QCAT	60min (1.6km)	209K (212)	25s	14s	32s	15s
MT COOT-THA	91min (4.6km)	637K (320)	28s	52s	16s	14s
NEWCOLLEGE	43min (2.5km)	278K (279)	16s	23s	15s	34s
HANNOVER2	18min (1.2km)	318K (319)	37s	28s	24s	27s

that the place recognition performance is not too sensitive to the data used to train the descriptor transform. It is more important to have a good choice for the vote score threshold.

IV. CONCLUSIONS

We have demonstrated an efficient place recognition approach using keypoint voting and geometric verification. The distributions of the keypoint vote scores for non-matching places is modeled as a log-normal process whose hyper-parameters can be empirically determined for a given type of environment. The model of the distribution is used to automatically determine a threshold for candidate places and can be set before the voting is computed. It is thus possible to find place matches in sub-linear time per place for large scale datasets. We have demonstrated our approach in two different and challenging environments using a handheld continuously scanning lidar sensor. We have also demonstrated that this approach works on publicly available lidar datasets.

In our future work, we plan to extend the modeling analysis of the vote scores to the matching places so that we may formulate a maximum a priori threshold and gain a better understanding of selecting the optimal number of keypoint neighbors to search for as the number of place overlaps increases. We would also like to formally compare our voting base approach to methods based on bag-of-words models, and compare and evaluate other descriptor models from the literature.

ACKNOWLEDGEMENT

We wish to thank Paul Flick for his assistance in designing and constructing the *Zebedee* hardware.

REFERENCES

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

[2] M. Bosse and R. Zlot. Place recognition using regional point descriptors for 3D mapping. In *International Conference on Field and Service Robotics*, 2009.

[3] M. Bosse, R. Zlot, and P. Flick. Zebedee: Design of a spring-mounted 3D range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5):1104–1119, October 2012.

[4] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6):647–665, June 2008.

[5] 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.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[7] D. Gálvez-López and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.

[8] K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos. Learning to close loops from range data. *International Journal of Robotics Research*, 30(14):1728–1754, December 2011.

[9] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629, 1987.

[10] M. Magnusson, H. Andreasson, A. Nüchter, and A. J. Lilienthal. Appearance-based loop detection from 3D laser data using the normal distributions transform. In *IEEE International Conference on Robotics and Automation*, 2009.

[11] A. Quadros, J. P. Underwood, and B. Douillard. An occlusion-aware feature for range images. In *IEEE International Conference on Robotics and Automation*, 2012.

[12] R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation*, 2009.

[13] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation*, 2011.

[14] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, May 2007.

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

[16] B. Steder, M. Ruhnke, S. Grzonka, and W. Burgard. Place recognition in 3D scans using a combination of bag of words and point feature based relative pose estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[17] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard. NARF: 3D range image features for object recognition. In *Proceedings of the IEEE/RSJ Int'l Conf on Intelligent Robots and Systems Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics*, 2010.

[18] A. Walthelm. Enhancing global pose estimation with laser range scans using local techniques. In *International Conference on Intelligent Autonomous Systems*, 2004.

[19] O. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner. Ground truth evaluation of large urban 6D SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[20] R. Zlot and M. Bosse. Place recognition using keypoint similarities in 2D lidar maps. In *Int'l Symposium on Experimental Robotics*, 2008.

[21] R. Zlot and M. Bosse. Mt Coot-tha Forest and QCAT 3D point cloud datasets. CSIRO Data Collection, 2013. DOI: 10.4225/08/511869FB9F5AB.