Recognizing places using spectrally clustered local matches

Edwin Olson
ebolson@umich.edu
Electrical Engineering and Computer Science Department
University of Michigan

2260 Hayward Street
Ann Arbor, MI 48109-2121
tel: 734-647-1049
fax: 734-763-1260 (shared)

September 16, 2009

Abstract

Place recognition is a fundamental perceptual problem at the heart of many basic robot operations, most notably mapping. Failures can result from ambiguous sensor readings and environments with similar appearances. In this paper, we describe a robust place recognition algorithm that fuses a number of uncertain *local* matches into a high-confidence *global* match. We describe the theoretical basis of the approach and present extensive experimental results from a variety of sensor modalities and environments.

Keywords: Data association; Simultaneous Localization and Mapping (SLAM); Spectral Clustering; Laser Scan Matching; SIFT

1 Introduction

Place recognition is a key problem for many mobile robot applications. It plays a central role in map building: without the ability to recognize previously-visited places, the position uncertainty of a robot increases without bound due to the ceaseless accumulation of dead-reckoning error. Place recognitions serve as constraints on the motion of the robot, allowing a correction of its dead-reckoning error. In the mapping context, place recognition is called "loop closing": if the robot drives in a loop and back to its starting position (and recognizes it), the robot "closes" the loop. With a greater number of loop closures, the accuracy of the map increases.

Place recognition is useful outside of metrical mapping. The topology of an environment is also discovered through place recognition. Place recognition is also useful for robots that do not explicitly build maps: service robots can use place recognition to identify charging stations and to determine when the robot has reached its desired destination.

Place recognition is difficult for two basic reasons. First, algorithms must be robust to modest changes in the environment caused by things like moving chairs or humans. These changes tend to make the same environment look different. Second, robots must be able to distinguish similar-looking environments. This problem is exacerbated by the practical limitations of many sensor systems. For example, The data produced by laser scanners is metrically accurate, but distinct environments can appear similar (see Fig. 1). Specifically, indoor environments tend to be composed of sets of straight walls and corners whose appearances are similar.

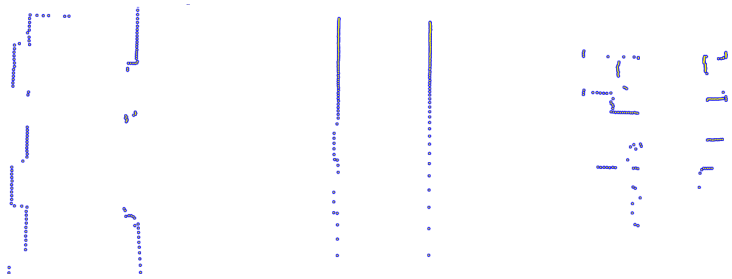

Figure 1: Three laser scans from the CSAIL dataset. Some environments present rich alignment cues (left: an elevator lobby). Incorrect matches arise from both spartan areas (middle: a corridor) and from repetitive/cluttered areas (right: office cubicles).

Camera-based systems, like those using the SIFT feature detector [1], provide a richer description of the environment (able to distinguish different posters on a wall, for example), but are still susceptible to ambiguity. Different offices, for example, may contain the same type of chair, and different outdoor environments contain the same types of street signs. The length of these feature vectors has a major impact on both total memory usage and the time required to index and search for similar features. Consequently, it is desirable to use shorter descriptors (such as PCA-SIFT [2]). Reducing the descriptor size generally increases the error-rate of matching: matching algorithms that are robust to higher error rates (like the one described in this paper) are thus very desirable. Another major challenge with vision-based systems is dealing with dynamic and self-induced features: moving objects and images of the robot (or its shadow) can lead to incorrect matches (see Fig. 2).

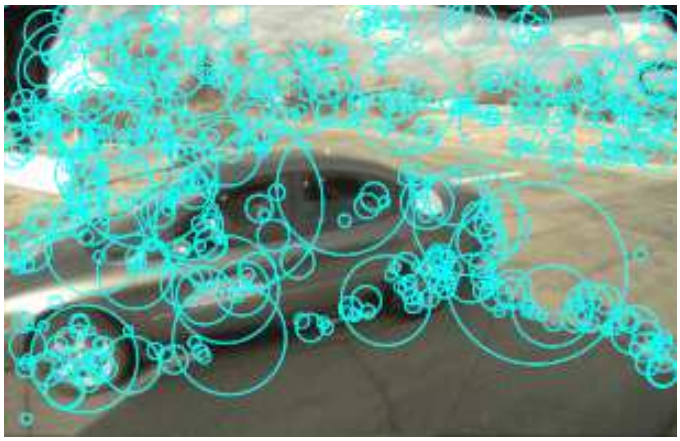

Figure 2: Sample SIFT detections. Cyan circles indicate SIFT features. Note how the other car, the shadow cast by our own car, and portions of our own car (black regions on left and top) cause undesirable SIFT detections.

Some authors have proposed using globally unique beacons, such as RFID tags [3], wireless networking base stations [4], or specialized sensor networks [5]. The resulting descriptors are concise and globally unique, but require the deployment of beacons and are not broadly applicable to all robotics applications.

Even when GPS data is available, its accuracy (especially in consumer-grade devices) is insufficient for many applications. For example: while GPS can usually tell a car what *road* it is on, it cannot reliably indicate which *lane* it is in. For many applications, greater accuracy is needed. This additional accuracy can be achieved by recognizing places: each recognition allows the map to be refined.

In a map-building context, place recognition is known as "loop closing", reflecting the common case in

which a robot travels around a large loop and returns to its original position (thus "closing" the loop). However, any place recognition adds an edge to the pose graph, creating a new cycle— the robot does not need to physically travel in a circle in order for a loop closure to occur.

Loop closing has been approached in a variety of ways. The discovery of topological loop closures has been performed with odometry alone using hidden Markov models [6]. Folkesson [7] uses negative information to validate loop closure hypotheses, while Kuipers uses bootstrap learning to identify places [8]. Bosse validated his loop closures using a loop-consistency test [9].

In robotic mapping applications, place recognition has generally been cast in terms of explicitly associating landmarks with previously observed landmarks. This process of *data association* on one (or just a few) features at a time is highly susceptible to errors, and data-association errors can cause catastrophic divergence of the map.

In an attempt to decrease their error rate, data association algorithms typically perform association on groups of observations at once. In analogy, it is easier to recognize astronomical constellations than single stars. Given the mean and covariance of a set of tracked landmarks, the probability of a set of observations matching that set can be computed. However, as the number of landmarks grows, the number of possible assignments grows exponentially. Many authors have proposed solutions to this problem, providing reasonably efficient search heuristics that often perform well [10, 11]. Error rates can also be addressed by revisiting data associations previously made [12].

Even if the number of landmarks is kept small enough to ensure good performance, such methods still require that the position estimates and covariances of landmarks be maintained. This is a major imposition for robots that do not otherwise care about landmarks and would not otherwise maintain their covariance. It is also an imposition for mapping robots that use an optimization method in which landmark covariance data is not readily available, such as FastSLAM [13], information-form filters [14, 15, 16], non-linear optimization methods (Gauss-Seidel relaxation [17], Multi-Level Relaxation [18]), and Stochastic Gradient Descent [19, 20, 21] its derivatives [22, 23].

Local matching methods are susceptible to false positives due to the fact that different places can have similar appearances. The main contribution of this paper is a principled means of determining whether a local match is globally correct. Our approach incorporates outlier and ambiguity testing based on Single Cluster Graph Partitioning (SCGP) [24]. Like CCDA [11], SCGP uses a pair-wise consistency graph, but is algorithmically more similar to other spectral methods [25, 26].

Local matching, which uses only a tiny subset of the data available to the robot, is generally computationally cheap in comparison to global data association. In this paper, we show how to combine local matching with a computational inexpensive and effective means of identifying false positives, resulting in efficient and reliable place recognitions. We can thus recognize places without performing explicit and global data association.

We are not the first to attempt to circumvent the difficult data-association problem. Cummins and Newman describe a scheme that is based solely on local sensor data [27] as do Ho and Newman [28]. In the absence of *some* knowledge of position, however, it is difficult to determine what constitutes incontrovertible evidence that the robot is in a particular place. In this paper, we argue that the amount of evidence required to reliably determine that two places are identical should scale with the robot's positional uncertainty.

Our approach is similar in several ways to that of Gutmann and Konolige [29], in that we attempt to find local matches within a search area provided by a prior, then combine multiple sensor scans to yield larger (and less ambiguous) local matches. Rather than correlating matches over relatively large submaps, we show how ambiguity can be detected from an ensemble of smaller pose-to-pose matches. We also address the issue of how large a local match must be in order for a loop closure to be reliable.

1.1 Contributions

In this paper, we describe an approach that recognizes places by performing a number of simple pose-to-pose matches. Given a number of naive matches ("hypotheses"), our goal is to identify those that are correct. A *set* of these hypotheses has an interesting property: the correct hypotheses all agree with each other (since there is only one true configuration), whereas the the incorrect hypotheses tend to be incorrect in

different ways (and thus do not agree with each other). Our algorithm exploits this property by computing the subset of hypotheses that is most self-consistent. In comparison to the exponential cost of performing data association amongst N features, the cost of our approach is $O(N^2)$ in the number of hypotheses. Our approach is similar to Combined Constraint Data Association (CCDA) [11], except that our notion of consistency is not limited to discrete boolean quantities. Algorithmically, our approach incorporates ideas similar to other spectral methods [25, 26].

Our approach does not require landmarks to be tracked, and thus does not require covariance information about the landmarks or a data association algorithm. All that is required is the ability to compute possible rigid-body transformations between two robot poses. Our approach can be viewed as a outlier-rejection step, extending previous pose-to-pose methods [30].

A major advantage of our proposed method is that it computes the "second-best" set of mutually-consistent hypotheses as well. When using Random Sample Consensus (RANSAC) [31] or Joint Compatibility Branch and Bound (JCBB) [10], the second-best solution is generally a trivial variation on the best solution: it is thus not very informative. In contrast, the second-best solution reported by our new approach is *orthogonal* to the best solution— i.e., represents a substantially different interpretation of the data. Our method allows the quality of these two solutions to be quantitatively compared, allowing ambiguous solutions to be safely discarded. (If the data can be equally well explained in two different ways, it is not safe to trust either explanation.) In these ambiguous cases, our method will occasionally reject hypotheses that are correct. Rejecting correct hypotheses has an impact on the quality of the final map (since some information is lost), but it does not result in the sort of rapid divergence that accepting an incorrect hypothesis can cause.

Finally, we describe a geometric test that allows us to determine when an ensemble of local matches is sufficiently unambiguous to assert a global loop closure. This test is based on the collective size of the submaps implied by the local matches versus the pose uncertainty. This "global sufficiency test" plays an important role in reducing false positives.

2 Method Overview

Our place recognition algorithm operates in several stages (see Fig. 3). Given two robot positions (at different points in time), we can estimate the probability that those two positions are close enough together for them to have observed the same environmental features. If so, we attempt to generate a hypotheses relating the position of those poses. The particular hypothesis-generation algorithm varies according to the types of sensors being used: cameras and lidars, for example, produce very different types of data that require different matching algorithms (see Section 3.2 for descriptions of our implementations).

We group the resulting hypotheses into "hypothesis sets". Within each set, we will identify those hypotheses that are correct. We begin by considering pairs of these hypotheses at a time, computing the consistency of each pair. Using an algorithm based on the spectral properties of the pair-wise compatibility matrix, we can then identify the subset of hypotheses (potentially consisting of more than two hypotheses) that is most consistent as a whole.

The resulting hypotheses must satisfy two different criteria: a local uniqueness test and a global sufficiency test. The local uniqueness test ensures that there is no similar loop closure nearby. In other words, if a place recognition hypothesis passes the local uniqueness test, that hypothesis is either correct or significantly wrong. This test addresses the picket-fence problem, where it is possible to find large local matches (comprising many fence posts) that are locally non-unique (see Fig. 4); in the case of picket fences, good matches can be obtained by translating the local match by one fence post. The local uniqueness test detects these problems in a general and mathematically elegant way, eliminating potential false positives.

