Michael Bosse
Robert Zlot
Autonomous Systems Laboratory,
CSIRO ICT Centre,
PO Box 883, Kenmore,
Queensland 4069,
Australia
{Michael.Bosse, Robert.Zlot}@csiro.au

Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM

Abstract

Reliable data association techniques for simultaneous localization and mapping (SLAM) are necessary for the generation of large-scale maps in unstructured outdoor environments. Data association techniques are required at two levels: the local level represents the inner loop of the mapping algorithm, and the global level where newly mapped areas are matched to previously mapped areas to detect repeated coverage and close loops. Local map building is achieved using a robust iterative scan matching technique incorporated into an extended Kalman filter where the state consists of the current pose and previous poses sampled periodically and at a fixed lag from the current time. The introduction of states at a fixed time lag significantly reduces the growth of errors in the location estimate and the resultant map. For global matching, we enhance existing histogram cross-correlation techniques, introducing entropy sequences of projection histograms and an exhaustive correlation approach for reliable matching in unstructured environments. This enables loop closure without depending on prior knowledge of map alignment. These data association techniques are incorporated into the Atlas SLAM framework, enabling the generation of accurate two-dimensional laser maps over tens of kilometers in challenging outdoor environments.

KEY WORDS—SLAM, data association, laser, scan matching, map matching, loop closure, ICP, histogram correlation.

1. Introduction

The problem of mapping large-scale environments is a fundamental problem in mobile robotics. Robots that operate in environments where no *a priori* map is available must often construct maps in order to function and adapt to their working environment, or to provide accurate models for an end user. Further exacerbating the mapping problem is the fact that existing methods of robot localization such as odometry or inertial sensors become increasingly inaccurate over time. Even the global positioning system (GPS) is not an option for indoor, subterranean, underwater, or extraplanetary applications, and cannot be relied upon outdoors (John A. Volpe National Transportation Systems Center 2001). This gives rise to the simultaneous localization and mapping (SLAM) problem in which a robot with no known map or accurate measure of its pose must estimate and maintain both. In this article, we address the online SLAM problem for large structured or unstructured environments using laser range data. We first describe our approach to constructing local maps from scans produced by one or more asynchronous lasers. The map building algorithm is based on a Kalman filter that maintains a collection of robot poses and estimates vehicle motion through an iterative laser scan matching technique. Odometry measurements are not required by our approach, though they may be incorporated into the filter if available. Second, we detail a method for matching maps to recognize regions that have been previously mapped. Using this data association technique, we are able to close large loops, thereby solving one of the fundamental challenges of the SLAM problem. The map building and map matching techniques have been integrated into the *Atlas* framework (Bosse et al. 2004), which has enabled us to construct maps of the order of several tens of kilometers in a variety of environments from unstructured rural roads to dynamic urban city streets. To our knowledge, these are the largest laser range-based SLAM maps created in terms of the distance covered. In addition, we believe these are the most complex SLAM maps to date with respect to the variety and authenticity of the operating environments. Our system processes large amounts of outdoor sensor data collected under real-world

The International Journal of Robotics Research
Vol. 27, No. 6, June 2008, pp. 667–691
DOI: 10.1177/0278364908091366

Figures 2–20 appear in color online: http://ijr.sagepub.com

driving conditions in the presence of unstructured and dynamic elements.

Mapping of large-scale environments introduces several key issues that must be addressed. Common to all SLAM algorithms, we assume there is no prior known map, and that odometry estimates (if available) are inaccurate. Environments may vary from structured, in which there are many flat surfaces—for example, within interiors of buildings or along downtown streets—to unstructured, in which there may be irregular arrangements of vegetation, rocks, or man-made structures. There may also be dynamic elements to contend with, even in structured environments; for example, other moving vehicles and people. When attempting to close loops, a solution should be robust to incomplete and noisy correspondences between associated map regions, and to the potential for large uncertainties when first revisiting an area. Additionally, the use of a two-dimensional laser range sensor introduces the possibility of modeling errors when the vehicle pitch or terrain slope changes quickly leading to ground strikes and other three-dimensional artifacts. Finally, in order to solve the SLAM problem online, the mapping algorithm should be efficient so as to run under real-time constraints.

1.1. Related Work

The data association method used in a SLAM approach can be heavily influenced by the choice of map representation. Map types can be categorized as *feature-based*, *location-based*, or *view-based*. Feature-based approaches (Smith and Cheeseman 1987; Ramos et al. 2007; Montemerlo et al. 2003; Williams and Mahon 2004) define a map as a list of parameterized objects in specific locations. A feature extraction algorithm must be used to estimate a set of features such as points, lines, or corners from the raw sensor data. Thus, feature-based approaches may be limited to environments for which the features are designed, such as structured indoor or urban environments. Common data association techniques for feature-based maps are: gated nearest-neighbor (Bar-Shalom and Fortmann 1988), which requires a good initial guess; signature string matching (Rencken et al. 1998; Bosse et al. 2004), which is sensitive to the choice of feature parameterization; and joint or batch correspondence methods (Neira and Tardós 2001; Neira et al. 2003; Bailey 2002), which are more robust to poor feature correspondences but are also more computationally expensive.

Location-based maps (Moravec 1988; Thrun et al. 2000) discretize the environment into a number of regions, each of which is labeled with a value (e.g., a grid with each cell marked as being free space or occupied). This provides more versatility since arbitrary shapes in the environment can be represented, and checking the state of a location is trivial. On the other hand, location-based maps are resolution-dependent, can be cumbersome to update, can have high memory requirements, and discard information by assigning a single value to each cell. Location-based data association involves matching a local submap to a global map (Moravec 1988). To avoid an exhaustive search, a hill-climbing algorithm can be used (Schultz and Adams 1998). However, this approach is susceptible to local maxima and its accuracy depends on the choices of map representation and search resolutions.

In view-based approaches (Eustice et al. 2006; Newman et al. 2006; Lu and Milios 1994; Gutmann and Konolige 1999), maps are composed of a collection of full sensor readings, including raw data and the sensor pose where the data was collected (e.g., a laser scan with the location and orientation of the sensor). View-based approaches can be considered a type of feature-based approach where entire sensor scans are the features; however, since there is no feature extraction algorithm required, no information is lost or misinterpreted in the process. At the same time, view-based maps can be even more expressive than location-based maps since they represent arbitrary shapes in the environment without aggregating all measurements in a given region into a single value. A disadvantage of view-based maps is that they can be memory and computationally intensive if all sensor data is saved and processed; thus subsampling or choosing representative scans, along with utilizing efficient data structures, are typically necessary to ensure tractability.

View-based representations are most often used for dense range-bearing data, such as laser scans, and the data association process is typically called *scan matching*. Several iterative techniques have been proposed for determining the alignment between two scans when a good *a priori* estimate is available (Lu and Milios 1994; Rusinkiewicz and Levoy 2001; Pfister et al. 2002; Biber and Straßer 2003; Tomono 2004; Diosi and Kleeman 2005). For instance, when the scans are taken sequentially by a moving vehicle, approximate alignments between successive scans can be inferred from temporal continuity and vehicle constraints. These techniques operate by determining point-to-point correspondences within the scan given the current alignment offset estimate, refining that estimate by solving an optimization problem, then repeating until convergence or a maximum number of iterations. Iterative methods often converge to a good solution quickly, but the requirement of a good initial guess suggests that they are not applicable in all situations. Additionally, the fact that scans do not match perfectly (due to incomplete overlap, occlusion, and noise) suggests that iterative techniques should be made robust to outliers.

When the uncertainty in the initial offset is large, and/or it is unknown whether the scans overlap at all, a more globally applicable matching algorithm is required. One possibility is to compute a signature for each scan point (based on the structure of its neighborhood), determine the correspondences, and record the individual alignments between the associated points. The most common alignments are considered as candidate transformations between the two scans (Tomono 2004).

Another approach is to compute a signature for the entire scan, then determine the offsets based on the differences between the two signatures. For instance, histograms encoding orientational and translational information can be used to this end (Weiss et al. 1994; Röfer 2002). The frequencies of surface angles and translational offsets present in a scan become features in the histograms, which can then be aligned to determine the relationship between the two scans. Generalized Voronoi diagram (GVD) representations have also been used to align topographical maps (Silver et al. 2006). Here, similarities in the GVD nodes suggest potential translational offsets. While the point-signature, histogram, and GVD methods require no *a priori* estimate and are not computationally expensive, all only provide an approximate solution and are generally effective only in structured environments (where most points fall along linear features in a limited number of directions).

Our approach to mapping is view-based. When a good initial estimate is available, we use an iterative approach to align laser scans. When we do not have a good initial estimate, a variant of the histogram correlation technique is able to find one to seed the iterative algorithm. This combination is able to exploit the strengths of both techniques and compensate for their weaknesses.

1.2. Outline

Figure 1 illustrates a simplified model of the data flow in the system and provides an outline for the remainder of this article. Section 2 introduces our map building approach. Initially, sequential laser scans are combined to form local maps using a robust iterative scan matching algorithm (left half of Figure 1). A Kalman filter update simultaneously estimates robot motion and constructs local maps based on the computed offsets between scans. Local maps can later be matched to other local maps using a global correlation technique that does not require a prior alignment estimate (right half of Figure 1; details are given in Section 3). In Section 4 we describe how both of the matching techniques can be combined in a SLAM framework to produce high-quality large-scale maps in real-time. Conclusions and areas of future work are given in Section 5.

1.3. Contributions

This article makes several contributions in the area of data association for view-based outdoor SLAM at both the local and global level. For local mapping, we improve existing scan matching algorithms with the addition of robust optimization techniques to handle outliers and imperfect correspondences between the data. Our scan matching algorithm is included in a state estimator used to build accurate local-scale maps (of the order of tens of meters) by including state variables that reduce uncertainties originating from poor scan overlap and sudden velocity changes.

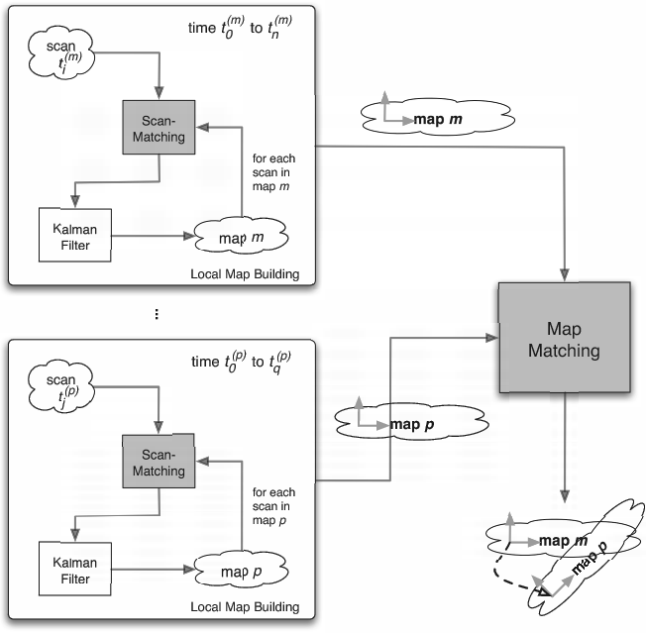


Fig. 1. Data flow in the mapping process. Map building (left-hand side) iteratively aligns laser scans using the scan matching algorithm to form a local map. Completed maps are then matched with existing maps (right-hand side) in order to determine their relative alignment if they overlap.

At a more global level, we introduce a map matching technique for finding correspondences with previously visited areas that works reliably even in highly unstructured environments. Our method builds upon an existing approach based on histogram correlation (Weiss et al. 1994; Röfer 2002), but does not rely on flat surfaces to define dominant angular features to match in each map.

The above techniques enable us to successfully generate what are, to the best of our knowledge, the largest SLAM maps to date. Maps produced from our experiments cover tens of kilometers of travel, an order of magnitude greater than the existing state of the art (Newman 2007; Wang and Thorpe 2004; Ramos et al. 2007; Montemerlo et al. 2003; Williams and Mahon 2004). Our algorithm is also able to handle greater environmental complexity than existing approaches, moving seamlessly between structured and unstructured environments in the presence of dynamic elements.

2. Local Map Building Using Scan Matching

In this section, we describe our method for building local maps and estimating robot state, both requiring only a sequence of laser scans. Robot motion can be estimated from successive laser scans, provided that the frequency of the scans is high

enough relative to the vehicle velocity to ensure sufficient overlap. The approach uses an extended Kalman filter (EKF) in which the current state of the robot is augmented with "snapshots" of the vehicle pose taken at previous time steps.

Scans from the current location are compared to those from the pose snapshots to update the state and estimate the current pose and velocity. This comparison between scans is implemented with a robust version of the iterative closest point (ICP) scan matching algorithm described in Section 2.3. In this fashion, local maps are built up incrementally until the state vector reaches a maximum size, at which point the current map is considered complete and a new local map is created. Determining the spatial relationships between the local maps (e.g., for loop closure) is the topic of Section 3.

2.1. Preliminary Definitions

A *scan* is a set of oriented points resulting from the returns of a single laser sweep, along with the pose of the sensor when each measurement occurred. The points are expressed in the coordinate frame of the sensor.

We distinguish between two levels of maps. A *local map* is a finite sequence of scans, where the pose for each scan is given with respect to a single local coordinate system. A *global map* is a collection of local maps where each local coordinate frame is transformed into a common global frame. Typically in our applications a local map includes the scans from tens of meters of travel, whereas a global map contains all of the local maps built over the entire history of the mapping process.

A transformation expressing coordinate frame b with respect to frame a is denoted by T_a^b. A coordinate frame may be referenced either by the time of a measurement or by a map index.

An *Atlas* graph is a network of local map coordinate frames (nodes of the graph), where each edge represents an estimate of the transformation between overlapping local map frames along with the transformation uncertainty. From the *Atlas* graph, a global map can be generated by computing the global transformations of each node to a common reference frame using the edge (adjacency) transformations.

2.2. Pose Snapshot Kalman Filter

An EKF is used to maintain the state and covariance of all the poses in a local map. There are no direct map features maintained in the filter, and no features need to be extracted from the laser scans. Each saved robot pose maintains a "snapshot" consisting of the raw measurements from the corresponding time step, hence the term pose snapshot Kalman filter (PSKF).

The pose of the vehicle T_m^t in local map m at time t is represented by its translation x_t, y_t and orientation θ_t with respect to a local coordinate frame. The state vector $\mathbf{X}_m$ of the PSKF is the concatenation of the pose at the current time t with poses from selected previous times $t_0, t_1, \ldots, t_N$:

$$\mathbf{X}_m = \begin{bmatrix} \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} \\ \begin{pmatrix} x_{t_N} \\ y_{t_N} \\ \theta_{t_N} \end{pmatrix} \\ \vdots \\ \begin{pmatrix} x_{t_1} \\ y_{t_1} \\ \theta_{t_1} \end{pmatrix} \\ \begin{pmatrix} x_{t_0} \\ y_{t_0} \\ \theta_{t_0} \end{pmatrix} \end{bmatrix} \equiv \begin{bmatrix} T_m^t \\ T_m^{t_N} \\ \vdots \\ T_m^{t_1} \\ T_m^{t_0} \end{bmatrix}.$$

Initially, the PSKF state consists of only the start pose of the robot. New poses are added to the state when there are no pose snapshots within a thresholded distance and angle from the current laser view. These pose snapshots are termed absolute-time (AT) poses, as they refer to a vehicle state at a particular time step. The prediction step of the PSKF only propagates the current robot pose. The state vector is augmented with the new AT pose just before the current pose is propagated. The concept of retaining previous AT poses in the EKF state vector is also known as "delayed-state" or "trajectory-based" SLAM in the literature (Leonard and Rikoski 2001; Newman et al. 2006; Eustice et al. 2006).

Maps are bounded naturally by limiting the number of AT pose snapshots, thereby limiting the maximum size of the state vector and covariance matrix. Since the maximum dimensionality of the PSKF state vector is bounded, the computation time necessary for each time step is also bounded.

The PSKF prediction step uses a process noise model based on the vehicle kinematics and the current velocity estimate (detailed in Section 2.2.1). For example, if we know the vehicle is non-holonomic, we can take advantage of the knowledge that the velocity perpendicular to the wheels is close to zero. Alternatively, if odometry measurements are available, they can be utilized in place of the velocity estimates.

For every time step, the current scan at time t is matched (with ICP) in succession to each of the saved scans (at times t_i). If there is significant overlap between the current scan and any saved scan, the transformation $T_{t_i}^t$ determined by the ICP algorithm is treated as an observation of the relative pose be-

tween the two. The relevant state variables are used in the filter measurement model $\mathbf{z} = h(\mathbf{X})$ to update the PSKF state and covariance,

$$\begin{aligned} \mathbf{z} &= h(\mathbf{X}), \\ T_{t_i}^{t} &= \ominus T_{m}^{t_i} \oplus T_{m}^{t} \\ &= T_{t_i}^{m} \oplus T_{m}^{t}, \end{aligned}$$

where $\ominus$ and $\oplus$ are the transformation inverse and composition operators, respectively. The relative pose $T_{t_i}^{t}$ and its measurement covariance is computed from the iterative scan match algorithm described in Section 2.3. Note that the updates correct the current pose as well as all of the past poses in the PSKF state with which some overlap between its associated scan and the current scan is detected.

2.2.1. Fixed-lag Pose Snapshots

We augment the PSKF state model with additional pose states that are used to reduce the growth of error while mapping. In addition to AT pose states that are at a distance-based interval, the augmented filter maintains pose states that are offset from the current pose at a fixed lag (FL):

$$\mathbf{X}_m = \begin{bmatrix} T_m^t \\ T_m^{(t-1)} \\ T_m^{(t-2)} \\ T_m^{t_N} \\ \vdots \\ T_m^{t_1} \\ T_m^{t_0} \end{bmatrix}. \tag{1}$$

The FL pose snapshots are the poses from the previous n scans (where n is typically less than or equal to three), and they are replaced with the latest n scans at every prediction step. FL poses are updated in exactly the same manner as the AT pose states. Note that in Equation (1) the FL poses are indexed relative to the current pose, whereas the AT poses have absolute indices.

There will typically be more overlap between the current scan and the scan from the FL state than the overlap between the current scan and the nearest AT state. This improves the quality of the transformation update. The filter will also be able to incorporate more updates overall with the use of FL states which reduces the growth of error during mapping (see Section 2.4). Figure 3 illustrates state evolution in a typical pose snapshot PSKF with a single FL pose. As indicated in the figure, the inclusion of FL states introduces additional measurements between successive poses whose scans are expected to share more overlap (dashed arcs).

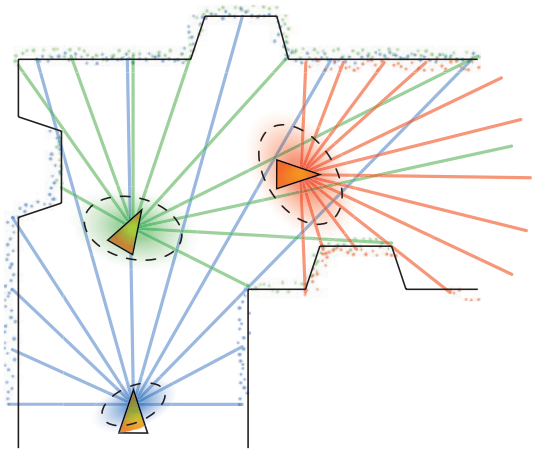


Fig. 2. The map is represented by saved robot poses (snapshots) and their respective scans. The alignment between the current scan and any suitable saved scan is used to localize the robot.

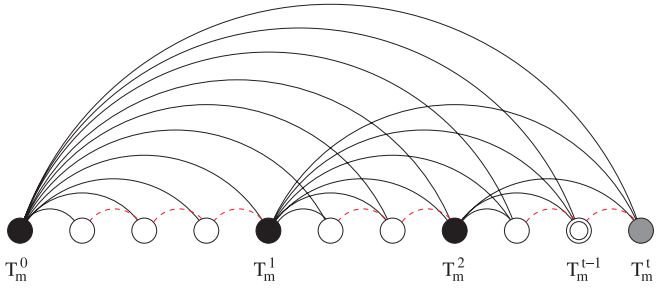


Fig. 3. A graphical illustration of PSKF state evolution indicating which state poses are matched during local mapping. Time increases along the horizontal direction; the circles going from left to right representing the robot pose at each time step. The solid circles represent the absolute-time (AT) poses (labeled with the transformation defined by that pose), whereas the open circles represent previous poses no longer in the state. The double circle corresponds to the current location of a single fixed-lag (FL) state. For a given node, the arcs emanating to the left represent the measurements (scan matches) that were made when the current state was at that position. The use of FL poses provides additional measurements (dashed arcs) which reduce the growth of uncertainty (i.e. more scan overlap is typically expected due to the proximity of the FL pose with the current pose).

In addition, the velocity estimated by the relative motion between the current and most recent FL pose state is used in the prediction step of the filter (i.e. it is used within the process noise model). This velocity estimate is more accurate than using the latest AT state and enables the filter to handle larger accelerations and turns without losing lock.

2.3. Iterative Scan Matching

ICP is a simple algorithm used to align two clouds of points with unknown correspondences. Each iteration of the algorithm proceeds in two steps. In the first step, point correspondences are found by associating each point from one scan to its closest point in the other scan. The second step then finds a coordinate transformation that minimizes the error between the matched point correspondences. These two steps are repeated until convergence is achieved or a maximum number of iterations has occurred.

The pseudocode for the iterative scan matching algorithm is as follows:

```
ITERATIVE-SCAN-MATCH
1  for i ← 1 to maxiter, or converged
2  do χ_ab ← FIND-CLOSEST-POINTS(s_a, s_b, T_b^a)
3     T_a^b ← UPDATE-TRANSFORM(χ_ab, s_a, s_b, T_a^b)
4  Σ_ab ← COMPUTE-COVARIANCE(χ_ab, s_a, s_b, T_a^b)
5  return {T_a^b, Σ_ab}
```

where s_a and s_b are the laser scans, T_a^b is the relative coordinate transform between the scans' centers, χ_{ab} is the correspondence between the points of the two scans, and Σ_{ab} is the covariance of the final alignment.

2.3.1. Surface Normals

Since surfaces are not sampled at exactly the same points in different scans, the Euclidean distance between points is not a good metric for evaluating the distance between scans. Therefore, we use a metric that more accurately represents the error between the surfaces: the component of the vector difference between two scan points in the direction of the surface normal to the point in the current scan (Potmesil 1983; Chen and Medioni 1991) (Figure 4). This avoids the need for interpolating scan points, and reduces biases from non-uniformly sampled surfaces. Empirically, it has been demonstrated that using this type of error metric improves the convergence speed of the ICP algorithm (Rusinkiewicz and Levoy 2001). Normals can also be used in the correspondence step by restricting to matches between points that have similar normal directions (Pulli 1999). As we are effectively matching surfaces rather than points, it is more accurate to call this algorithm "iterative closest surface" (ICS) rather than "iterative closest point".

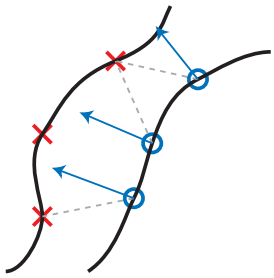


Fig. 4. Scan surface point correspondence using normals. The dashed lines indicate the matches between the points from one scan (circles) to points in the other scan (crosses). The error between scans is that measured in the direction of the surface normals, indicated by the arrows.

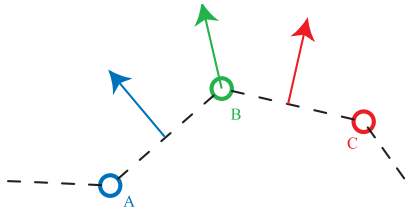


Fig. 5. The normal for point B is computed from the average of the normals of the lines $\overline{AB}$ and $\overline{BC}$.

The surface normals for each scan point are approximated using the assumption that points are sampled from a continuous surface. Given the sequential points A, B, and C, the normal for point B is taken as the average of the normal for lines $\overline{AB}$ and $\overline{BC}$ (see Figure 5). The continuous surface assumption does not hold at occlusion boundaries or at step edges in the scan. Therefore, if the distance between points A and B or between B and C is greater than a threshold, only the shorter line is used to determine the normal. When both distances are too large—e.g., when a thin pipe is measured—the normal is simply set to point towards the origin of the scan.