The global sufficiency test ensures that the size of the local match is large enough to be unambiguous with respect to the positional uncertainty of the robot. Intuitively, if the robot knows about where it is, very limited local matches (such as observing a single feature) can be unambiguous. However, if the robot's positional estimate is highly uncertain, the same match is no longer sufficient. The global sufficiency test ensures that the place recognition hypothesis is sufficiently strong for the robot's positional uncertainty.

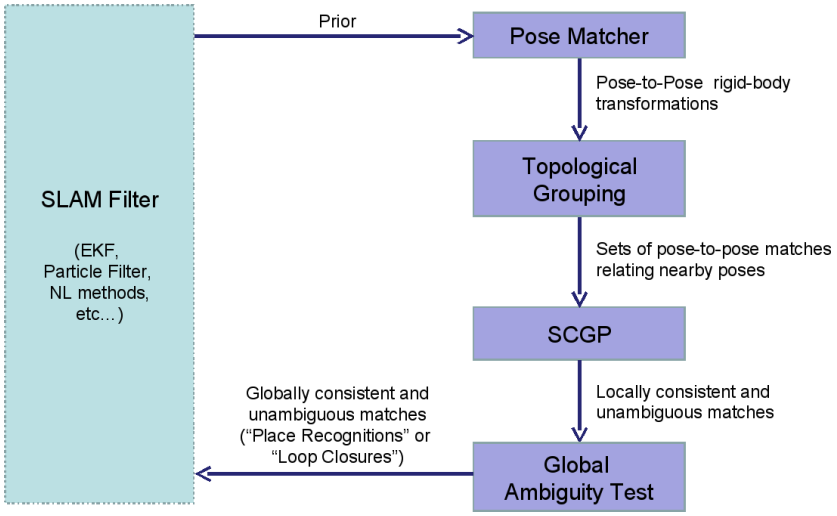

Figure 3: Method Overview.

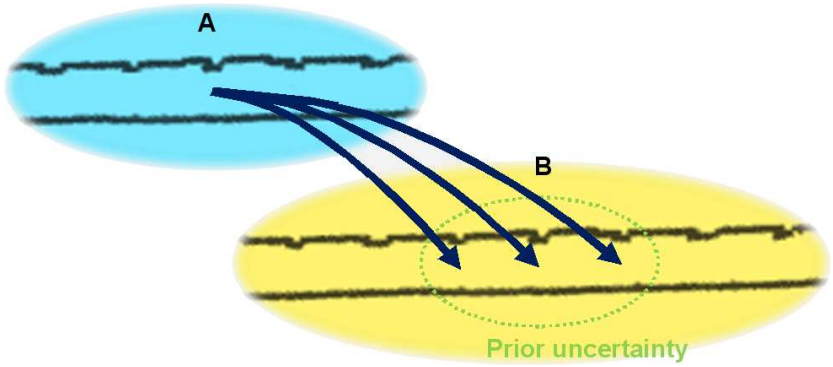

Figure 4: Picket Fence Failure Mode. In self-similar environments, it is possible to find large local matches that are not locally unique: small translations can result in another possible place recognition. Any robust place recognition algorithm must address these failures.

We will illustrate our algorithm with a variety of real world data. Section 3 describes how we generate hypotheses in a variety of domains. These hypotheses tend to be very good, and so is relevant to those interested in sensor processing. But these local methods are also imperfect: like any local method, they are susceptible to false positives. The central focus of this paper is filtering these hypotheses, as described in Section 4.

3 Hypothesis Generation

3.1 Determining when poses overlap

The first step in identifying loop closure hypotheses is to identify those poses that might contain overlapping sensor readings. Given a pose a, we wish to find all other poses in the graph whose sensor readings might overlap those from pose a. This could be determined from the posterior distribution by marginalizing out pose a and any landmarks. Instead, we describe a conservative approximation that is much easier to compute. In particular, our method does not require knowledge of the posterior pose and landmark covariances. This

is important, since storing landmark covariances requires $O(N^2)$ memory, and maintaining those covariance requires $O(N^3)$ computation time. (Information form filters can do somewhat better, though recovering the mean now becomes an expensive operation.)

3.1.1 Dijkstra Projection

We estimate the relative positional uncertainty between nodes a and b by searching for the minimum-uncertainty path through the pose/feature graph that connects them. This path can be found by using variant of Dijkstra's Algorithm [32], and its application to map building is due to Bosse [9]:

Algorithm 1 Dijkstra Projection from node a

```
Initialize a set of paths P = {}
for all edges e leaving node a do
  P = P ∪ e
visited(a) = true
while some nodes unvisited do
  find p, the minimum-uncertainty path in P
  if !visited(destination(p)) then
    optimal-path(b) = p
    visited(b) = true
    for all edges e leaving node b do
      P = P ∪ compose-transforms(p, e)
```

In order to compare the uncertainty of two paths, we use the determinant of the paths' covariance matrices. Geometrically, the determinant can be interpreted as the area of the covariance ellipse, and is thus proportional to the area of the region in which pose b is likely to be found given the position of pose a. Larger search areas generally result in more computational effort. Consequently, picking the path with the smaller determinant generally minimizes the amount of computation needed to find pose b, unlike other measures of uncertainty like the trace of the covariance.

The Dijkstra Projection is conservative in the sense that it will overestimate the uncertainty. For example, if a pose graph contains multiple independent paths between nodes a and b, the uncertainty of their relative positions will be less than that computed by the Dijkstra projection. Over-estimating the uncertainty will result in larger search areas and thus greater computational effort when searching for loop closures.

However, even if the exact covariance could be easily computed, the search areas are often artificially enlarged in order to ensure that loop closures can be found whenever possible. Over-confident sensor-noise characterizations, for example, cause the "exact" covariance to be too small, and can result in missed loop closures. If loop closures are consistently missed, the quality of the map will suffer. In short, the conservatism of the Dijkstra algorithm is not problematic. Anecdotally, the bounds are often tighter than one might expect: the relative uncertainty between two paths is generally dominated by the single shortest path between them.

3.1.2 Sensor range

In addition to the relative positions and uncertainties of two nodes, we must know the sensor range in order to determine whether two sensor scans might overlap: large sensor ranges make it more likely that nodes will have overlapping fields of view. In indoor environments, the effective sensor range varies dramatically depending on the environment: a robot may only be able to sense a few meters when inside an office, but may be able to sense tens of meters outdoors.

We model the sensor range at a particular node a as a circle with center c_a and radius r_a. We compute c_a as the centroid of the landmarks observed, with r_a being the average distance of the landmarks to the centroid. In order to test whether sensor observations at node b might overlap with those at node a, we similarly use a circle for node b with center c_b and radius r_b, where the relative position of c_a and c_b is provided by the Dijkstra projection between nodes a and b. Let Σ_{ab} the uncertainty of the transformation

from a to b as given by the Dijkstra projection. We can write the Mahalanobis distance of the likelihood that the two sensor ranges overlap as:

$$\Delta c = (c_b - c_a) \quad (1)$$

$$s = \max(0, ||\Delta c|| - r_a - r_b)\frac{\Delta c}{||\Delta c||} \quad (2)$$

$$mahl = s^T \Sigma_{ab}^{-1} s \quad (3)$$

In the equations above, Δc is the Euclidean vector between the circle centers, and s is the separation vector between their sensor ranges (the vector Δc reduced in magnitude by the sum of the sensor ranges). If the resulting Mahalanobis distance is less than 3, it is reasonably likely that the two nodes have overlapping sensor data. In aggregate, our approach for computing the pair-wise "compatibility" is similar to the metric used by GraphSLAM [33], with the improvement that we explicitly consider sensor range. Our use of these pair-wise compatibilities differs, however.

3.2 Sensing modalities

Given a prior on the relative position between a and b (as computed by the Dijkstra projection), and sensor observations from pose a and b, the task is now to compute a rigid-body transformation that is likely to correctly align a and b. The techniques for doing this are sensor-dependent; we have implemented both a vision-based method and a laser-based method. The details of these methods are not necessary to understand the place recognition algorithm we propose. However, since hypothesis generation is a critical *input* to our approach, their properties are relevant to the results we present in Section 5. Consequently, we summarize our basic hypothesis generation approaches for the three sensor modalities discussed in this paper: cameras using SIFT, LIDARs, and a benchmark camera dataset using fiducials.

It is important to note that our proposed place recognition algorithm was used in each problem domain, without any modality-specific parameter tuning or tweaking.

3.2.1 Vision-derived Local Landmark Matching

Our vision data set was collected during the 2007 DARPA Grand Challenge National Qualifying Event by Team MIT's vehicle, Talos [34]. Our vision data sets use SIFT features extracted from an array of calibrated cameras providing a nearly 360 degree field-of-view. Note that these cameras were essentially monocular: their fields of view did not significantly overlap.

In addition to cameras, Talos had a high-end inertial measurement unit (IMU)/GPS system giving it very good positioning capabilities. In our experiments, we evaluate the performance of our algorithm on both the high-quality IMU data as well as artificially degraded data that is representative of a more typical vehicle. In the latter case, the IMU position estimate serves as ground truth.

SIFT features are extracted from each camera frame and are matched against a set of active "tracks". A track is comprised of a SIFT feature descriptor and a list of the observations of that feature (the position of the car and the bearing to the feature).

Periodically (at fixed spatial intervals with respect to the vehicle's motion), a new pose node is added to the graph. At this time, the tracks for each tracked SIFT feature are tested in order to determine whether the vehicle-relative position of that feature can be reliably determined. Since each observation of a feature consists of only a bearing (or equivalently, a ray) to the feature, the feature must be observed from several different positions in order to compute its position.

When the rays from different observations intersect in a sufficiently dense cluster (see Fig. 6), we can estimate of the location of the feature. These feature localizations are associated with the new node in the graph and are later used for loop closure. At this point, the track is reset so that the feature can be independently localized again and again.

The result of this process is a chain of robot poses, with each pose associated with a set of vehicle-relative (and statistically independent) feature localizations.

Figure 5: Overhead view of vision dataset area. The car made three counter-clockwise loops, with each loop approximately 610 m in length.

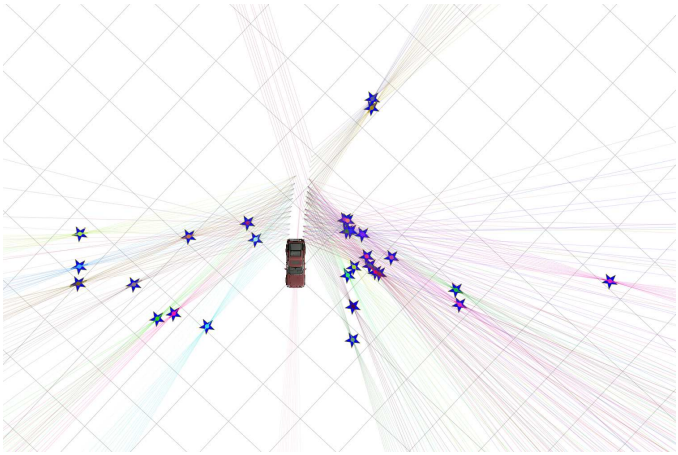

Figure 6: SIFT Landmark Initialization. Landmarks are computed by tracking SIFT features over short motions of the robot: when the intersection of the bearings converges near a single point, a landmark is initialized. In the figure, the vehicle is traveling downwards, and the stars represented localized features.

We now wish to find loop closures by matching the features observed from different poses. In order to find the rigid-body transformation between two nodes, we compute the matches between the SIFT landmarks of the two nodes. These SIFT correspondences are computed based on the SIFT descriptors. Given two sets of matching SIFT features, we compute a rigid-body transformation that best aligns those landmarks. We can score each rigid-body transformation by counting the total number of landmarks from node b that are projected near their counterpart in node a; using this metric, we find the best rigid-body transform using RANSAC [31].

The rigid-body transform output by RANSAC is a hypothesis relating nodes a and b. The quality of this hypothesis is strongly dependent on the reliability of matching SIFT features based on their descriptors. When the whole 128-dimensional SIFT descriptor is used, these SIFT matches are often correct, leading to good rigid-body transformations.

SIFT descriptors can be reduced in size, with the result that ambiguity is increased. By purposefully degrading SIFT features, we can measure the robustness of our algorithm to that ambiguity. Of the features we examined, SIFT features are unique in their ability to be easily decimated in order to allow an evaluation

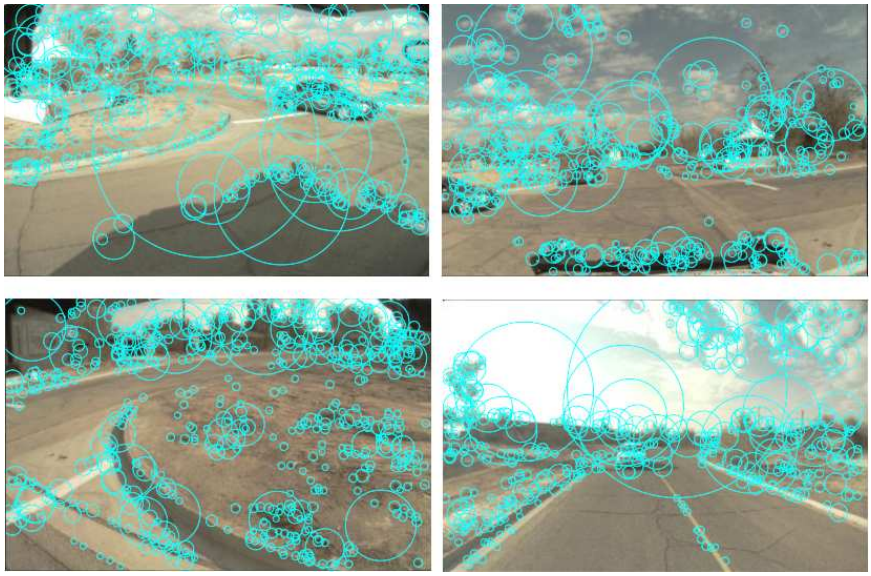

Figure 7: Extracted SIFT features. Clockwise: left camera, forward camera, right camera, rear camera. Cyan circles represent extracted SIFT features: the size of the circle encodes the scale of the feature.

of place recognition robustness versus feature ambiguity. Decimated SIFT features can thus serve as proxies for a variety of image based features that are more ambiguous than SIFT features, such as edges, lines, Harris corners [35], and FAST corners [36]. The matching performance of SIFT as the descriptor is reduced in length is shown in Figure 8.

SIFT features can be reasonably decimated by summing together contiguous spans of the feature vector. Each of the 128 individual features corresponds to bucket of a histogram. Summing these buckets is equivalent to having used a lower resolution histogram. In other words, this simple decimation procedure preserves the logic and principles of the SIFT feature— it simply uses coarser histogram buckets.

Another source of error results from correct but undesirable SIFT matches. For example, portions of the car are visible in some cameras: these produce SIFT observations that move with the car. While these obvious parts of the car could be masked out, other related effects are difficult to mask. For example, as the car drives around, it casts a shadow on the ground. This shadow generates many SIFT detections. Other moving cars can also cause undesirable SIFT matches. Since many of these sources of error are unavoidable, we did not attempt to mask any of them— instead, we let our proposed method detect and correct the resulting errors.

3.2.2 Laser Matching

We have processed a number of datasets with laser scanner data, including several common mapping benchmarks. These datasets were collected indoors using a single scanner with a 180-degree field-of-view. The complexity of these datasets varies significantly in terms of the noise of the range measurements, the rate at which observations are made, and the complexity and size of the environment.

Line features are extracted from individual point scans. Two correspondences between node a and b (two lines from each pose) are used to compute a rigid-body transformation. This rigid-body transformation is then scored by spatially correlating the points from b with a rasterized version of the points from a. The rigid-body transformation yielding the highest correlation is then subjected to a brute-force refinement step, in which the rigid-body transformation is numerically optimized [37].

The result is a rigid-body transformation T that causes the points from scan b to "line up" with the points from scan a. If the Mahalanobis distance of T is less than a threshold (typically 3) from the prior

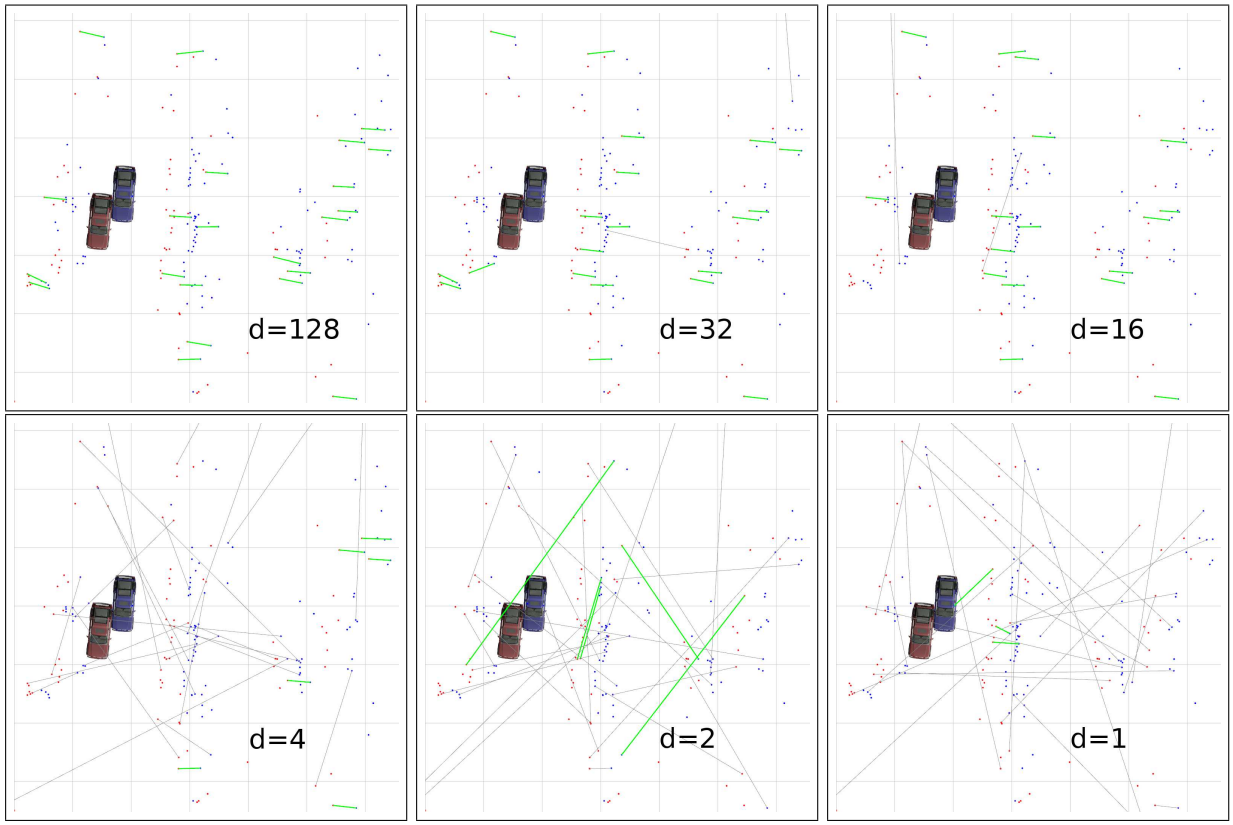

Figure 8: Matching decimated SIFT features. The SIFT features from two different visits to the same place are shown: dots correspond to features detected from the correspondingly-colored vehicle. Lines correspond to SIFT descriptor matches; thick green lines are those matches that are accepted by RANSAC. As the descriptor is decimated, the SIFT matches become less reliable. As the number of outliers increases, RANSAC eventually fails as well; in the figure, RANSAC fails to find the correct correspondences for d=2 and d=1. Our proposed loop closing method can effectively handle the increased error rate.

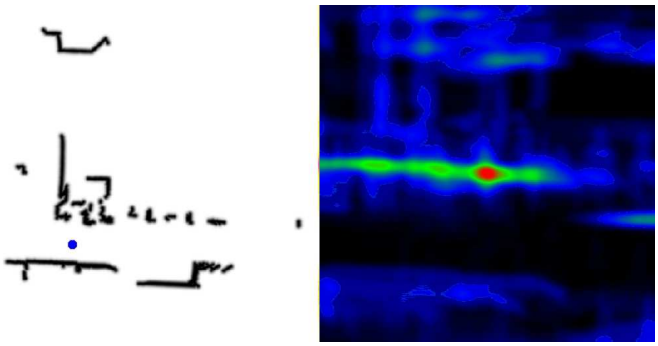

Figure 9: Scan Matching Cost Lookup Table. The cost function is precomputed into a 2D lookup table (left) allowing the probability of a particular alignment to be computed very quickly. The cost surface for an incoming scan, parameterized by the rigid-body transformation, can be very complicated (right). In particular, note the multiple local maxima. This cost surface is one slice of a 3-dimensional surface, corresponding to pure translations given the correct rotation.

derived from the Dijkstra projection, T is output as a hypothesis.

In indoor environments (where there is often repeated structure), the probability of an incorrect match increases with the uncertainty of the prior. These hypotheses can be incorrect when physically distinct environments are similar looking. Our place recognition algorithm, as we will demonstrate, can robustly identify these incorrect matches.

3.2.3 DLR Circles

The final sensor modality that we will use for our evaluation is the vision-derived DLR circles dataset, provided by Udo Frese for the RSS workshop on data association. In this dataset, sensor observations consist solely of sparse (and indistinguishable) point features (see Fig. 10). These point features are derived from camera observations of manually-distributed fiducial marks. No other data, aside from mediocre odometry, is available.

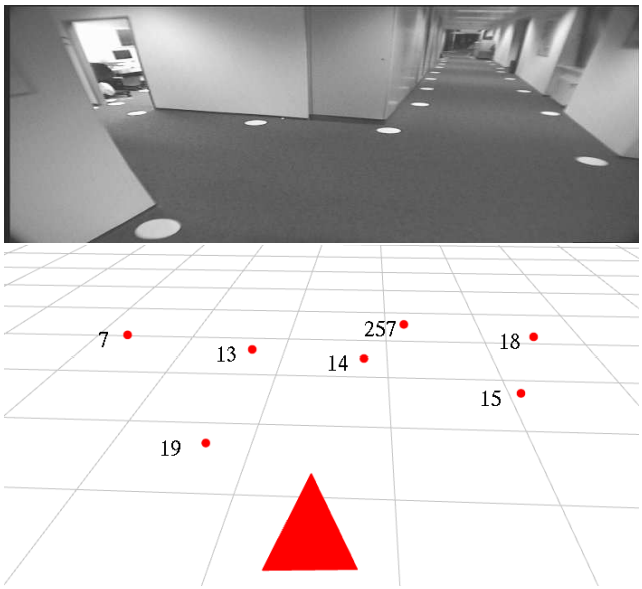

Figure 10: DLR "circles" example data. White circles were manually distributed in an office environment (top). These circles were automatically extracted using computer vision methods (bottom). The resulting dataset was annotated with the ground-truth identity of each detected circle [38]. Note that ground-truth *positions* were not available.

The challenge is to recognize places using a constellation of these features. This is challenging, due to low density of fiducials and their relatively uniform distribution (almost always along the walls of the hallway, almost always a a couple meters apart). The resulting data is highly ambiguous: from any two vantage points, it is almost always possible to find a set of correspondences that align three or more targets (and most of these, of course, are incorrect).

The pose-to-pose is generated via RANSAC: two points are randomly selected from both a and b, and using Horn's algorithm [39], a rigid-body transformation is computed that optimally aligns those points. For each point in a, we compute the distance to the closest point in b, and vice versa. Let the minimum distance for point a_i be $\text{dist}(a_i)$. We also incorporated a negative information penalty P, described below. The probabilistically motivated consensus score S is then:

$$S = \sum_{i \in a} e^{-\beta \text{dist}(a_i)^2} + \sum_{j \in b} e^{-\beta \text{dist}(b_j)^2} - P \tag{4}$$

The parameter β was set to 10.0 in our experiments. We also reject any rigid-body transformations that are more than three Mahalanobis distances away from the Dijkstra-projection prior. The rigid-body transformation achieving the highest consensus score S becomes a pose-to-pose match *hypothesis*, and is subject to further filtering as subsequent sections will describe. We call them hypotheses in order to emphasize the fact that they may be incorrect.