2.3.2. Transformation Update

In each ICS iteration, after the scan surface point correspondences have been computed, the prior alignment transformation T_a^b is updated. The update should minimize the alignment error given the current correspondences. The formula for the alignment error uses the difference between each corresponding scan point in the direction of the normals:

$$E_{\text{align}} = \sum_{(a,b)\in\mu_{ab}} \left(\mathbf{n}_a^T \left(\mathbf{p}_a - T_a^b \mathbf{p}_b\right)\right)^2, \qquad (2)$$

where $\mathbf{p}_a$ and $\mathbf{p}_b$ are the scan point vectors, $\mathbf{n}_a$ is the normal of point a, T_a^b is the alignment transform, and E_{align} is the total alignment error. This error metric is similar to the tangent-based method of Lu and Milios (1994).

The objective is to determine the transformation that minimizes the alignment error. There is no linear solution for the optimal transformation due to the rotational component. Instead, the error function is linearized about an initial guess for the transformation, and an optimal linear correction is solved for. The alignment error can be written in vector form:

$$E_{\text{align}} = \mathbf{h}(T_a^b)^T \mathbf{h}(T_a^b), \tag{3}$$

where each row of the vector function $\mathbf{h}(\cdot)$ is a term from the summation in Equation (2). The linearized error can then be expressed with a first-order Taylor expansion:

$$\mathbf{h}(T_a^b) \approx \mathbf{h}(T_{a0}^b) + \mathbf{H} \begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix}, \tag{4}$$

where $\mathbf{H}$ is the Jacobian of $\mathbf{h}$ with respect to x,y, and θ.

Differentiating Equation (3) with respect to the unknowns (x, y, and θ), using the approximation in Equation (4), and setting the expression to zero yields the linear system that is solved for the optimal linear solution:

$$\begin{aligned} 0 &= 2\mathbf{H}^T \left(\mathbf{H} \begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} + \mathbf{h}(T_{a0}^b) \right), \\ \begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} &= -\left(\mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{h}(T_{a0}^b), \end{aligned} \tag{5}$$

where each row $\mathbf{H}_i$ of $\mathbf{H}$ is

$$\mathbf{H}_i = \begin{bmatrix} -n_x \\ -n_y \\ \begin{bmatrix} n_x \\ n_y \end{bmatrix}^T \begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} p_{bx} \\ p_{by} \end{bmatrix} \end{bmatrix}^T.$$

The updates are added to the transformation parameters, and subsequently the linearization procedure may be iterated several times to ensure convergence.

It is important to note that this solution contains only one constraint per correspondence as opposed to two constraints when using point correspondences. This distinction prevents the minimization from being biased by the scan point locations.

2.3.3. Robust Outlier Weighting

To mitigate the effect of outliers in the data (from non-overlapping regions of the scans, moving objects and/or ground strikes) the error of the matches is modified by a Lorentzian which smoothly down-weights errors when they become too large. The Lorentzian weighting is equivalent to assuming a Cauchy (instead of a Gaussian) error distribution (Press et al. 1992).

The alignment error equation is modified by a Lorentzian-like function $\rho(\cdot)$:

$$\rho(x) = \log\left(\bar{r}^2 + x^2\right), \tag{6}$$

$$E_{\text{robust}} = \sum_{i=1}^{N} \rho\left(h_i(T_a^b),\right) \tag{7}$$

where $\bar{r}$ defines the soft outlier threshold.

Again the parameters which minimize the error are computed by differentiating the error function and setting the results to zero. The only difference between using Equation (7) and Equation (3) is that the Lorentzian adds the term $\frac{1}{(\bar{r}^2+h_i^2)}$ to each row. Each term can be seen as a weight for the constraint from each surface point correspondence that depends on the initial error. The weights are collected into a diagonal matrix $\mathbf{W}$ and the weighted least-squares solution is computed:

$$\mathbf{W} = \begin{bmatrix} \ddots & & 0 \\ & \frac{1}{\left(\bar{r}^2+h_i\left(T_{a0}^b\right)^2\right)} & \\ 0 & & \ddots \end{bmatrix}, \tag{8}$$

$$\begin{aligned} 0 &= \mathbf{H}^T\mathbf{W} \left(\mathbf{H} \begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} + \mathbf{h}(T_{a0}^b) \right), \\ \begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} &= -\left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1} \mathbf{H}^T\mathbf{W}\mathbf{h}(T_{a0}^b). \end{aligned}$$

Even though the optimal minimization of this Cauchy error distribution is non-linear and requires multiple iterations for convergence, empirical evaluations have determined that one iteration suffices, since the minimization step is repeated in the iterations of the ICS algorithm in any case.

The final weights for each point correspondence after convergence is used to initialize the weights in subsequent scan matches, and the aggregate weights from multiple matches are used to determine the outlier points in the environment which can be ignored in later processing steps.

2.3.4. Alignment Transformation Covariance

The covariance of the scan match transformation is determined after the final ICS transformation update step. The final surface correspondence is used with the alignment error (Equation (2)) to determine the average point error variance. Subsequently, the point error variance is used in conjunction with the Jacobians of the alignment error with respect to the transformation parameters to form the covariance of the alignment transformation.

The variance σ_s^2 of the surface point match error is computed as the sample variance of the terms in the summation of Equation (2):

$$\sigma_s^2 = \frac{1}{N-1}\sum_{i=1}^{N} h_i(T_a^b)^2 = \frac{1}{N-1}E_{\text{align}}, \tag{9}$$

where N is the number of surface point correspondences.

The covariance of the transformation parameters $[x\ y\ \theta]^T$ is computed by taking the covariance of the final update (Equation (5)) using the assumption that $E[\mathbf{h}\mathbf{h}^T] = \sigma_s^2\mathbf{I}$:

$$\begin{aligned}\Sigma_{ab} &= E\left[\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix}\begin{bmatrix} dx & dy & d\theta \end{bmatrix}\right] \\ &= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T E[\mathbf{h}\mathbf{h}^T]\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1} \\ &= \sigma_s^2(\mathbf{H}^T\mathbf{H})^{-1}. \end{aligned} \tag{10}$$

The covariance matrix is used as the measurement noise covariance when updating the PSKF with Equation (1). The covariance of the transformation update is dependent on the number of corresponding scan points and the average error of each correspondence, as well as the geometry of the scans. For example, the covariance will be ill-conditioned when the normals of the scans points do not span a two-dimensional space. This is particularly evident in environments such as long hallways where only two parallel flat walls are observed. In such a case, the Kalman filter will be unable to reduce its uncertainty along the direction of the hallway.

2.4. Scan Matching Experiments

2.4.1. Robust Versus Gated Scan Matching

To demonstrate the effect of using a robust approach to scan matching, we compare our algorithm to a gated least-squares scan matching algorithm. A SICK LMS 291 laser scanner was mounted on the roof of a sport utility vehicle providing a 180° field of view directed to the front of the vehicle parallel to the local ground plane. Two laser scans were taken (approximately 2 m apart) as the vehicle drove along a hilly suburban road, thereby creating a series of ground strikes in each scan. As the vehicle attitude is different between the two scans, the position of the ground strike scan points are not consistent between the scans and therefore represent outliers for the matching algorithm.

Table 1. Performance comparison of robust and least-squares ICS algorithms for selected threshold values

Scan match algorithm	Threshold (m)	Convergence area (%)	Error (cm)
Robust (Cauchy)	0.125	100	11.3
	0.25	100	15.2
	0.5	100	23.5
Least squares (Gaussian)	0.125	0.14	8.4
	0.25	2.16	6.4
	0.5	9.96	5.7
	1.0	52.7	30.3

In the gated least-squares scan matching algorithm, each point correspondence is weighted equally unless the distance between each point is greater than the gate threshold, in which case it receives zero weight. In the robust case, each correspondence is weighted according to Equation (8), effectively defining a soft threshold.

The two scan matching algorithms are evaluated based on their convergence areas, the number of iterations required to converge, and the error of the solution when the algorithm converges to (approximately) the correct solution. To test these convergence properties, artificial initial offsets in position and angle are added to the second of the two scans. The size of the artificial offsets are larger than the maximum initial offsets we expect to see in typical use in our SLAM framework (Section 4). The upper row of Figure 6 shows the convergence region of the gated least-squares algorithm for different values of the gate threshold. The lower row shows the convergence regions for the robust scan matching algorithm for three different various $\bar{r}$ values. Here we can clearly see that the gated least-squares algorithm is very sensitive to initial conditions and requires a fairly close initial guess to converge. In contrast, the robust version converges everywhere within the region of interest.

Table 1 contains a summary of statistics from the experiments, including artificial initial angular offsets between $-10°$ and $10°$. Here, the convergence area is defined as the percentage of the 4m × 4m × 20° offset region from which the scan matcher is able to converge near the correct solution. The error given is the median positional error of the resulting solutions. We observe that, even with angular offset, the robust algorithm is able to converge over the entire region. The gated least-squares algorithm has a significantly smaller convergence area, but can be more precise when it does converge.

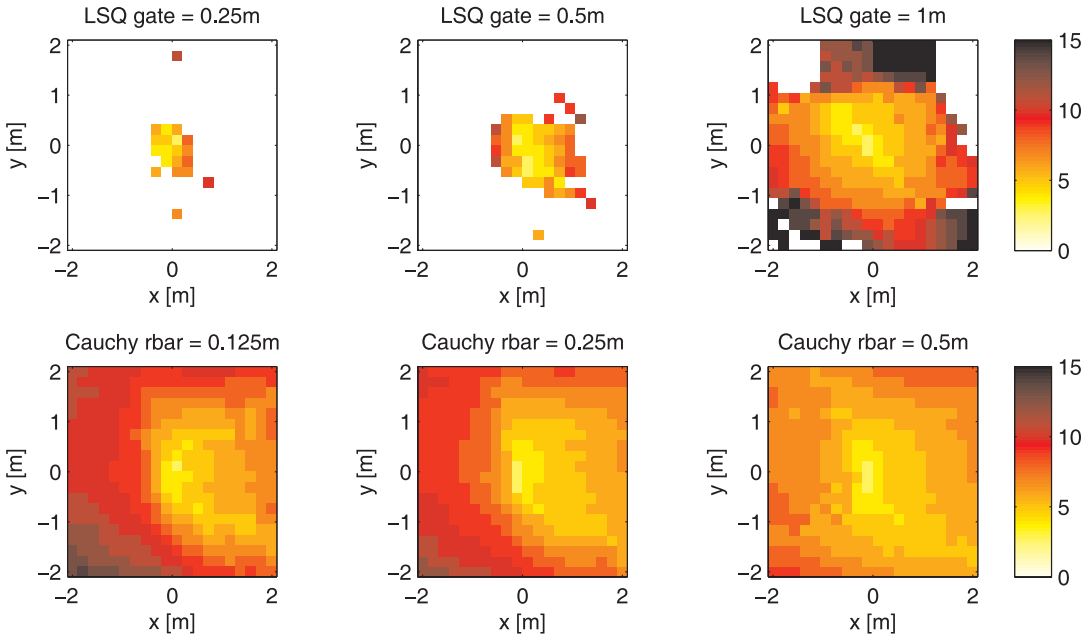


Fig. 6. A comparison of the local convergence regions of the robust and gated least-squares scan matching algorithms with varied parameters. Two unregistered scans containing some outlier points (due to ground strikes) are input to each algorithm with an additional artificial initial positional offset (and a 0° angular offset) indicated by the coordinate axes in the subplots. A white pixel signifies that the algorithm did not converge from the initial conditions; otherwise the pixel intensity indicates the number of iterations to converge to the (approximately) correct solution.

The effects of the hard and soft thresholds on the resulting error can be more clearly seen in Figure 7, which shows the error of the match solution as the hard and soft threshold parameters are increased. Here we observe that the gated least-squares algorithm has smaller error for small threshold values, but the error quickly increases as the threshold approaches the distance between the outlier points. The robust scan matching algorithm has a more graceful transition as the soft threshold results in less sensitivity to the threshold value. The gated least-squares algorithm is much more brittle as the error can suddenly jump with a small change in the parameter. In practice, this makes it difficult to select an optimal threshold, since outlier distances are unpredictable.