When there is environmental ambiguity (like a picket fence), the pose-to-pose matching will generate many different and incompatible solutions. This is due to the fact that each pose-to-pose match is independently computed by a greedy local optimization. The ambiguity of the environment is implicitly encoded in the dissonances of the pose-to-pose matches. Like the other modalities, these false positives will be filtered by our approach.

Notably, the DLR circles dataset provides ground-truth data associations, making it possible to quantitatively evaluate the reliability of our local matching system.

Negative Information in the DLR data set

The data in the circles dataset is highly ambiguous: individual features are indistinguishable and it is often possible to find erroneous rigid-body transformation that align two or more points reasonably well. However, it is rare to *fail* to detect a feature if one is present. Consequently, we estimate which landmarks should be visible and penalize alignments that result in missing features.

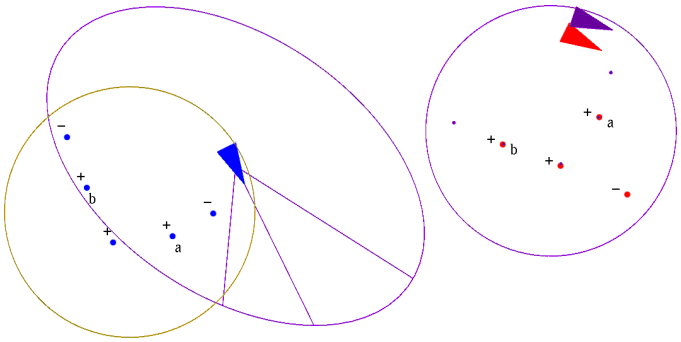

Figure 11: Pose-to-Pose matching. We use RANSAC to find a pair of points from each pose (labeled a and b) that best align the ensemble of points. The two large circles indicate the estimated sensor range. The ellipse indicates the positional uncertainty of the two poses. Points which are successfully matched are labeled with a plus sign; those that have no counterpart incur a penalty, and are labeled with a negative sign.

To do this, we must first estimate the size and shape of the overlapping views from the two poses. Of course, we would need to know the relative positions of the two poses in order to compute this exactly, and this is information is unknown.

The RANSAC method computes scores assuming that two pairs of features match. In other words, if a given RANSAC iteration is correct, the sensor range overlap must include at least the points upon which the alignment is conditioned. Thus, we model the overlapping sensor range as the smallest circle that includes both points. Each unmatched observation within these circles incurs a penalty of 1.0 (equivalent to the score resulting from a single *good* match.) In our experiments, the use of negative information significantly improved hypothesis generation.

4 Robust Place Recognition

Supposing that our low-level sensor processing system has identified a possible match (using one of the methods described in the previous section, for example), the critical question is whether the the putative match actually corresponds to a correct match. This section represents the central contribution of this paper.

The individual place recognition hypotheses generated in the previous section can have fairly high error rates. This section describes the process through which correct place recognition hypotheses are identified from high error-rate hypotheses. This process involves grouping related hypotheses together, identifying outliers, measuring the degree of local ambiguity (and thus addressing the "picket fence" problem), and finally, performing a global geometric sufficiency test (i.e., determining whether there is evidence for a place recognition in comparison to the robot's positional uncertainty). We will describe each of these components in turn.

4.1 Grouping

Our approach is to combine these into topologically-related place recognitions (i.e., hypotheses that relate similar portions of the robot's trajectory are grouped together). This grouping serves two several purposes: first, it provides a mean of performing outlier rejection, since the groups contain some redundant information. Second, whereas single hypotheses represent local matches consisting of sensor information available from only individual poses, groups of hypotheses allow spatially larger matches to be constructed. Matching larger areas is important when the positional uncertainty is large, as we will see.

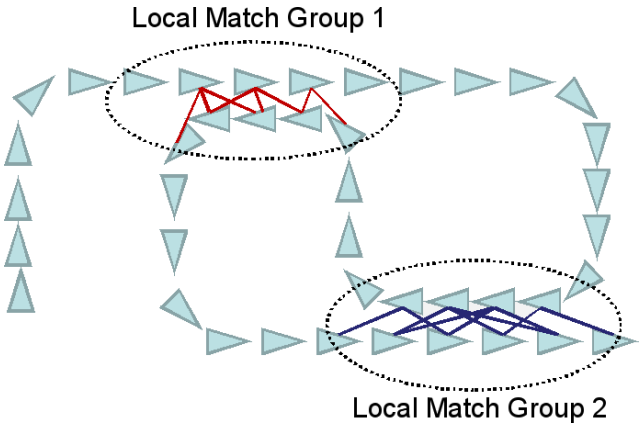

Figure 12: Grouping. Related hypotheses are grouped together into hypothesis sets. This makes it possible to perform outlier rejection. It also increases the effective size of matches, since they are comprised of multiple hypotheses.

In general, the error (and uncertainty) between two poses is related to the length of the trajectory between them: longer trajectories allow for greater accumulation of error. With this in mind, we can construct hypothesis sets so that the Dijkstra links will be derived from relatively short trajectory segments.

Each hypothesis relates a pair of nodes, which we denote as (a, b). Our strategy is to restrict hypothesis sets to contain hypotheses of the form $(a+k_i, b+k_j)$ where each k is a relatively small number ($|k| \leq k_{max}$). Given two hypotheses $(a+k_1, b+k_2)$ and $(a+k_3, b+k_4)$, we can construct a loop using two Dijkstra links, $(a+k_1, a+k_3)$ and $(b+k_2, b+k_4)$. This strategy keeps the Dijkstra links short (at most $2k_{max}$ poses long), allowing us to limit the amount of error they introduce (see Fig. 12)..

The value of parameter k_{max} is based on the local-navigation accuracy of the robot. In our systems, we set k_{max} to 8 poses, which typically translates to about 8 m of trajectory for our indoor robots, or about 16 m of trajectory for our full-sized car. Larger values of k_{max} increase the number of opportunities to generate hypotheses for any given hypothesis set: this is important since we require a set to have a minimum number of hypotheses before it is processed (we used a minimum size of 4 in our experiments).

Large values of k_{max} can lead to inaccurate Dijkstra links. In general, this causes pair-wise consistency values to be smaller than ideal, not larger. (It is more likely that an error will result in a bad loop than a good loop.) Our method is robust to these types of errors: the algorithm continues to identify sets of

hypotheses that are correct, but there is an increase in the number of falsely rejected hypotheses. In general, the performance of our method is good over a broad range of k_{max} values.

4.2 Pairwise Hypothesis Consistency

We now have a group of related hypotheses, similar to that shown Fig. 13. The next step in our place recognition algorithm is to compute the pairwise consistency between each hypothesis in the group.

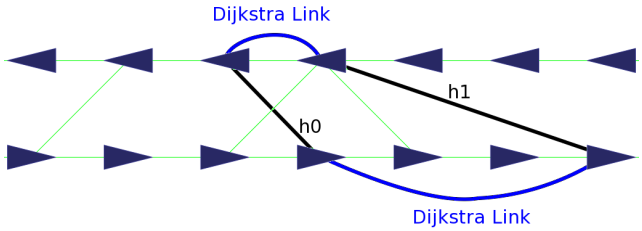

Figure 13: Pair-wise hypothesis test. From a group of five similar hypotheses (lines connecting poses between the top and bottom trajectory fragments), we select two hypotheses (h_0 and h_1). We form a loop of rigid-body constraints by combining the two hypotheses with two rigid-body transformations derived from the Dijkstra projection estimates. If the hypotheses are consistent, the composition of the four rigid-body transformations should be close to the identity transform.

For each pair of hypotheses, we can construct a loop of rigid-body constraints (see Fig. 13). This loop incorporates two additional rigid-body constraints derived from dead-reckoning. Since the constraints form a loop, the composition of their rigid-body transforms should be the identity matrix. Each rigid-body transformation is associated with a covariance matrix, allowing us to compute the probability that the loop is the identity matrix: high probabilities indicate pair-wise consistency of the two hypotheses.

When the rigid-body transformations of the two hypotheses (call them i and j) and two Dijkstra links are composed, we obtain a new rigid-body transformation T. Since each of these four links is a random variables, their composition T will also be a random variable.

For clarity, let us consider the details of this composition operation. Consider two rigid body transformations A and B, each of which is parameterized by three random variables x, y, and θ. The matrix corresponding to rigid-body transformation A is then:

$$A = \begin{bmatrix} \cos(\theta_A) & -\sin(\theta_A) & x_A \\ \sin(\theta_A) & \cos(\theta_A) & y_A \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The composition of rigid-body transformations A and B can be computed by computing the matrix product $C = AB$ and converting back into parameterized form. This can be done directly, without the matrix multiplication:

$$\begin{aligned} x_C &= \cos(\theta_A)x_B - \sin(\theta_A)y_B + x_A \\ y_C &= \sin(\theta_A)x_B + \cos(\theta_A)y_B + y_A \\ \theta_C &= \theta_A + \theta_B \end{aligned} \tag{6}$$

We can compute the covariance of random variable C by projecting the covariances of A and B through Eqn. 6. Specifically, let J_A be the Jacobian of the parameters C with respect to A according to Eqn. 6. We similarly define J_B. The covariance of C is then given by:

$$\Sigma_C = J_A \Sigma_A J_A^T + J_B \Sigma_B J_B^T \tag{7}$$

Composing the rigid-body transformations around the loop simply repeats this process three times, yielding the rigid-body transform T with uncertainty Σ_T. Finally, we can compute the pairwise consistency $A_{i,j}$ for hypotheses i and j:

$$A_{i,j} = e^{T\Sigma_T^{-1}T^T} \tag{8}$$

This quantity is proportional to the probability density that the rigid-body transformation is the identity matrix (i.e., $T = [0\ 0\ 0]$). In other words, Eqn. 8 is a measure of how likely it is that the four links actually form a loop. If either hypothesis i or j is incorrect, the composition of the four links is will not be a loop. Alternatively, if *both* i and j are incorrect, it is possible that their errors will "cancel out", but this is highly improbable. In short, the pairwise consistency $A_{i,j}$ is generally small if one or both of i and j are incorrect, though we cannot tell at this point which hypothesis is wrong.

4.3 Local Uniqueness and Outlier Rejection

Given a set of N hypotheses, we compute the pair-wise consistency for each pair, yielding an $N \times N$ "consistency" matrix A. We now wish to find the subset of hypotheses that is maximally self-consistent. In other words, we wish to find a subset of hypotheses whose pair-wise consistency is, on average, the greatest.

A hypothesis set can be viewed as a graph, with each hypothesis represented as a node. The (weighted) adjacency matrix of this graph is the pair-wise consistency matrix A.

Our solution is based on our previous work, an algorithm called "Single Cluster Graph Partitioning" (SCGP) [24, 40, 41]. SCGP is a graph partitioning method that attempts to find a single cluster of well-connected nodes, rejecting other nodes.

Let v be an $N \times 1$ *indicator vector* such that $v_i = 1$ if the i^{th} hypothesis should be accepted, and $v_i = 0$ otherwise. The sum of the compatibilities between the accepted hypotheses is $v^T A v$, and the number of accepted hypotheses is $v^T v$. Thus, the average pair-wise consistency $\lambda(v)$ for a subset of hypotheses v is simply:

$$\lambda(v) = \frac{v^T A v}{v^T v} \tag{9}$$

As an intuition-building exercise, suppose we have a set of accepted hypotheses v_0, and we want to decide whether to add another hypothesis to it. Adding a new hypothesis will add N pair-wise consistencies (of differing weights) to the numerator, but will also increase the denominator by 1. If adding those pair-wise consistencies increases the average amount of pair-wise consistency per node, then we should add that hypothesis. Note that our actual algorithm, unlike this exercise, does not greedily select hypotheses one by one.