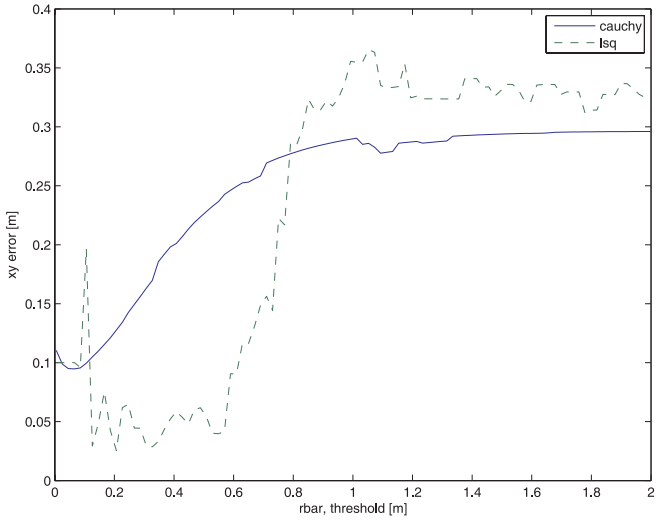


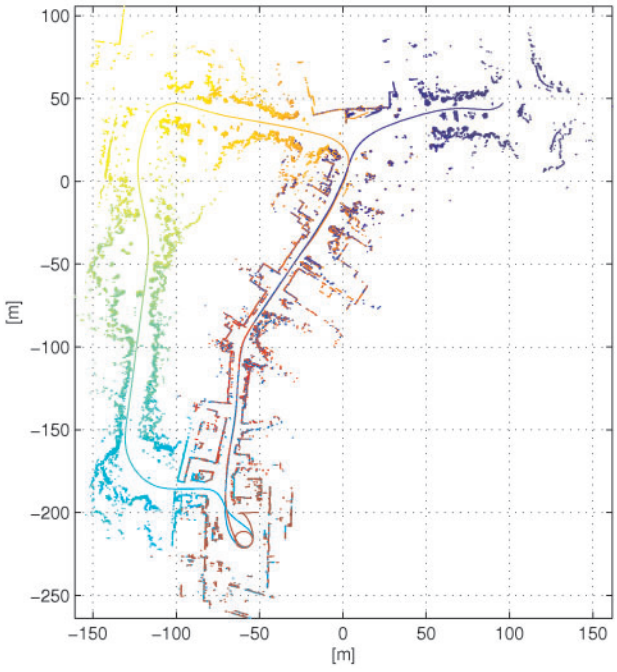
Fig. 7. A comparison of scan match errors as the threshold is varied for the robust scan matching algorithm and a gated least-squares scan matcher.

2.4.2. Effect of Fixed-Lag States

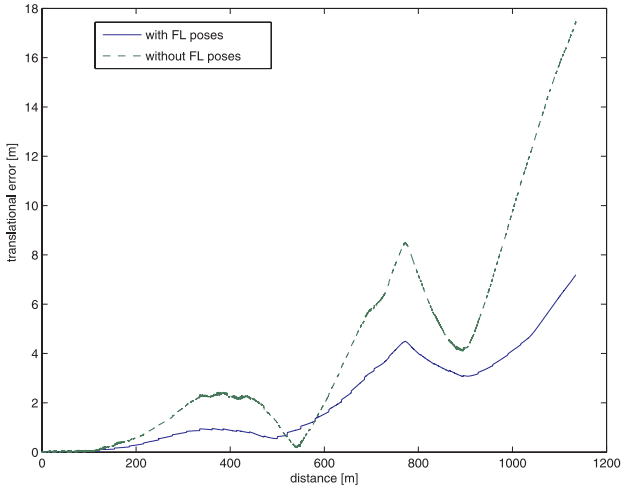
Additional experiments were carried out to demonstrate the effect of including FL poses in the PSKF state vector during local map building. Given a ground truth reference map, we can evaluate the behavior of our approach with respect to the accumulation in positional error (i.e. the distance between the ground truth value and the localizer's estimate) by mapping the same data set with and without FL poses. Two SICK LMS laser scanners were mounted on top of our sport utility vehicle to provide a 360° field of view parallel to the local ground plane. The vehicle was driven around for a short distance of approxi-

mately 1140 m at speeds of around 30 km/h. The trajectory of the vehicle included structured areas containing buildings, as well as unstructured areas containing trees and other sparsely scattered objects. Scans were processed at a rate of approximately 20 Hz for each laser, with 361 range readings per scan. No odometry or velocity information was available from the vehicle during these experiments. Figure 8(a) shows a map of this traverse constructed using our full SLAM framework detailed in Section 4 (and uses the map building algorithm described in this section). We consider this map as a reference map due to the unavailability of absolute ground truth data.

On the same data set used to construct the reference map, we ran the mapping algorithm (PSKF with scan matching) without the full SLAM framework (i.e. without global map matching to detect revisited areas and close loops) and recorded the accrued trajectory error with respect to the reference trajectory. This process was repeated with FL pose snapshots included in the PSKF, and without FL poses. The estimated positional errors over the course of the trajectories are shown in Figure 8(b). Here, we can see that in both cases the error is small as a percentage of distance traveled for this length of trajectory, but clearly the addition of FL poses reduces the open-loop error significantly.



(a) Reference SLAM map.



(b) Accrued positional error.

Fig. 8. Results from a small-scale mapping run. (a) A reference map created using a full SLAM framework (details in Section 4). Note the presence of both structured and unstructured surroundings. (b) The positional error accrued over the course of the trajectory using scan matching only (no global data association to detect revisited locations), with and without FL pose snapshots in the PSKF state. The error is with respect to the reference map, rather than ground truth (which was unavailable).

3. Map Matching

The previous section describes an algorithm for estimating the alignment between laser scans when a good initial guess is available, and a method for constructing accurate local maps (consisting of a sequence of aligned scans) based on these matches. In this section, we consider the problem of determining the correspondences at a larger scale—between local maps. This facilitates the detection of loop closures and instances of repeated traversal including approaches to the same region from different directions. Here, there is typically a larger uncertainty in the initial arrangement between the maps, and in many cases the maps should not match at all. Therefore, some extra steps are required to decide *if* the maps overlap, and to make a rough guess as to *how* they overlap. This estimate can be used to seed the ICS algorithm to ensure that it converges quickly to the global minimum.

3.1. Histogram Matching

In order to quickly align two maps, each local map requires a compact representation of its salient characteristics; this representation can be used to distinguish the local map from other maps and to determine its transformation to another map if they overlap. Our map matching representation consists of an *orientation histogram* of the scan normals plus a set of weighted *projection histograms* created from the orthogonal projections of scan points onto lines from discrete orientations. Strong correlations between histograms can be used to

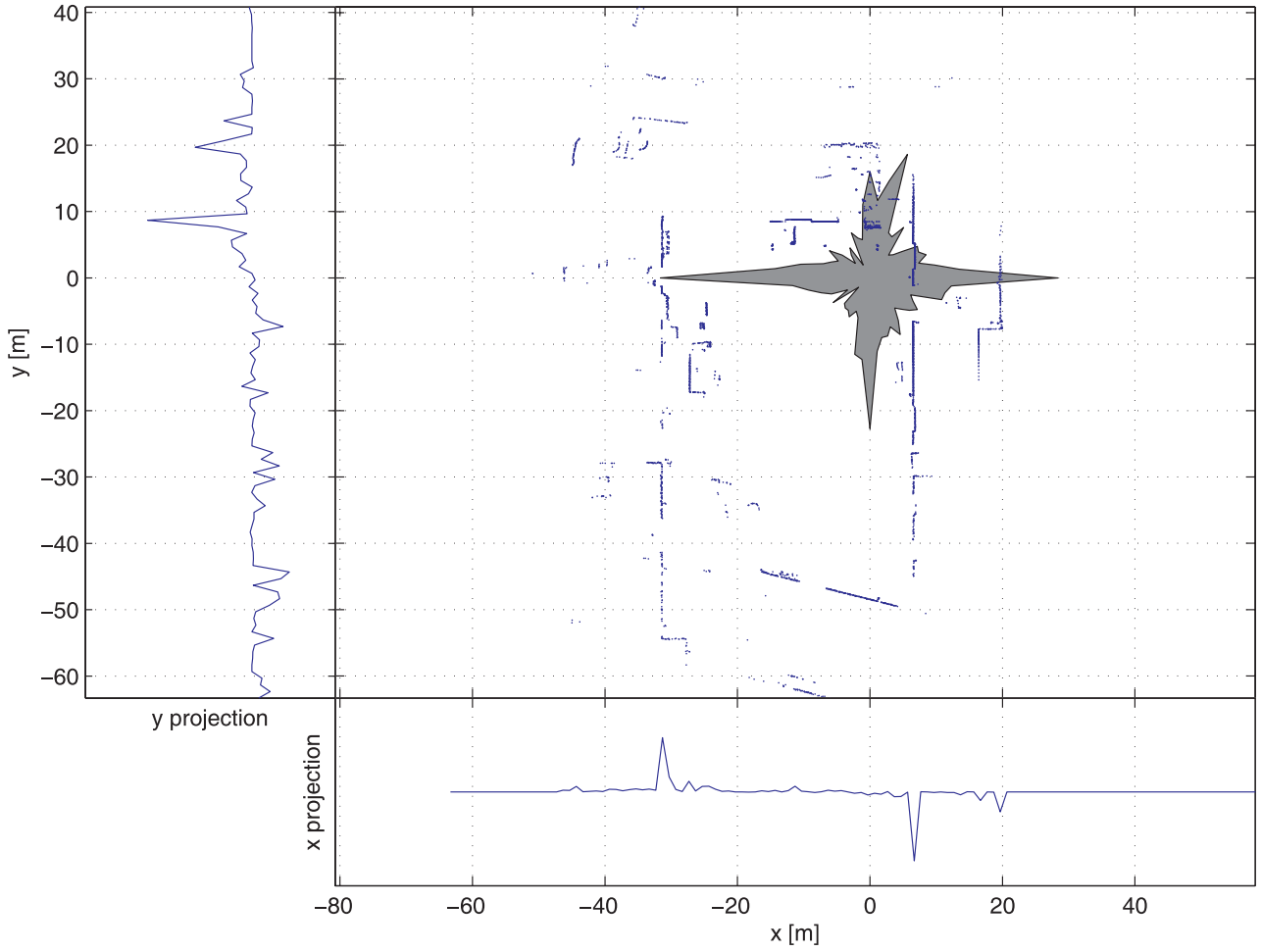


Fig. 9. Examples of orientation and projection histograms for a local map. The orientation histogram is depicted as a polar plot (star-shaped polygon) in the center of the image. The peaks represent the dominant surface orientations. For example, the diagonal wall towards the bottom of the map appears as the non-axis-aligned peak near the top of the orientation histogram. The projection histograms along the x and y axes are shown on the bottom and left of the map, respectively. The y-aligned walls on the left side of center are easily identifiable as the positive peak in the x-oriented projection histogram. The y-aligned walls on the right contribute to the large negative peak as their surface normals are in the negative x direction.

deduce potential map matches: first the orientation histograms can be used to resolve angular offset, followed by the projection histograms associated with that direction to resolve the translational component. We have made several enhancements to similar existing histogram matching techniques (Weiss et al. 1994; Röfer 2002), improving the quality of the match and generalizing the approach to additionally function in unstructured outdoor environments.

The *orientation histogram* is used to compute the rotational offset between pairs of local maps irrespective of any translational offset. To construct a histogram, the unit circle is discretized into a set of equally-sized bins and the frequency of surface normals (see Figure 5) in the map for each bin represents the histogram value for that bin. The peaks of the orientation histogram, most apparent when there are numerous flat surfaces visible in the laser scans, represent the dominant surface orientations. In general, the histogram's bin size should be commensurate with the noise and certainty of the scans in the local map. We have determined empirically that an angle bin size of 5.625 degrees ($360^\circ/64$) works well in our environment of industrial buildings, city and suburban streets, and residential neighborhoods. An example of a map and its associated orientation histogram is given in Figure 9.

The set of weighted *projection histograms* are used to determine the translational offsets between pairs of maps once their rotational offset has been determined. Each projection histogram in the set $H_m(\theta_p, d)$ is generated by orthogonally projecting every scan point, (x_i, y_i) in map m, on to a line

with angle θ_p and creating a histogram of the offsets d_i of the points with bin centers d, weighted by the dot product of their surface normals (n_{x_i}, n_{y_i}) with the line:

$$d_i = x_i \cos\theta_p + y_i \sin\theta_p,$$

$$H_m(\theta_p, d) = \sum_{\|d_i - d\| < \frac{\Delta}{2}} n_{x_i} \cos\theta_p + n_{y_i} \sin\theta_p,$$

where Δ is the histogram bin size.