Now, our task is to find an indicator vector v that maximizes $\lambda(v)$. Unfortunately, this problem has an exponentially-large search space. In the tradition of spectral graph-cutting methods like MinMaxCuts [26] and NormalizedCuts [25], we relax v to allow continuous values. We will find an optimal answer for continuous-valued v, and then later discretize v back into a proper indicator vector.

With v now a continuous-valued vector, we can look for extrema of $\lambda(v)$ by differentiating $\lambda(v)$ with respect to v and setting the result to zero:

$$\frac{d\lambda(v)}{dv} = \frac{Avv^Tv - v^TAvv}{(v^Tv)^2} = \frac{Av - \lambda(v)v}{v^Tv} = 0 \tag{10}$$

Note that we have exploited the fact that A is symmetric. Rearranging terms, we get:

$$Av = \lambda(v)v \tag{11}$$

By inspection, we see that the value of v that maximizes $\lambda(v)$ is the dominant eigenvector of A (and $\lambda(v)$ will be the dominant eigenvalue).

We note that the densest subgraph can be found in polynomial time by performing a number of max-flow computations on the hypothesis graph [42]. Our spectral approach is faster and (perhaps more importantly) leads to a confidence metric, as described below.

4.3.1 The Eigenvectors of the Consistency Matrix

As we have shown, the dominant eigenvector of the consistency matrix A yields the optimal continuous-valued indicator vector v. The second-most dominant eigenvector of A, however, is also useful.

Recall that A is symmetric and that the eigenvectors of a symmetric matrix are orthogonal [43]. Intuitively, this means that the hypothesis sets described by the first and second eigenvalues represent *different explanations of the data*. Thus, the ratio of the first and second eigenvalues conveys the relative confidence of the solution.

In algorithms like JCBB or RANSAC, it is also possible to keep track of the second-best answer, but the second-best answer is almost invariably a trivial variation on the best answer. SCGP's ability to evaluate the next-best *orthogonal* solution is unique, and extremely useful.

The first two dominant eigenvectors e_1 and e_2 of a symmetric matrix A can be efficiently computed using the Power Method as follows:

Algorithm 2 Power Method for extracting the two dominant Eigenvectors and Eigenvalues of a symmetric matrix A. A total of K power iterations are performed, with $K = 5$ being a reasonable value.

```
for i = 1; i <= 2; i = i + 1 do
  e_i = a random N × 1 positive vector, ||e_i|| = 1
  for iters = 0; iters < K; iters = iters + 1 do
    for j = 1; j < i; j = j + 1 do
      α = e_j · e_i
      e_i = e_i − α e_j
    e_i = e_i / ||e_i||
    e_i = A e_i
    λ_i = ||e_i||
  e_i = e_i / ||e_i||
```

This algorithm performs five iterations of the power method per eigenvector, which is adequate in virtually all cases. (When the eigenvalues are closely spaced and convergence is slow, the confidence test based on the ratio of the eigenvalues would reject the eigenvectors anyway.) The inner loop in j removes any component of earlier eigenvectors from the current eigenvector.

The confidence of the hypothesis set can be measured by computing the ratio of the largest two eigenvalues, λ_1/λ_2. When this ratio is less than a threshold (we used 2 in our experiments), it is not safe to use any of the hypotheses as a place recognition.

While it would be reasonable to discard these hypotheses completely, a significant amount of computational effort has been invested into collecting them. Instead of discarding the hypothesis sets immediately, we allow the sets to grow further. These additional hypotheses can resolve the ambiguity, allowing the correct hypotheses to be accepted.

4.3.2 Discretization of the Indicator Vector

In an earlier step, we allowed v, an indicator vector nominally consisting of only zeros and ones, to take on arbitrary continuous values. We now need to discretize v back into an indicator vector. We will denote the new discretized version of v as w. Given a threshold t, we write w as:

$$w_i(t) = \begin{cases} 1 & \text{if } v_i \geq t \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

There are at least two reasonable ways of selecting t: picking t such that it maximizes the dot product $\hat{w} \cdot v$, or so that it maximizes Eqn. 9. Both work well, often producing the same answers. In our experiments, we have used the dot-product method.

The dot-product method can be implemented in $O(N \log(N))$ time. First, note that there are at most N different w vectors, since there are at most N distinct values in v. If the values of v are sorted, the dot

product $w(t) \cdot v$ for increasing values of t can be incrementally computed in $O(1)$ time. Thus the total time complexity is $O(N + N\log(N)) = O(N\log(N))$.

We note that the discrete indicator vector w is not necessarily the globally optimal indicator vector. Instead, it is a discretized version of the optimal continuous indicator vector. While v is an approximation, its performance is very good.

Once the discrete-valued indicator vector has been computed, we reject hypotheses with $v_i = 0$ to the graph, since they are not part of the maximally-consistent subset. This step is very effective in eliminating outliers.

4.4 Global Sufficiency

Suppose that a robot has constructed two local maps of its environment (A and B) at different points along its trajectory, and suppose that these two maps are locally consistent (i.e., they match). We wish to determine whether A and B represent the same location, or whether they are two physically distinct (but similar-looking) places (see Fig. 14).

We begin by assuming that we have some prior knowledge of the relative position of the two environments, as would be provided by a simultaneous mapping and localization (SLAM) algorithm. This allows us to compute a bounding ellipse relative to area B in which area A *must* be found. Naturally, if B is not contained within this ellipse, B cannot be the same location as A.

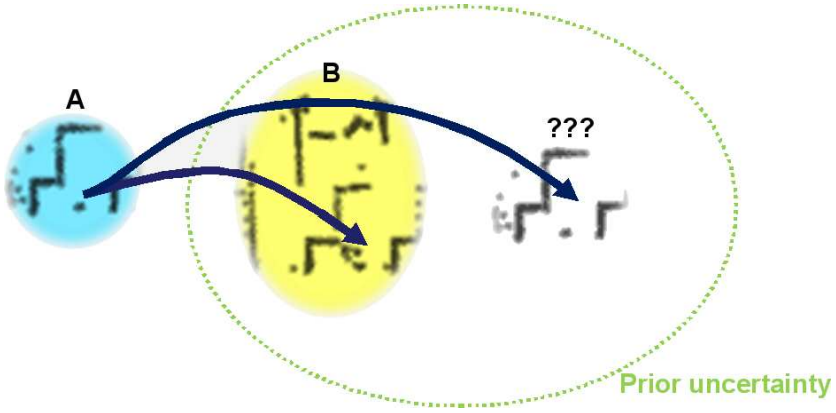

Figure 14: Global Sufficiency. A local match must be large in comparison to the robot's positional uncertainty. Otherwise, it may be possible that the robot is somewhere else entirely within its uncertainty ellipse.

The more interesting case is when local map B *is* within the ellipse that contains local map A. The main idea of this paper is that a local match is globally consistent if two conditions are satisfied. The first condition requires that the local match cover a spatial area that is comparable in size to the uncertainty ellipse. If the uncertainty ellipse is large enough to contain two or more identical-looking regions that might match area A, then we cannot be sure that area A is the same as area B. Conversely, if the uncertainty ellipse is only large enough to hold one "copy" of the locally matched region, then A and B must be the same location, since two distinct regions of that size could not both exist within the uncertainty ellipse.

5 Results

We have evaluated our place recognition on a variety of modalities (see Section 3.2 for some of the implementation details. These modalities include SIFT-based vision, LIDAR in indoor environments (both real and simulated), and the DLR circles dataset.

Loop closure algorithms are difficult to evaluate quantitatively: there are no standardized datasets for this purpose. The benchmark datasets that we have processed are composed of raw sensor data, but because sensor processing methods vary between researchers, the resulting sets of hypotheses are not the same. Naturally, the quality and reliability of the sensor processing (and hypothesis-generating) systems has huge impact on the difficulty of the loop-closing problem.

We address this problem in two ways. First, we evaluate the performance of our system repeatedly on the same dataset, with the quality of hypotheses purposefully degraded to varying degrees. This allows us to determine the point (if any) at which our algorithm can no longer produce useful output. Second, we make use of a unique data association dataset that has been carefully annotated with ground truth.

5.1 "Area C" Dataset

During the DARPA Urban Challenge, "Area C" was a testing area in which the ability of autonomous cars to queue at intersections (waiting for other cars) was tested. Team MIT's vehicle logged a great deal of sensor data, including vision data, during its test. This logged data now serves as a vision-based mapping test[1]

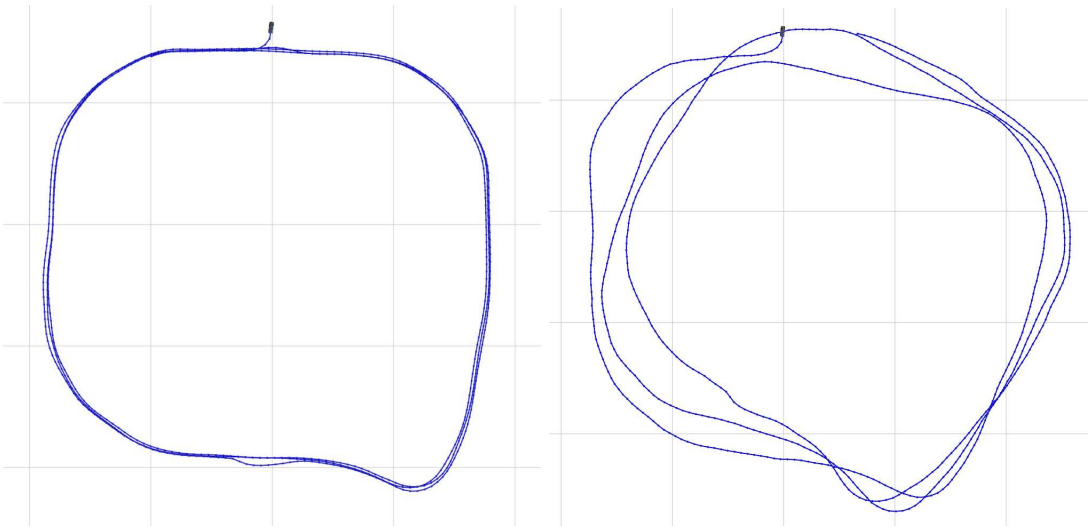

Figure 15: Area C dead-reckoning performance. Left: the dead-reckoning estimate, output by the high-end IMU, has an SSE of 1.02 versus the GPS-augmented ground truth. To simulate a more typical vehicle, additional noise was added (right): the initial configuration has an SSE of 487.

We used SIFT features to initialize local landmarks. These local landmarks were then associated with each other in order to compute a rigid-body transformation relating two poses. This rigid-body transformation served as a hypothesis.

SIFT descriptors are 128-dimensional vectors. At their full size, they provide a fairly robust means of matching landmarks. However, hundreds of SIFT features can be detected in a single camera frame, making the storage of these descriptors problematic. Data structures to perform matching on these features also becomes more complicated. Other authors [2, 44] have described ways of reducing the dimensionality of SIFT descriptors, but the risk is that the discriminative power of the descriptor might be reduced. When designing new descriptors, it is important to know how much discriminative capability the descriptors must have in order to be able to close loops.

[1]The method described here was not used by MIT during the DARPA Urban Challenge itself; the data here has been post-processed.

We answer this question by reducing the dimensionality of SIFT vectors in a simple way; this causes an increase in the hypothesis error rate. We then measure the ability of our approach to identify correct loop closures and reject incorrect hypotheses.

We reduce the dimensionality of SIFT vectors by dividing the descriptor into equally sized blocks and summing the elements in each block to form a shorter descriptor. While simplistic, our purpose is not to explore how to optimally reduce the dimensionality of SIFT descriptors, but to explore what happens when descriptors become ambiguous. Our technique is a simple and easily repeatable means of doing this.

In Fig. 8, two sets of SIFT features observed at different times (but near the same place) are shown. The two sets of features are matched based on descriptor's nearest-neighbors; these correspondences are shown as lines. RANSAC is used to identify a set of "good" correspondences from which a hypothesis is generated. As the descriptor is decimated, the matching performance drops as expected, producing increasingly poor hypotheses.

Significantly, our algorithm is able to identify correct hypotheses even when generated hypotheses are of very low quality (due to severe SIFT feature decimation). In fact, our algorithm is still able to identify correct hypotheses even when those hypotheses are the result of scalar-valued SIFT descriptors.