The dynamic range of the projection histograms is enhanced by weighting each point according to its surface orientation. Points with surface orientations parallel to the projection line are de-weighted so that they do not blur the histogram, whereas points with surfaces perpendicular to the line are given more weight. Furthermore, since the weights can be negative, it is possible to distinguish contributions from scan points with opposite normal orientations (Figure 9). This improves the dynamic range and saliency of the projection histograms, since the cumulative contribution from long walls will not wash out fine structure when the projection lines are parallel to the walls. Likewise, walls perpendicular to the projection line will only match with walls of the same orientation.

The orientation of each projection line and ultimately the number of projection histograms generated are determined by the number of angle bins from the orientation histogram. For the projection histograms, the size of the offset bins should be small enough that the details of the environment's structure are captured, but not so small that the bin counts are dominated by noise. We have determined empirically that a bin size of 1 m works well for our environments.

An example of an entire set of weighted projection histograms is depicted as an image in Figure 10. Each column in the figure represents a projection histogram with projections taken along a line at a specific angle θ_p aligned with the orientation histogram bins.

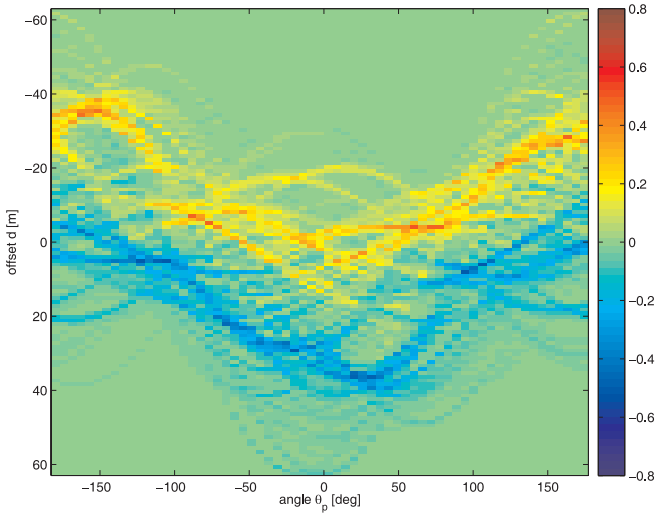


Fig. 10. A sample set of weighted projection histograms displayed as an image. Each column represents a projection histogram taken along a different projection line angle θ_p. The negative weights help to distinguish contributions from scan points with opposite orientations. The columns at $\theta_p = 0^\circ$ and $\theta_p = 90^\circ$ correspond to the x- and y-aligned histograms in Figure 9.

3.2. Histogram Correlations

The goal of the histogram matching algorithm is to determine quickly whether or not any pair of local maps match and, if so, to find the transformation between their coordinate frames. The general idea of this approach is to determine the angular offset between the two maps by first correlating the orientation histograms. Then, given that angular offset, the translational offsets along and perpendicular to that direction can be found through correlation of projection histograms.

As we will see, histogram matching for the purpose of determining angle is best-suited for structured environments which lead to salient modes in the orientation histogram corresponding to flat surfaces in the maps. For unstructured environments a more reliable technique, introduced in Section 3.3, is required.

3.2.1. Orientation Histogram Correlation

The first step of the matching process is to compute the correlation between the maps' orientation histograms in order to determine possible rotational offsets. In order to avoid boundary effects, the correlation is computed by circular convolution, where the histograms have been normalized by their Frobenius norms such that the autocorrelation of the histogram with itself has a maximum value of one. The location of the peak in the correlation represents an estimate for the rotational offset between the maps. Due to noise, incomplete overlap, and periodicity, other local maxima may also be indicative of the true orientational offset; therefore, multiple peaks are carried through for later evaluation.

3.2.2. Projection Histogram Correlation

A peak from the orientation histogram correlation suggests a candidate angular alignment correction between the two maps; thus, for each candidate offset, projection histograms are correlated for the two maps with the projection lines in each offset by this peak angle. More specifically, given a candidate rotational offset θ_o, if the projection line angle in the first map is taken to be θ_p, then the line angle used in the second map is $\theta_p + \theta_o$ (i.e. a projection histogram for map a $H_a(\theta_p, d)$ corresponds to the projection histogram $H_b(\theta_p + \theta_o, d)$ for map b).

The algorithm to find the translational offset then works as follows. From the first map, two perpendicular projection histograms are selected (two are required in order to resolve the alignment components in both dimensions) and correlated with their corresponding histograms from the second map (with the projection lines rotated by θ_o) to compute the translational offsets. We have empirically determined that it is typically best to choose one of the two projection histograms to be the one with the least entropy (typically the one corresponding to the largest peak in the orientation histogram) and the other its orthogonal partner, as this choice slightly improves the reliability of the match. The peaks in the correlations of projection histograms represent estimated offsets given a candidate rotational offset. Examples of projection histogram correlations are given in Figures 15(d) and (e) later. (The offset angles used for the correlations are based on a peak from Figure 15(c).)

The translation vector between the maps' coordinate frames is calculated by solving the simple linear system for t_x and t_y:

$$\begin{pmatrix} \cos\theta_p & -\sin\theta_p \\ \sin\theta_p & \cos\theta_p \end{pmatrix} \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} d_{\theta_p} \\ d_{\theta_p+\frac{\pi}{2}}, \end{pmatrix}$$

where θ_p is the angle that the first projection line makes with the x-axis, d_{θ_p} is the offset of the maximum peak of the correlation of the first projection histogram, and $d_{\theta_p+\frac{\pi}{2}}$ is the offset computed from the correlation of the second projection histogram.

The accuracy of the resulting transformation is dependent upon the bin sizes. Even though the alignment may be coarse, as in Figure 15(f), the accuracy should be sufficient for the iterative scan matching algorithm (as described in Section 2.3) to converge on the precise match.

3.3. Entropy Sequences

We have observed that the orientation histograms are not reliable in unstructured environments. The absence of flat surfaces in many typical outdoor scenes results in mostly uniform orientation histograms lacking any dominant peaks.

Therefore, as an alternative to using orientation histograms, we have found it more reliable to compute the rotational offset using a sequence of measures $\mathcal{E}(\theta_p)$ based on the entropies of projection histograms $H(\theta_p, d)$. Intuitively, the entropy encodes the uniformity of the histograms: maximum entropy occurs for a uniform histogram, and the minimum entropy occurs when only a single histogram bin is non-zero. Therefore, the entropy effectively measures the presence and sharpness of peaks, which varies with the angle of the projection line. While variance might also be considered to describe the spread of a histogram, unlike entropy it is unable to capture the "fuzziness" in the case of multimodal distributions (which is the norm for our application) and would be sensitive to the boundaries of the map.

Each projection histogram in the set for a given map can be normalized into a probability distribution from which the entropy is computed. The *entropy sequence* is composed of the entropy measure from each angle θ_p of the projection line:

$$\begin{aligned} \hat{H}(\theta_p, d) &= \frac{|H(\theta_p, d)|}{\sum_d |H(\theta_p, d)|}, \\ E(\theta_p) &= -\sum_d \hat{H}(\theta_p, d) \log\left(\hat{H}(\theta_p, d)\right). \end{aligned}$$

The absolute value of each projection histogram element is necessary since the weights may be negative.

Prior to matching, the entropies are transformed first to exaggerate the peaks (by exponentiation) and then negated, shifted, and normalized to make the sequence analogous to an orientation histogram:

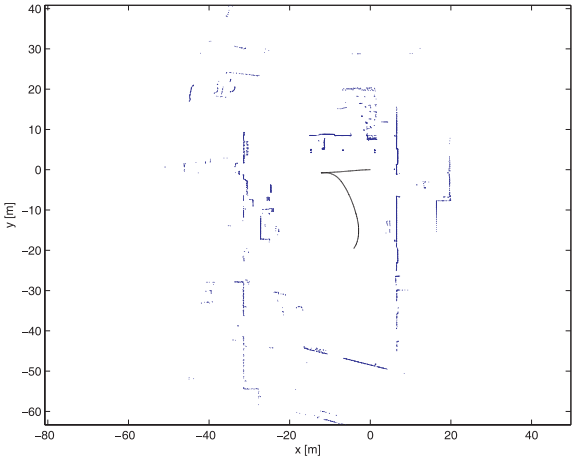
$$\mathcal{E}(\theta_p) = -\frac{2^{E(\theta_p)} - \max_{\theta_p}(2^{E(\theta_p)})}{\sqrt{\sum_{\theta_p}(2^{E(\theta_p)} - \max_{\theta_p}(2^{E(\theta_p)}))^2}}.$$

Since two projection lines differing by an angle of 180° contain the same distribution of points, entropy sequences repeat every 180°. Thus, each peak in the correlation of entropy sequences produces two candidate orientation offsets, 180° apart. This ambiguity, however, is often resolved during the subsequent steps due to the fact that the false offset will have a low response when correlating the projection histograms.

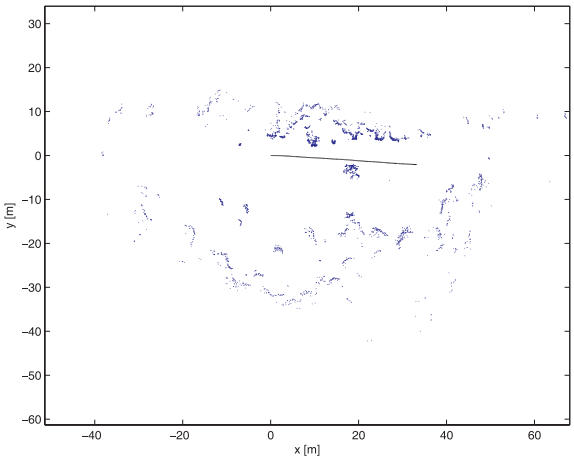
Figure 11 illustrates and contrasts the behavior of orientation histograms and entropy sequences in structured and unstructured environments. The maps used in the figure are taken from the data set used in Section 2.4 where a vehicle with two mounted SICK laser scanners drove 1140 m. Figures 11(a) and 11(b) show example maps of a structured environment containing walls and buildings, and an unstructured one containing natural obstacles such as trees.

In Figures 11(c) and 11(d) the orientation histograms and entropy sequences are given for the two maps. For the structured map, both the histogram and the entropy sequence have similar peak locations, although the entropy sequence peaks are broader. This is generally the case for structured environments, where the entropy sequences are a less informative measure. In contrast, for the unstructured map, the orientation histogram lacks any obvious peaks and is dominated by noise, whereas the entropy sequence is more informative, having a strong signal with clear peaks.

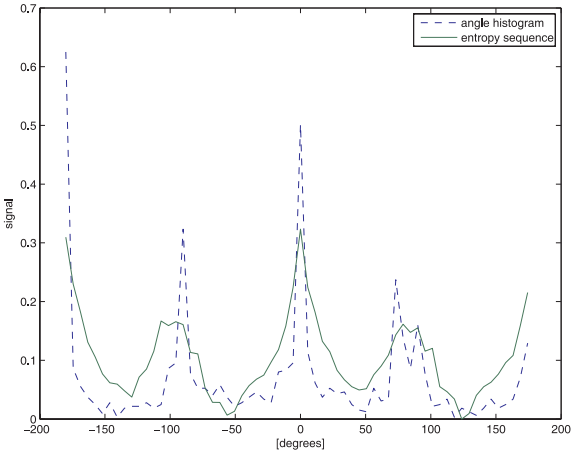
The autocorrelations of the signals are shown in Figures 11(e) and 11(f). The autocorrelation indicates what we might expect to see under ideal conditions, where the maps we are comparing match perfectly. Here, we see that both the angle histogram and entropy sequence have similar autocorrelation peaks for the structured map (although the histogram peaks are stronger), while for the unstructured map clear peaks are only present for the entropy sequence autocorrelation. This



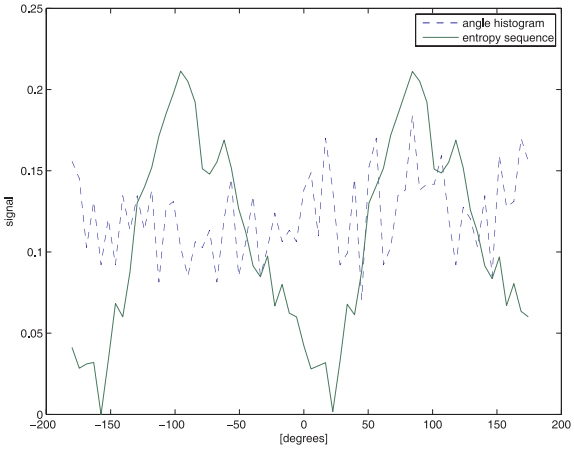
(a) Structured map.