As expected, our algorithm rejects a large fraction of the hypotheses when the SIFT descriptor is small (see Fig. 16); the acceptance rate rapidly increases with increasing SIFT descriptor size. It is not possible to measure our false negative rate (i.e., the rate at which we are rejecting *good* hypotheses) since providing ground truth for the data associations would be prohibitively difficult. (Our false negative rate *is* reported for the DLR dataset, which is described in Section 5.6). We do know, however, the posterior maps that result from the filtered hypotheses are of high-quality (see Fig. 17), as measured by their agreement with the ground-truthed trajectory.

The posterior map resulting from our vision-based approach, based on degraded odometry, was *better* (SSE=0.87) than the map resulting from the map derived from the expensive IMU alone (SSE=1.02). If we combine the high-quality IMU data with our vision method, the SSE drops to 0.07— even if using SIFT descriptors of size 3.

SIFT feature extraction and tracking were implemented using relatively slow methods and ran at about 10% of real time. Other authors have described ways of accelerating these steps [45]. Loop closing and the algorithms described in this paper, in contrast, took between 7-9 s (depending on the size of the SIFT feature descriptors)— about 62 times faster than real-time.

5.2 CSAIL Dataset

We collected the CSAIL dataset in the Computer Science and Artificial Intelligence Laboratory's 7th floor (see Fig. 18). Its relatively short length coupled with a significant amount of clutter and fairly long loops make it an interesting test case. Like all of the datasets that follow, its primary sensor is a 180-degree field-of-view laser scanner. The robot trajectory was estimated via incremental scan matching, and loop closure hypotheses generated using a combination of local feature matching and scan matching refinement.

We will use this dataset to examine an actual hypothesis set in detail. The hypothesis set occurred in the area shown by Fig. 19. This hypothesis set contained 46 hypotheses, 34 of which were accepted as correct. Six representative samples of both inliers and outliers are shown in Fig. 20. The outliers (on top) are subtly misaligned; in contrast, the inliers (bottom) are well-aligned.

The pair-wise consistency matrix of those 46 hypotheses is shown as a graph in Fig. 21. In this figure, each node represents a hypothesis (e.g., one of the panels from Fig. 20): the *length* of a line represents the consistency between two hypotheses, with shorter lines indicating greater consistency. The cluster of inliers is readily discernible on inspection, and indeed, SCGP identifies it automatically.

A total of 5.3 s of CPU time was required to automatically generate loop-closure hypotheses and filter them using the mechanisms in this paper. The graph had 201 poses, and a total of 1057 hypotheses were generated. Of these, 710 were accepted. (184 were rejected because they were members of hypothesis sets that had fewer than four hypotheses; 63 were rejected because they were members of a hypothesis set whose first two eigenvalues were too similar; and 100 were explicitly classified as outliers.)

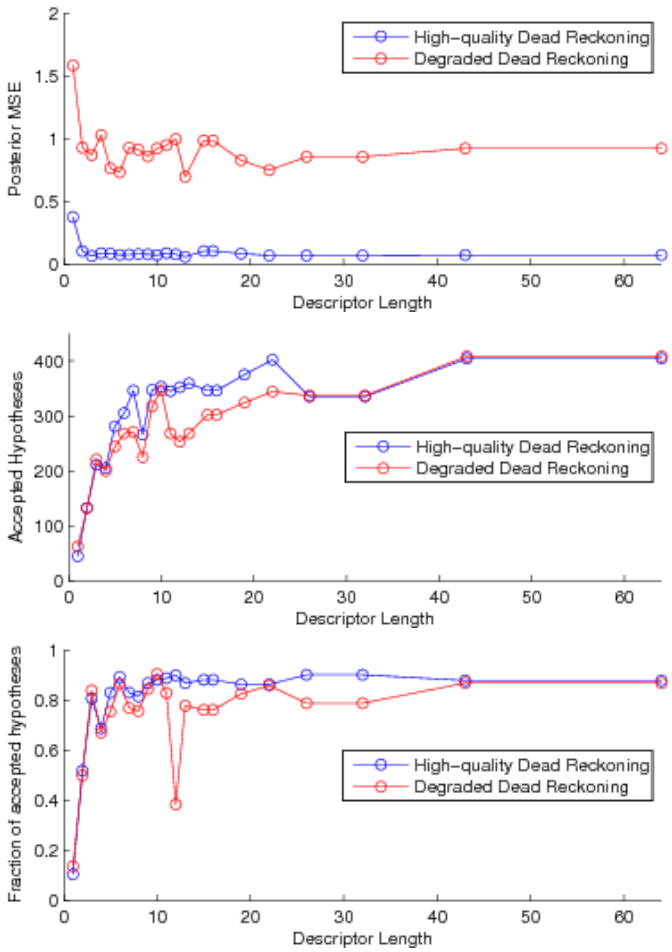

Figure 16: Area C Robustness. Shorter SIFT descriptors present greater perceptual ambiguity, and are more likely to produce incorrect matches. However, our method produces good maps in each case, and most surprisingly, can do almost as well with 3-dimensional vectors as with 128-dimensional vectors. The downward blip at $x = 12$ (bottom) was due to an usually large number of ambiguous hypothesis sets, in which the ratio of the first to second eigenvalues was less than the threshold of 2.0.

5.3 Killian Court Dataset

The Killian Court dataset, first collected and processed by Bosse [9], is challenging due to several long loops (see Fig. 23). Significant amounts of error can accumulate while the robot is traversing a long loop, which makes closing the loop harder. Another interesting characteristic of the Killian dataset is that it was recorded with a laser scanner with relatively high noise. This sensor noise makes it more difficult to recognize areas based on fine details. As a result, the hypothesis generator produced a greater number of incorrect hypotheses. The dataset also suffers from particularly poor wheel-derived odometry, as seen in Fig. 23.

As a partial compensation for this greater noise, loop closing was performed on larger sets of laser scans; i.e., instead of matching a single pose to another single pose, a small group of poses was matched at a time. This provided greater context for the hypothesis generator and reduced the rate of false positives.

In total, the Killian dataset consists of 1462 poses. These generated 6091 hypotheses, of which 4142 were accepted. 1187 were rejected because they were members of small hypothesis sets, 342 were rejected due to ambiguous eigenvalues, and 420 were classified as outliers. A total of 82 s of CPU time was required to

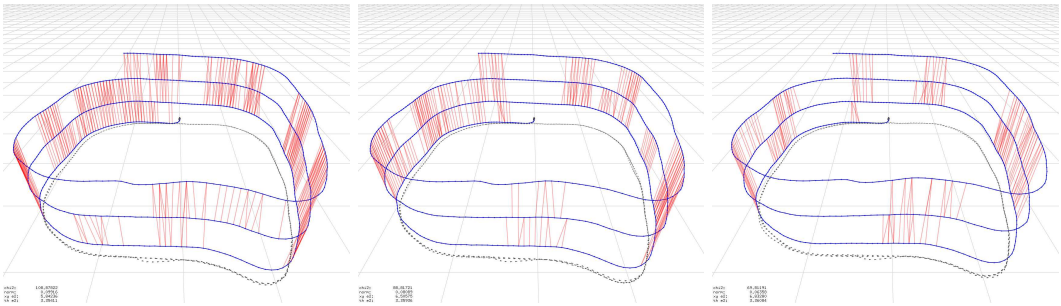

Figure 17: Area C Pose Graphs versus Descriptor Length. The pose/feature graph for the Area C dataset using 128-dimensional descriptors, 8-dimensional descriptors, and 1-dimensional descriptors. As the size of the descriptor goes down, ambiguity increases, resulting in fewer loop closures (red lines).

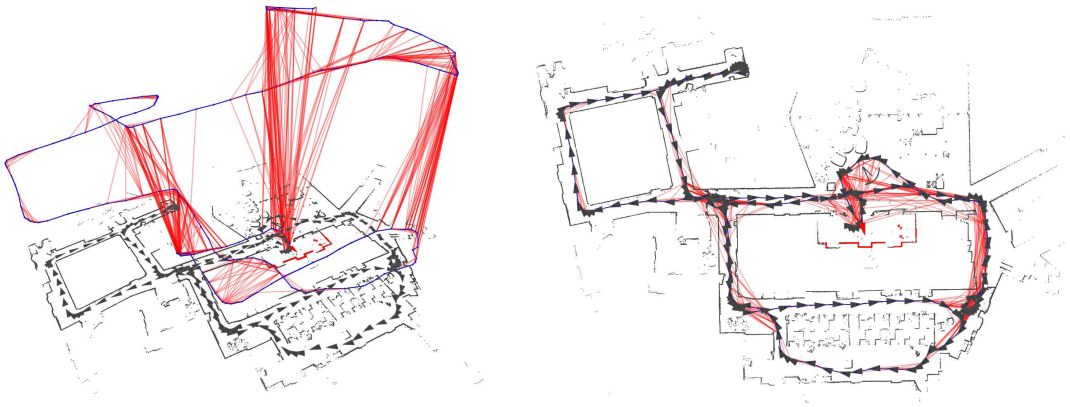

Figure 18: CSAIL Dataset Overview. Left: the pose graph is shown, with the trajectory in blue and loop closures in red. Right: the same pose graph, shown from above.

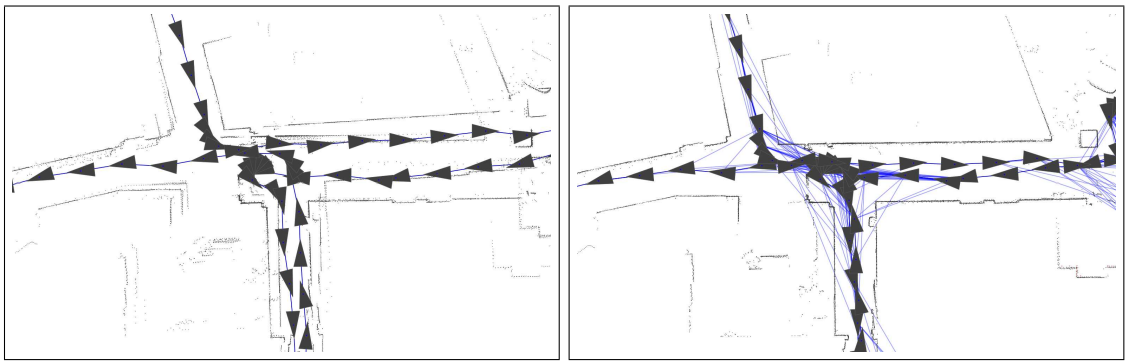

Figure 19: CSAIL loop closure close-up. Before loop-closing (left), laser scans show significant discrepancies. After recognition (right), not only are the discrepancies resolved, but the topological relationships are discovered.

generate and filter the hypotheses. The dataset corresponds to 2 hours and 9 minutes of robot driving.