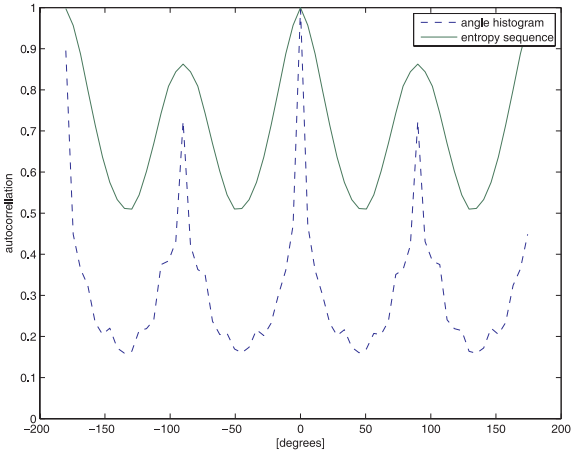
(b) Unstructured map.



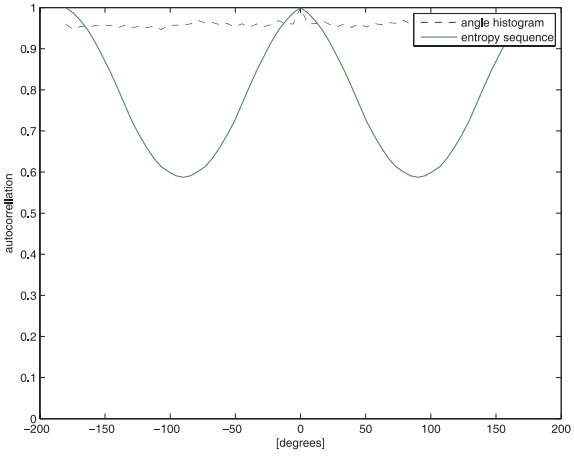
(c) Orientation histogram and entropy sequence (structured).



(d) Orientation histogram and entropy sequence (unstructured).



(e) Autocorrelation of orientation histogram and entropy sequence (structured).



(f) Autocorrelation of orientation histogram and entropy sequence (unstructured).

Fig. 11. A comparison of the orientation histogram and entropy sequence correlation techniques for structured and unstructured maps. Autocorrelation is used to demonstrate the match characteristics that would be obtained under ideal conditions where the maps overlap perfectly. Note that, for the unstructured map, the entropy sequence is able to produce salient peaks where the histogram correlation is unable to do so.

example demonstrates that in unstructured environments orientation histograms can be too noisy to be reliable for matching, necessitating the use of entropy sequences.

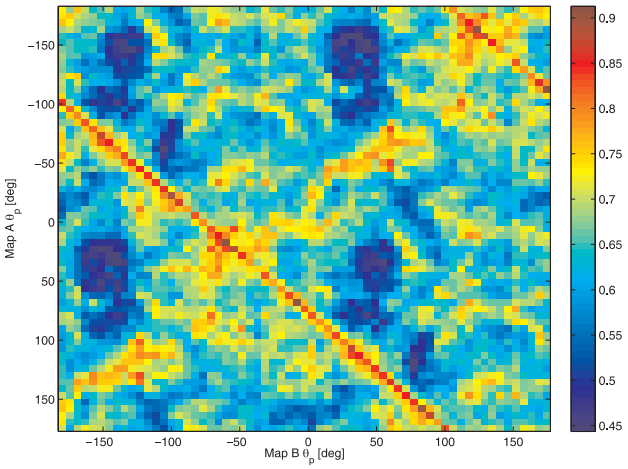
3.4. Exhaustive Approach

When both the orientation histogram and entropy sequence correlations are too noisy or have no reliable peaks, an exhaustive approach can be used to determine the offsets using only the projection histograms. Here all projection histograms from one map are correlated against every projection histogram in the other map and the global peak value of that correlation is recorded (Figure 12(a)). All of the maximum correlation values from histograms with equal rotational offsets are averaged and arranged in a sequence indexed by the offset, which we call the *exhaustive correlation sequence* (Figure 12(b)). Local peaks in this sequence determine candidate rotational offsets between the two maps (analogous to the peaks in the orientation histogram and entropy sequence correlations; Figure 13). We can now carry on as in Section 3.2 to determine the translational offsets.

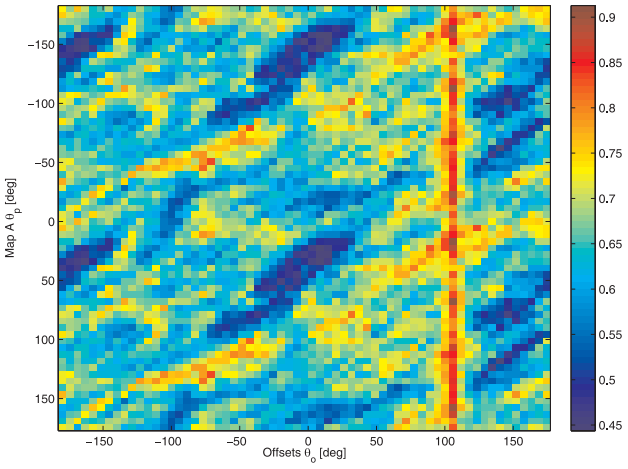
The exhaustive method is computationally expensive in comparison to using the orientation histogram or entropy sequence correlations, and therefore should only be used as a last resort when the other two methods do not find quality matches. When implemented naïvely, the exhaustive method runs in $O(n^4)$ time, whereas the regular correlations have time complexity $O(n^2)$. However, the correlations can be implemented as multiplications in the Fourier domain, thereby improving the running time of the exhaustive matching algorithm to $O(n^3 \log n)$. The main effort is in computing the inverse FFTs since the FFTs of each projection histogram need only be computed once. In practice, we have observed that the exhaustive approach is rarely necessary, so the overall runtime is not noticeably affected.

3.5. Match Metrics

Our map matching algorithm is able to compute a transformation between any two maps, but we are only interested in retaining matches for which we have a high confidence that the maps actually overlap. Even though false matches will generally be discarded during the ICS phase, limiting the number of false matches resulting from the map matching process will save a significant amount of computation time. Therefore, it is important to assess the match quality from the histogram or entropy sequence correlations. The quality of a match can be measured by various metrics using the peak values from the four correlations performed (orientation histograms, entropy sequences, and the two projection histograms), or based on signal-to-noise ratios of the correlation sequences. These values can be combined into a single scalar metric by taking their summation or product, for example.



(a) Matrix of peak values of pairwise cross-correlations between two histogram projection sets.



(b) The matrix in (a) reordered such that angle *offsets* are represented on the horizontal axis.

Fig. 12. Matrices of peak values from all pairwise cross-correlations between the entire set of projection histograms from two maps. A strong response is evident for a rotational offset of approximately 100°. The mean of each column is the exhaustive correlation sequence and can be used in place of the orientation histogram or entropy sequence correlation to determine the rotational alignment between maps.

The signal-to-noise ratio, SNR, can be defined as follows:

$$\mathrm{SNR}(\mathbf{a}, \mathbf{b}) = \frac{\min\left(\max(\mathbf{a}/\|\mathbf{a}\|), \max(\mathbf{b}/\|\mathbf{b}\|)\right)^2}{2\left(1 - \eta(\mathbf{a}, \mathbf{b})\right)},$$

where $\eta(\mathbf{a}, \mathbf{b})$ is the cross-correlation coefficient between signals $\mathbf{a}$ and $\mathbf{b}$.

The ability of four different metrics to distinguish correct matches was assessed by computing receiver operating char-

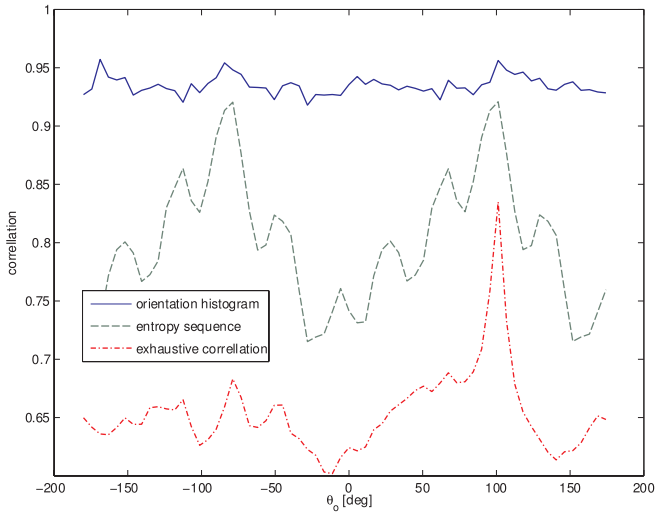


Fig. 13. A comparison of the orientation histogram, entropy sequence correlations, and exhaustive correlation sequence for the same pair of maps analyzed in Figure 12. Unlike the entropy sequence which is duplicated every 180°, the exhaustive correlation sequence does not repeat within the full 360° range.

acteristic (ROC) curves on a large data set of maps with known correspondences (Figure 14). ROC curves for a particular test or metric is a plot of the probability of detection against the probability of false alarm as the detection threshold is varied. The metrics considered here are the sum of the correlation peaks, the product of the correlation peaks, the sum of the signal-to-noise ratios, and the product of the signal-to-noise ratios (in all cases the sums and products include correlations between orientation histograms, entropy sequences, and the selected pairs of projection histograms). The ROC curves are also used to determine the optimal threshold for the metric which leads to a high probability of detection, $P(D)$, and a low probability of false alarm, $P(FA)$. The metric with the best performance is the sum of cross-correlation coefficients (Figure 14(a)). Future work will investigate other metrics that can take into account the saliency of a peak response.

The threshold on the quality metric for accepting matches is chosen empirically. The probabilities of detection, $P(D)$, and false alarms, $P(FA)$, for each threshold value are estimated by analyzing a large set of quality metrics of known correct and false matches, as in Figure 14. Using these probability functions, we have chosen a threshold value of 3.4 for the sum of the cross-correlation coefficients metric, which has a detection probability of 0.51 and a false alarm rate of less than 0.01. In practice, this low detection probability is often acceptable because the probability of missing multiple adjacent map matches is still quite low ($(1 - P(D))^n$). Generally, one might want to limit the frequency of false alarms based on an analysis of the computation cost of performing extra scan matching operations versus the functional cost of missing loop detections. Another strategy would be to maintain a priority queue of matches for further consideration prioritized by the match quality metric, then process the potential matches in an anytime fashion based on the available computational resources.

All transformations computed with valid quality metrics are subsequently verified (and improved) by the iterative scan matching alignment. Furthermore, any false matches from truly ambiguous environments are filtered by a *cycle verification* procedure described in Section 4.

3.6. Map Matching Experiments

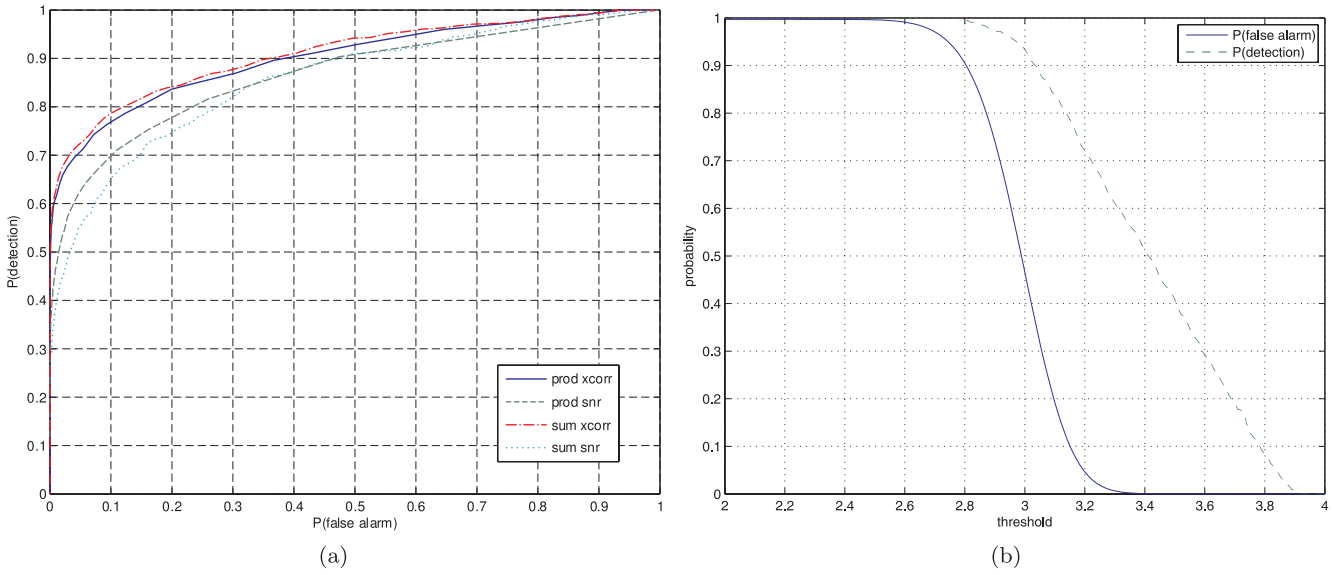
Figure 15 demonstrates the matching process for two maps generated during a driving experiment on suburban streets. The two maps shown in Figures 15(a) and 15(b) were created non-sequentially as the vehicle traversed through a location it had already visited (note that there is only minor overlap between the two maps). The entropy sequences and their correlation are displayed in Figure 15(c) (the orientation histograms did not produce any salient peaks) and the correlation of projection histograms taken along two orthogonal directions (suggested by the entropy sequence match) is shown in Figures 15(d) and 15(e). The two maps are shown aligned in Figure 15(f). That the algorithm was able to successfully match maps with minor overlap illustrates the versatility of the map matching approach.

4. SLAM Framework

The scan matching and map matching algorithms can be utilized as components of a more comprehensive SLAM system. In this section, we describe one such approach, the *Atlas* framework, which we use for our experiments in Section 4. We also discuss some of the finer-grained implementation details that allow the scan and map matching algorithms to perform fast enough for real-time online SLAM in large environments.

4.1. The Atlas Framework

Atlas is a SLAM framework in which existing small-scale mapping algorithms can be used to achieve real-time performance in large-scale, cyclic environments (Bosse et al. 2004). The approach does not maintain a single, global coordinate frame but rather an interconnected set of local coordinate frames (analogous to pages in an atlas). The representation consists of a graph of multiple local maps of limited size. Each vertex in the graph represents a local coordinate frame, and each edge represents the transformation between adjacent local frames. In each of these coordinate frames, a map is



(a) (b)

Fig. 14. (a) The probability of detection versus probability of false alarms receiver operating characteristic (ROC) curve for various match metric threshold settings. (b) The probability of detection and probability of false alarms versus the sum of cross-correlation coefficients.

built that captures the local environment and the current robot pose with associated uncertainties (called a *map frame*). Each map's uncertainties are modeled with respect to its own local coordinate frame. The uncertainty of the edges (adjacency transformations) in the *Atlas* graph are represented by a Gaussian random variable, and are derived from the output of the SLAM algorithm running in a local region. A limit is placed on the per-map computation by defining a measure of complexity for each map frame, which is not allowed to exceed a threshold (the map capacity). Rather than operating on a single map of ever-increasing complexity, the *Atlas* framework simply switches its focus to a new or *adjacent* map frame.

Atlas is intended to be a generic framework in which a variety of techniques could be used as the local mapping module. In our case, we use the scan matching algorithm (Section 2) to build up local maps, and the map matching algorithm (Section 3) to determine correspondences relating to loop closure. A typical global map construction works as follows (some details are omitted for brevity but can be found in a previous publication (Bosse et al. 2004)). Initially, the robot builds up a local map using the pose snapshot Kalman filter and the robust ICS scan matching algorithm. Once the number of saved poses exceeds the predefined limit, the current map is completed, and a new one is initiated from that location. When a map is completed, it can be checked against the other maps believed to be close to it (based on the current uncertainty) for possible loop closures using the histogram/entropy sequence-based map matching algorithm. If a match is detected with sufficient quality score, an *unverified* edge is placed between the two nodes in the *Atlas* graph indicating a hypothesized adjacency between the map frames. Unverified links can later be confirmed by the *Atlas cycle verification* process which uses small cycles completed by subsequent matches to validate the edge (Bosse 2004).

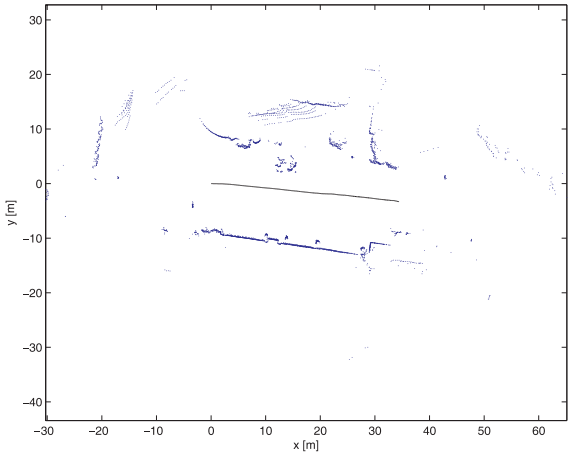
For visualization purposes, a post-processing step computes a global transformation to place each local map frame into a common reference which minimizes the errors for every adjacent link. Since this process only manipulates the coordinate frame and not the map data, it can be computed quite quickly (e.g., for an *Atlas* graph with over 650 nodes, the optimization takes only 270 ms on a 3.2 GHz desktop computer). The residual resulting from the optimization process provides a useful metric for evaluating the accuracy of the maps.

4.2. Efficient Implementations for Matching

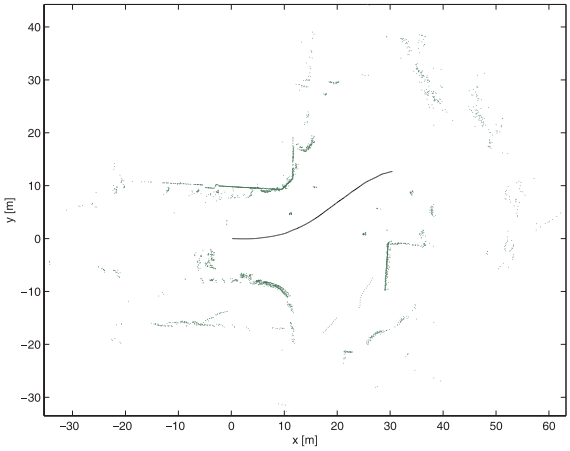
In this section, we focus on some of the implementation details that allow the scan and map matching algorithms to function as components of a real-time SLAM framework.

4.2.1. ICS Speedups

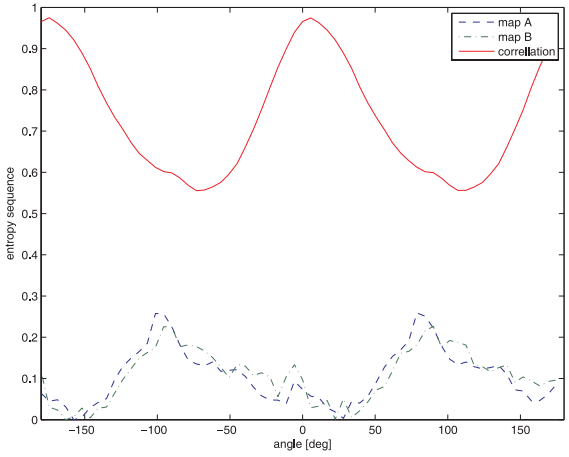
During map construction, scans are expected to have small relative displacements; therefore, the ICS algorithm should be able to converge quickly to the global maximum. Since the scan points are ordered by bearing, the search for the closest point can be sped up by limiting the search to the points that fall within the radius defined by the closest point found so far.



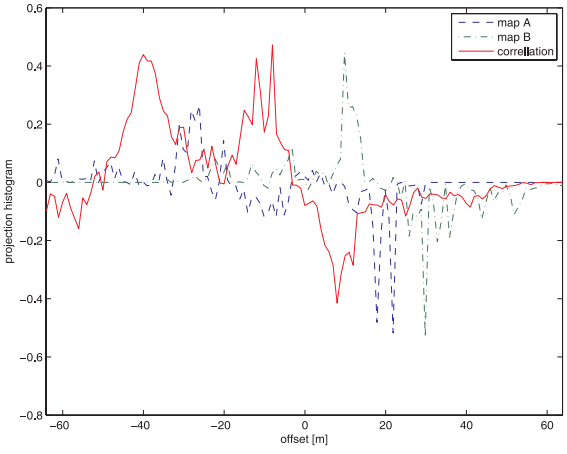
(a) Map A.



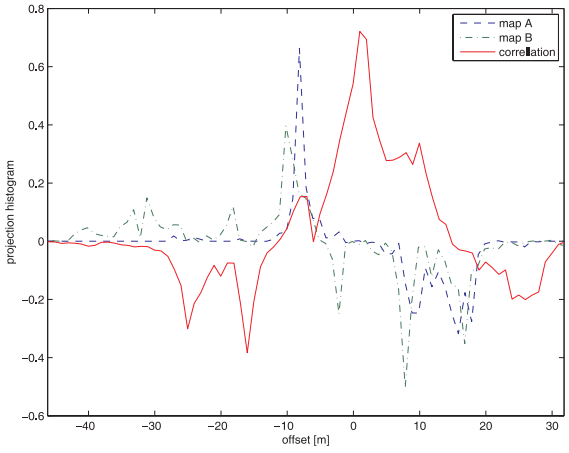
(b) Map B.



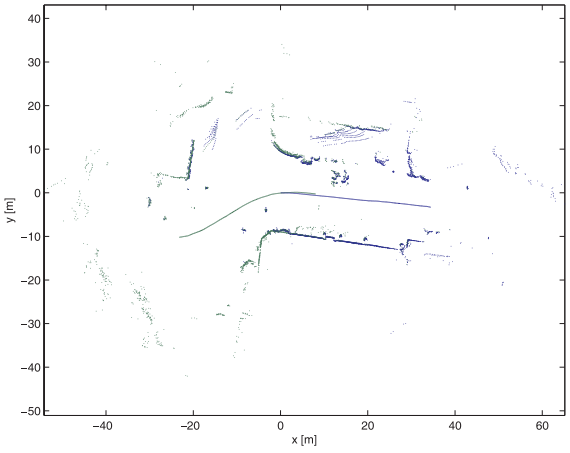
(c) Entropy sequences and their correlation.



(d) A set of projection histograms and their correlation.



(e) Orthogonal projection histograms and their correlation.



(f) The aligned maps.

Fig. 15. Matching of two non-sequential maps created at different parts of the traverse. (a), (b) Two example local maps, where the dots are the scan points and the solid line is the trajectory of the vehicle. (c) The entropy sequences and their correlation. (d), (e) The two orthogonal projection histograms used to score the match. (f) The maps brought into alignment using the peaks of the match.

This results in a constant computational time per scan point on average (i.e. linear in the number of scan points).

In the case of map matching, ICS is used to refine the solution from histogram matching. Initially ICS can be run on a low-resolution version of the local map to quickly validate the match before continuing with the full resolution map. To generate the low-resolution map, all points from each saved scan in the local map are transferred to the base coordinate frame, and subsequently down-sampled with the aid of a grid. The down-sampled map consists of the average of all the points (locations and normals) from the original scans that fall within each grid cell. Nearest-neighbor lookups are performed by directly indexing the corresponding four closest grid cells which are likely to contain a close match. If the corresponding grid cells are empty, the nearest non-empty cell can be precomputed by using the distance transform from filled cells on the grid.

Unlikely map matches can be quickly detected after the low-resolution ICS converges (or does not) by thresholding the percentage of overlap between the maps. It is often unnecessary to attempt the computationally intensive full-resolution match. If the high-resolution ICS is performed, the nearest-neighbor matching is also done with the assistance of a grid. In this case, grid cells contain a list of points within that cell rather than an average. The nearest-neighbor lookup is done only with the points within the closest four corresponding grid cells. The size of each grid cell is set to ensure that we find a neighbor as long as it is within a reasonable radius from the point (related to the width of the Cauchy distribution; typically three times $\bar{r}$).

4.2.2. Precomputation of Histograms

When a local map is completed, the orientation and projection histograms as well as the entropy sequences can be immediately computed and stored. Thus they are readily available for all future matching attempts and do not need to be recomputed for each map matching comparison involving that map.