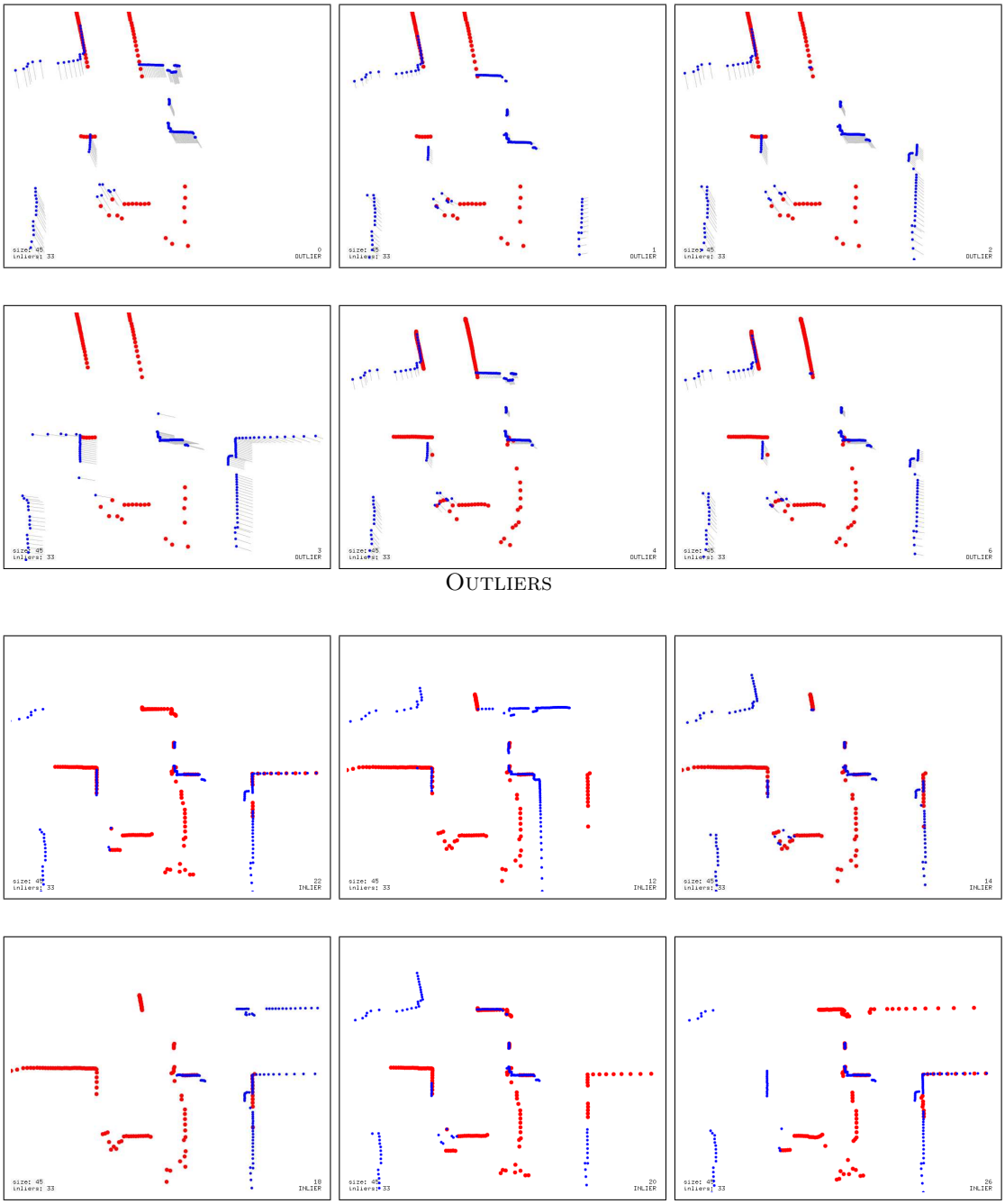

Figure 20: Filtered loop-closure hypotheses. Twelve representative hypotheses from a hypothesis set of size 45 are displayed. The outliers (top) and inliers (bottom) were automatically labeled using Single-Cluster Graph Partitioning. The alignment errors (derived from the posterior) are shown as gray lines: these errors are apparent in the outlier set.

5.4 Stanford Gates Building

Like many of the other datasets, the Stanford Gates dataset (a standard benchmark) was recorded with a single 180-degree field-of-view lidar. The dataset is reasonably large, and contains several often-visited areas.

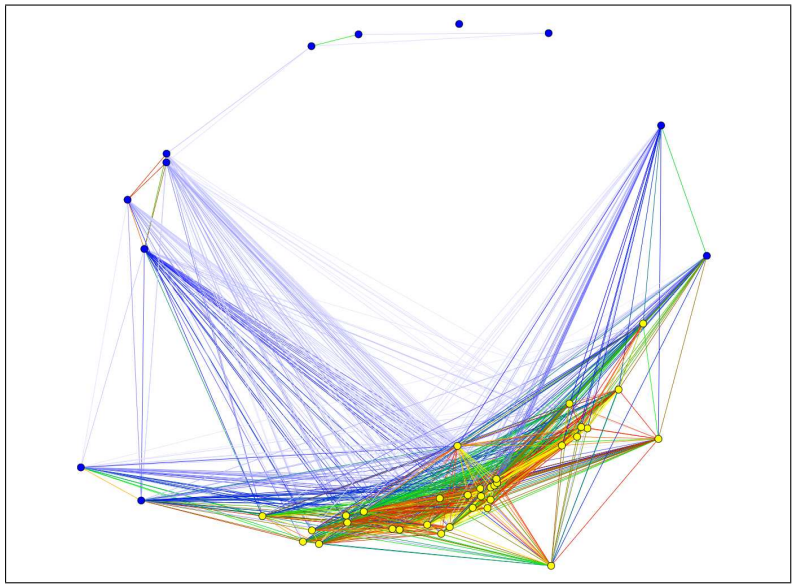

Figure 21: Adjacency graph for a set of 45 hypotheses. Each node represents a pose-to-pose hypothesis: the inlier set is indicated by yellow nodes (towards the bottom), with outliers in blue. Brightly-colored edges indicate greater pair-wise compatibility. The distance between two nodes is roughly proportional to the joint probability of the two hypotheses.

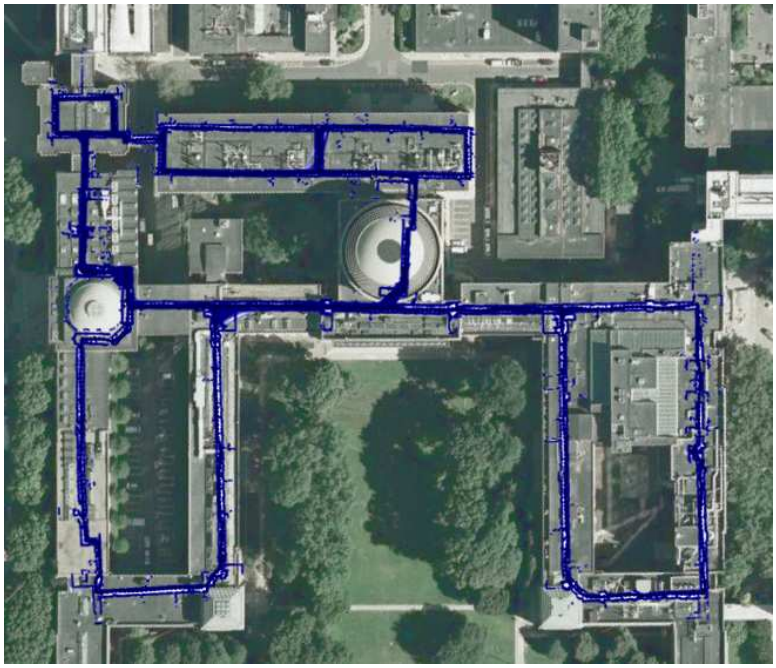

Figure 22: Killian Satellite Overlay. The long loops provide few opportunities for loop closing, leading to small distortions near the bottom of the figure. Macroscopically, however, the map aligns well with the satellite imagery.

These result in a large number of loop closures.

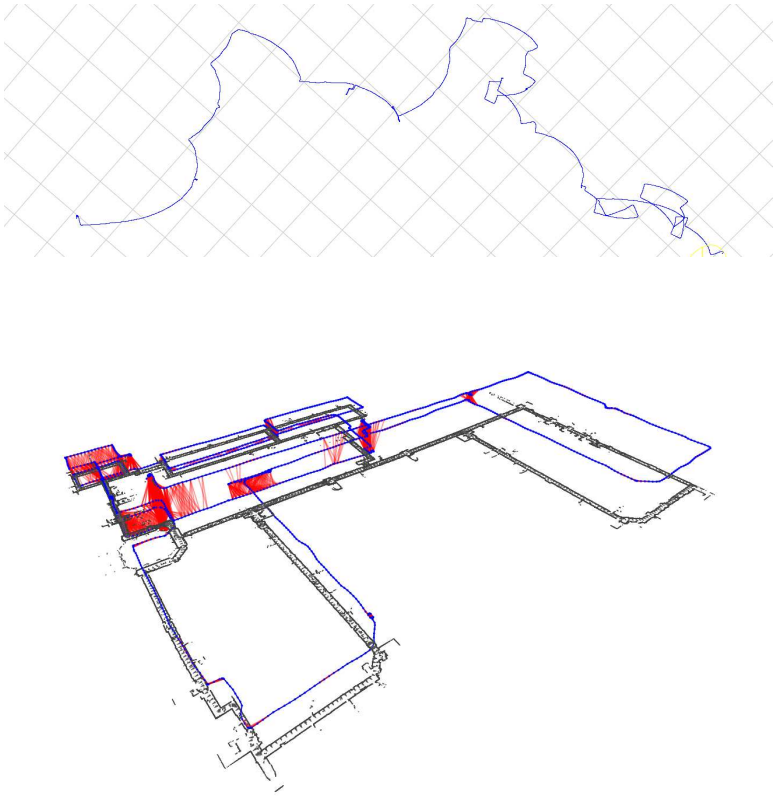

Figure 23: Killian Odometry and Pose Graph. Top: the raw wheel odometry (plotted on a 50 m grid). Bottom: The pose graph shows the relatively few areas in which loop closures are possible.

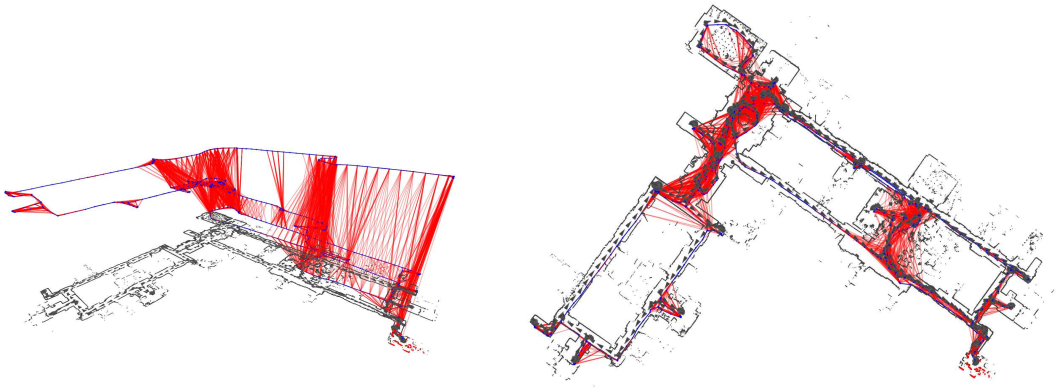

Figure 24: Stanford Gates Building. Left: the pose graph resulting from our algorithm. Right: the optimized map, viewed from above.

The Stanford dataset consists of 679 poses, which generated 7549 hypotheses. Of these, 6398 were accepted, with 470 rejected because their hypothesis set was too small, 253 rejected due to poor eigenvalue confidence, and 428 classified as outliers. Processing time was 64 s; the dataset represents 34.5 minutes of data.

5.5 Intel Research Center

The Intel Research Center is a common benchmark dataset characterized by a large number of loop closures. The dataset was provided by Dieter Fox and is available on Radish [46]. The robot travels several times around the entire lab space, and visits each room. Of the datasets we processed, this dataset has the most loop closures.

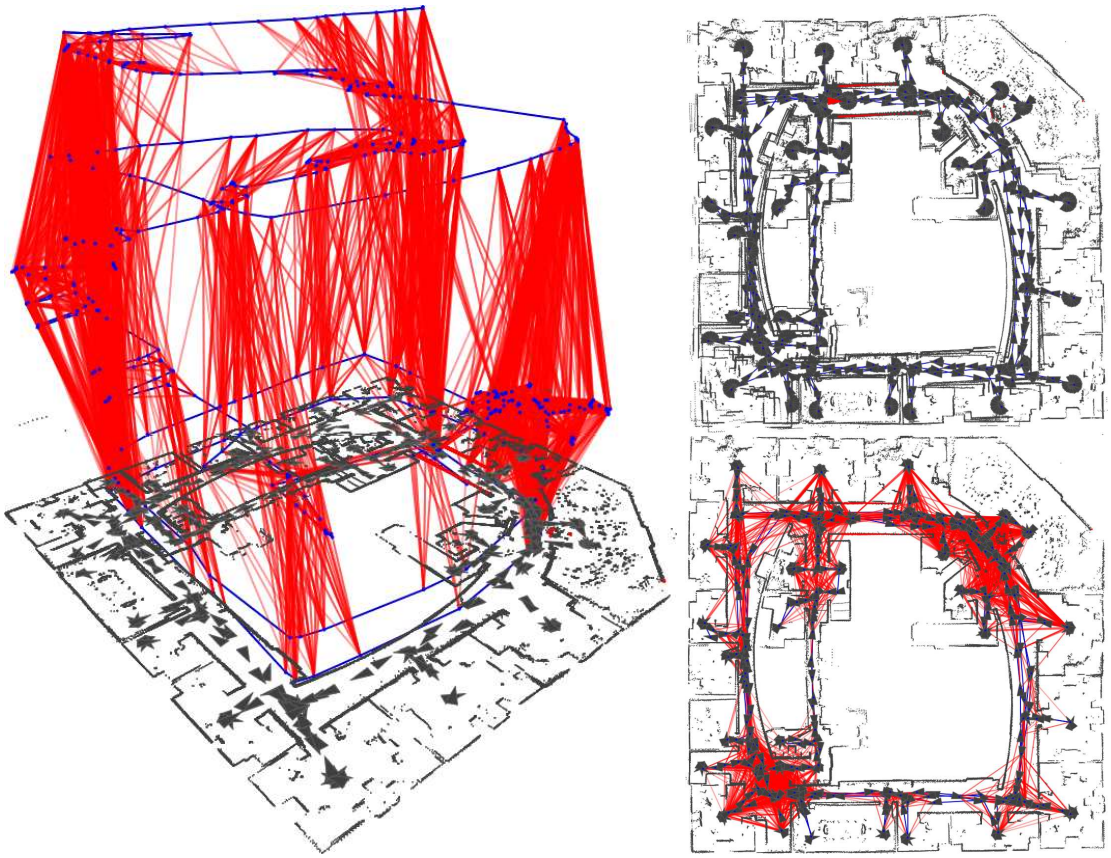

Figure 25: Intel research center. Left: the pose graph resulting from our algorithm. Top right: the uncorrected map. Bottom right: the optimized map, viewed from above.

The Intel dataset consists of 875 poses, spanning 44.9 minutes of robot driving. A whopping 15611 loop closure hypotheses were generated, and 12900 were accepted. Of those rejected, 758 were members of small sets, 715 were members of sets that failed the confidence test, and 1238 were classified as outliers. Total processing time was 180 s.

5.6 DLR Circles

We tested our algorithm on the DLR "circles" dataset, using it to identify loop closures. These loop closures classically represent difficult data association problems.

A total of 4644 pose-to-pose hypotheses belonging to 278 potential local matches were generated, 2043 of which were accepted. We instrumented our algorithm to remember which data associations were used to generate each hypothesis. Using ground truth data, we can determine whether each hypothesis (generated by RANSAC) was the result of "good" or "bad" data associations. The performance of our algorithm on a typical run is tabulated in Fig. 26. Notably, there were *no* false positives.

Hypothesis Outcome	Good Assoc.	Bad Assoc.
Accepted Hypotheses	2043	0
Rejected (small set)	714	32
Rejected (ambiguous)	1168	105
Rejected (inconsistent)	544	38

Figure 26: Hypothesis Error Rates. The "good" and "bad" columns denote hypotheses arising from correct (and incorrect) data associations, based on ground truth data. Most importantly, the algorithm exhibits no false positives. However, a large number of false positives are present due to the noise in the observations: this noise can result in poor hypotheses even when the data associations are correct. The numbers above reflect individual pose-to-pose matches; these were grouped into a total of 278 local matches.

Since our algorithm includes randomized components, the performance varies from run to run. False positive rates are consistently low (usually zero), though failures do occasionally occur. These appear to be the result of repeated iterations of RANSAC yielding similar (but incorrect) pose-to-pose matches. These incorrect matches can form a self-consistent local match. We believe these failures can be avoided by preventing multiple pose-to-pose matches from using the same pair of data associations. Even when a failures occurs, it generally has a minuscule effect on the quality of the posterior map. This is because the errors are rare, small in magnitude, and generally surrounded by dozens of correct matches that mitigate their effect.

Our false negative rate, based on Fig. 26 appears to be quite high. Indeed, a significant number of good hypotheses *are* rejected. However, the numbers are not as bad as they appear: observation noise can cause hypotheses to be poor even if the data association is correct. Since our pair-wise consistency test is based on the agreement of the rigid-body transformations (and *not* the data associations), these poor hypotheses are frequently rejected.

A total of 199 seconds of CPU time (on a 2.4GHz Intel processor) were required for feature matching (including RANSAC) and hypothesis filtering. Similarly good loop-closing results can be obtained even when the original dataset is decimated, but with much lower computational costs.

6 Conclusion

Recognizing places is a critical capability for many robot systems. Loop closing is a form of place recognition that is central to the task of map building: it prevents the unbounded growth of dead-reckoning error. However, it is a difficult task due to both perceptual ambiguities and the large potential number of data associations.

In this paper, we described an automatic loop closure system that can process a set of unreliable loop closure hypotheses and produce a set of correct hypotheses. It exploits the property that correct hypotheses generally agree with each other, whereas incorrect hypotheses tend to disagree with each other. The set of correct hypotheses is identified by examining the dominant eigenvectors of the pair-wise consistency matrix.

A critical capability of our method is the ability to determine the confidence of the solution. The second-best solution is not trivial variation on the best solution (as is the case in other algorithms), but is an orthogonal explanation of the data. The ratio of the eigenvalues of these two solutions serves as a useful confidence metric.

The method is also very fast, running much faster than real-time on all of the datasets we examined. It is also adaptable to a variety of sensing modalities: we demonstrated the system using local vision-based landmarks and laser-based scan matching.

The robustness of the system was demonstrated using a variety of sensor modalities. We first presented results using decimating SIFT features. Originally 128-dimensional vectors, these were purposefully degraded to 1-dimensional scalars. Even though feature matching performance was severely affected, our system was able to identify correct loop closure hypotheses. The resulting loop closures allowed better posterior maps

to be computed. We also presented results from standard LIDAR benchmarks. The DLR circles dataset is unique among our experiments in that it includes ground truth for data association; this allowed a definitive evaluation of the performance of our system.

A disadvantage of the local matching approach is that it requires that matches can be computed using only locally-available information. This in turn requires that observation noise be sufficiently low to generate decent matches and that it be possible to generate a rigid-body transformation given a pair of observations. When these conditions are satisfied, however, our local matching approach is fast, reliable, and fairly easy to implement.

Improving the reliability of place matching continues to be a central challenge in map building. The present work is applicable to many sensor modalities and, we believe, extends the reach of mapping systems. Much work remains, however: improving the quality of the underlying sensing systems (the source of the hypotheses that we process) is an active area of future work. Additional gains may be achievable in filtering as well: perhaps the false negative rate can be reduced, for example.

Approaches like FABMap [27], which exploiting highly distincitve landmarks, approach place recognition differently than our geometrical approach. As described here, our approach does not close loops when the size of the local match is small, even if highly distinctive landmarks are present. In contrast, FABMap does not close loops when large local matches exist, but are all of very low distinctiveness. An approach that combines both mechanisms in a cohesive and principled manner might exhibit better performance.

Data

Previously unpublished datasets used in this paper are available from the author, and/or from the author's website: http://april.eecs.umich.edu.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, Corfu, Greece, 1999, pp. 1150–1157. [Online]. Available: citeseer.ist.psu.edu/lowe99object.html

[2] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 02, pp. 506–513, 2004.

[3] A. Kleiner, J. Prediger, and B. Nebel, "RFID technology-based exploration and SLAM for search and rescue," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4054–4059, October 2006.

[4] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking.* New York, NY, USA: ACM, 2004, pp. 70–84.

[5] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, August 2000, pp. 32–34. [Online]. Available: citeseer.ist.psu.edu/priyantha00cricket.html

[6] H. Shatkay and L. Kaelbling, "Learning topological maps with weak local odometric information," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1997, pp. 920–929.

[7] J. Folkesson and H. I. Christensen, "Closing the loop with graphical SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 731–741, 2007.

[8] B. Kuipers and P. Beeson, "Bootstrap learning for place recognition," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Edmonton, Canada, 2002, pp. 174–180. [Online]. Available: citeseer.ist.psu.edu/kuipers02bootstrap.html

[9] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework," *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, December 2004.

[10] J. Neira and J. D. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, December 2001.

[11] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, August 2002.

[12] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun, "Towards lazy data association in SLAM," in *Proceedings of the International Symposium of Robotics Research (ISRR)*. Sienna, Italy: Springer, 2003.

[13] M. Montemerlo, "FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.

[14] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., April 2003.

[15] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 2428–2435.

[16] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *Int. J. Rob. Res.*, vol. 26, no. 4, pp. 335–359, 2007.

[17] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, San Francisco, CA, 2000, pp. 3841–3846.

[18] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 196–207, April 2005.

[19] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2262–2269.

[20] ——, "Spatially-adaptive learning rates for online incremental SLAM," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[21] E. Olson, "Robust and efficient robotic mapping," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[22] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

[23] G. Grisetti, D. L. Rizzini, C. Stachniss, E. Olson, and W. Burgard, "Online constraint network optimization for efficient maximum likelihood map learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[24] E. Olson, M. Walter, J. Leonard, and S. Teller, "Single cluster graph partitioning for robotics applications," in *Proceedings of Robotics Science and Systems*, 2005, pp. 265–272.

[25] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 888–905, August 2000.

[26] C. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A MinMaxCut spectral method for data clustering and graph partitioning," Lawrence Berkeley National Laboratory, Tech. Rep. 54111, 2003.

[27] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Rome, April 2007.

[28] K. L. Ho and P. Newman, "Detecting loop closure with scene sequences," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 261–286, 2007.

[29] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proceedings of the IEEE International Symposium on Computational Intelligence (CIRA)*, 2000.

[30] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, April 1997.

[31] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis andd automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[32] R. L. Rivest and C. E. Leiserson, *Introduction to Algorithms*. New York, NY, USA: McGraw-Hill, Inc., 1990.

[33] S. Thrun and M. Montemerlo, "The GraphSLAM algorithm with applications to large-scale mapping of urban structures," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–430, May-June 2006.

[34] J. Leonard, D. Barrett, J. How, S. Teller, and et al., "Team MIT DARPA urban challenge technical report," Massachusetts Institute of Technology, Tech. Rep., April 2007.

[35] K. G. Derpanis, "The harris corner detector," 2004.

[36] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 2008. [Online]. Available: http://lanl.arXiv.org/pdf/0810.2434

[37] E. Olson, "Real-time correlative scan matching," Kobe, Japan, June 2009.

[38] U. Frese, "Deutsches zentrum für luft- und raumfahrt (DLR) dataset," 2003.

[39] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America. A*, vol. 4, no. 4, pp. 629–642, Apr 1987.

[40] E. Olson, J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, October 2006.

[41] K. E. Bekris, M. Glick, and L. E. Kavraki, "Evaluation of algorithms for bearing-only SLAM." in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2006, pp. 1937–1943.

[42] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan, "A fast parametric maximum flow algorithm and applications," *SIAM J. Comput.*, vol. 18, no. 1, pp. 30–55, 1989.

[43] G. Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 1993.

[44] J. Sivic and A. Zisserman, "Video Google: Efficient visual search of videos," in *Toward Category-Level Object Recognition*, ser. LNCS, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds. Springer, 2006, vol. 4170, pp. 127–144. [Online]. Available: http://www.robots.ox.ac.uk/ vgg

[45] S. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc, "Feature tracking and matching in video using programmable graphics hardware," *Machine Vision and Applications*, November 2007. [Online]. Available: http://dx.doi.org/10.1007/s00138-007-0105-z

[46] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: http://radish.sourceforge.net/

Author Information

Edwin Olson received his B.S., MEng., and PhD in Electrical Engineering from MIT in 2000, 2001, and 2008, respectively. He is an Assistant Professor at the University of Michigan in Ann Arbor, where he conducts both applied and theoretical research into robot perception, navigation, path planning, and learning.