4.3. Experiments

Experiments to demonstrate the effectiveness and versatility of our SLAM framework were performed in a variety of environments ranging from mostly structured urban and industrial, to mostly unstructured suburban and rural. As in Section 2.4 the data was gathered from two SICK LMS laser scanners mounted back-to-back on the roof of a sport utility vehicle, see Figure 16. No odometry or GPS sensors were utilized. To the authors' knowledge, the maps generated are the largest published SLAM maps with no known data association in terms of the total distance traversed (up to approximately 30 km),

Fig. 16. Test vehicle: two SICK LMS scanners are mounted back-to-back on the roof rack of a sport utility vehicle.

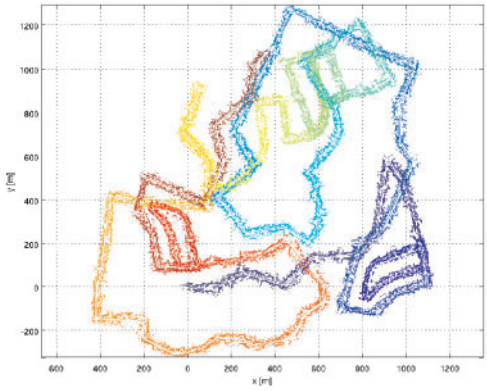
size of loops, and number of scans processed (millions). Furthermore, no SLAM algorithm has been demonstrated in such a wide spectrum of environmental structure.

In the Kenmore run (Figure 17) the vehicle was driven 17.8 km through suburban streets generating 581 local maps, each with 30 scans separated on average by approximately 2 m[1]. Figure 17(a) shows the open-loop map when no map matching is used. As is clear from the figure, the error in the laser odometry can grow unboundedly over distance traveled. If map matching is done without histogram or entropy sequence correlations (i.e. ICS uses the current map alignment prior computed from the shortest path through the *Atlas* graph as its initial guess), then it is not possible to detect the larger loops. This results in a poorly aligned map as shown in Figure 17(b). The closed loop map is depicted in Figure 17(c), where adjacent maps have been discovered via map matching. Figure 17(d) overlays the map onto an aerial photo, demonstrating the correctness of the map's topology. Here, we can see that the map is properly registered with loops correctly identified and closed.

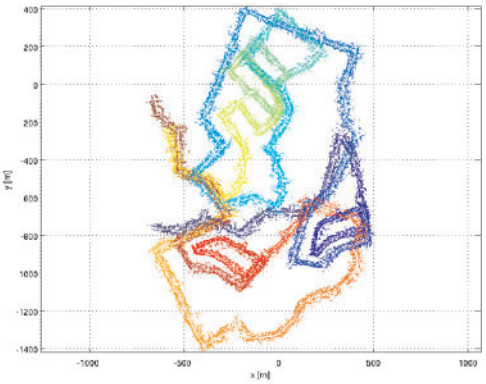
The adjacency structure of the *Atlas* graph for the Kenmore run, can be seen in Figure 18(a). The off-diagonal entries identify discovered loop closures. The same network structure is visible in the image of histogram match quality metrics, Figure 18(b).

The real-time nature of our mapping framework is illustrated in Figure 19(a) where the contributions from the map building and map matching processes are given. We observe that map building requires essentially a constant amount of computation time per measurement, whereas the map match-

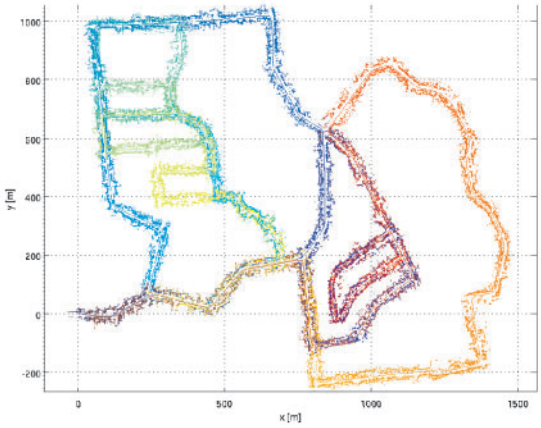
1. The data set from this run is available on RADISH `http://radish.sourceforge.net`



(a) Open-loop map (no map matching).



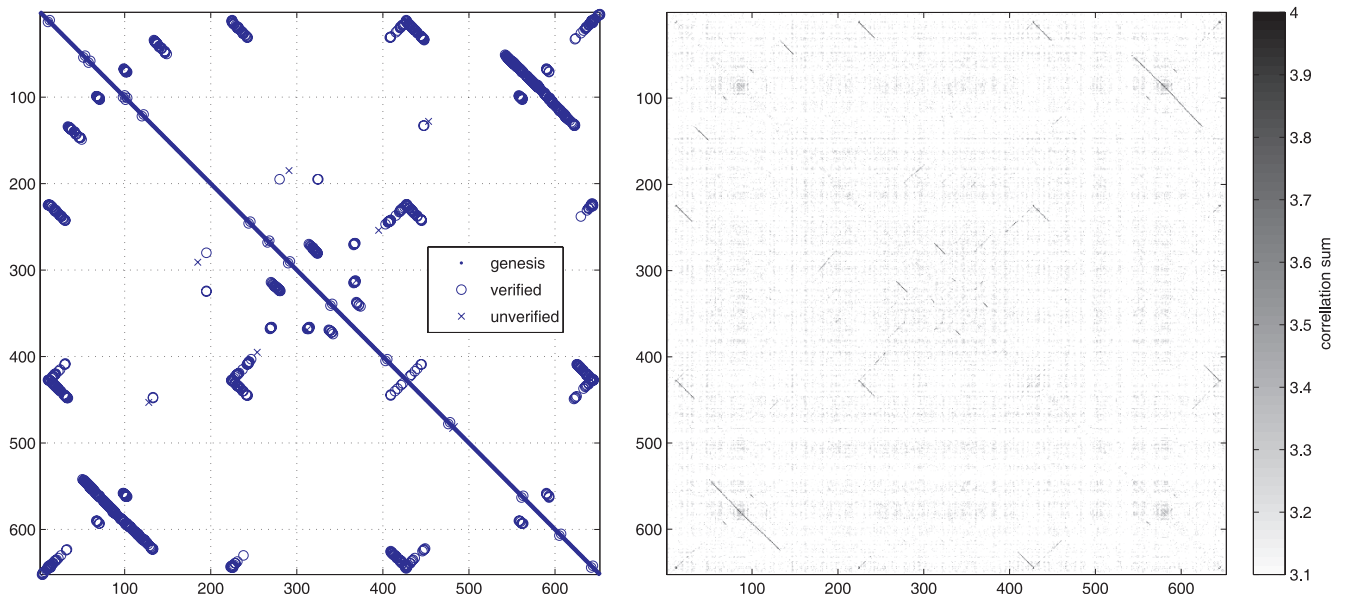
(b) Closed-loop map (without histograms).



(c) Closed-loop map (with histograms map matching).

(d) Map rotated and overlaid onto aerial image.

Fig. 17. Kenmore map. A map constructed from a 17.8 km traverse through suburban streets. (a) The map showing no loop closures (map matching turned off). (b) The result of attempting loop closures based on the current map alignment prior without the use of histogram correlation. (c) The optimized map including loop closures. (d) The map overlaid onto an aerial image.



(a) Adjacency graph. (b) Histogram correlation matrix.

Fig. 18. (a) The *Atlas* graph's adjacency matrix. Genesis edges indicate a link between maps that are created sequentially, unverified edges represent identified map matches that have not passed the cycle verification procedure. (b) The sum of the match correlations for each pair of maps. The repeated loops of the environment are indicated by the off-diagonal stripes.

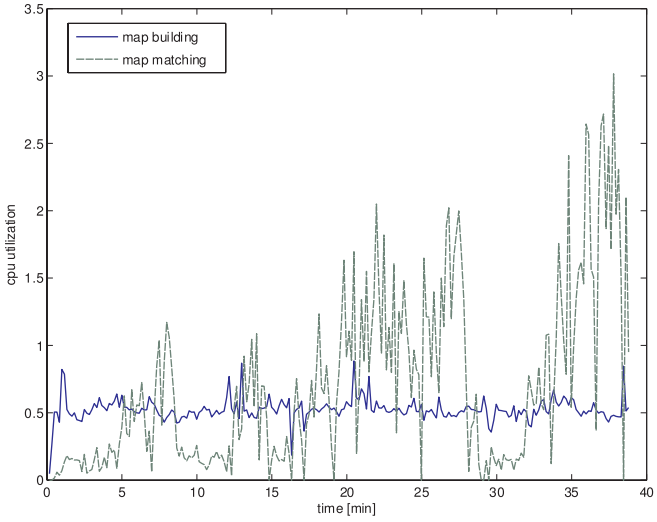
ing CPU utilization is highly dependent on the number of map matches considered (Figure 19(b)). This quantity is determined by the number of maps in the uncertainty region of the current map, which generally grows with distance traveled until a loop is closed. Hence the CPU utilization is dependent on the uncertainty since we must test each map in the region for a match. We are investigating development of sublinear methods for determining which maps should be considered for matching in order to remove the computational dependence on the uncertainty.

Additional maps created by our SLAM framework are shown in Figure 20. The variety of environments present in these maps demonstrates the versatility of our algorithm. The Brisbane central business district (CBD) run (Figure 20(a)) highlights an environment that is relatively structured, but contains many examples of dynamic elements, such as fleets of moving tour buses, non-stationary cars, and moving pedestrians. Not only are these actors moving while constructing the maps, but they may be present during some passes through a region and absent in others (e.g., parked vehicles that have moved). Dynamic features are likely to have low weights from the robust scan matching algorithm; thus, these outliers are easily suppressed from the local maps and do not influence the map matching process. Of course, if too large a proportion (roughly more than 50%) of the local environment is not static, then the mapping process can break down. However, the system degrades gracefully as the number of outliers increases.

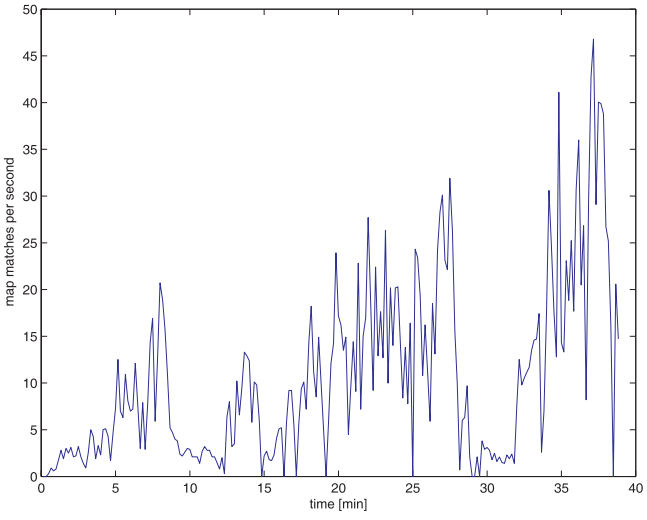
The Brookfield map (Figure 20(b)) is our largest map at 29.6 km of traverse. As this data was collected on rural roads, this map contains the most unstructured regions overall. As expected, many of the orientation histogram correlations were unsuitable for matching, necessitating the use of entropy sequences. The global map is comprised of 918 maps and approximately three million scan points. Figure 20(c) shows a 6.7 km traverse through suburban roads using a single side-mounted laser. In this run, 231 local maps were generated, with 15 scans per map spaced approximately 2 m apart. We are also able to solve the lost robot problem using this data, as described in an earlier publication (Bosse and Roberts 2007).

5. Conclusions and Future Work

In this article, we have presented reliable data association techniques for view-based SLAM enabling the generation of large-scale maps in unstructured outdoor environments. Experiments using the *Atlas* framework demonstrate the ability of our SLAM framework to build accurate maps with no odometry in a variety of real-world conditions including downtown city streets containing moving vehicles and pedestrians, as well as rural and suburban roads in which geometric structures such as flat surfaces are not always available. The maps generated are an order of magnitude larger than the largest published SLAM maps of real-world environments, and they raise



(a) CPU utilization.



(b) Map matches per second.

Fig. 19. Performance of the mapping framework for the Kenmore data set. (a) The number of seconds to process 1 s of incoming data for the map building process and the map matching process. (b) The average number of map matches per second. Note how the map matching CPU time is correlated with the number of map matches per second.

the bar for future outdoor SLAM research. Our largest map presented in this article has a path length of 29.6 km containing over 360,000 scans taken over 54 min at 56 Hz from two lasers. This data set amounted to 918 local maps of 30 laser scans each.

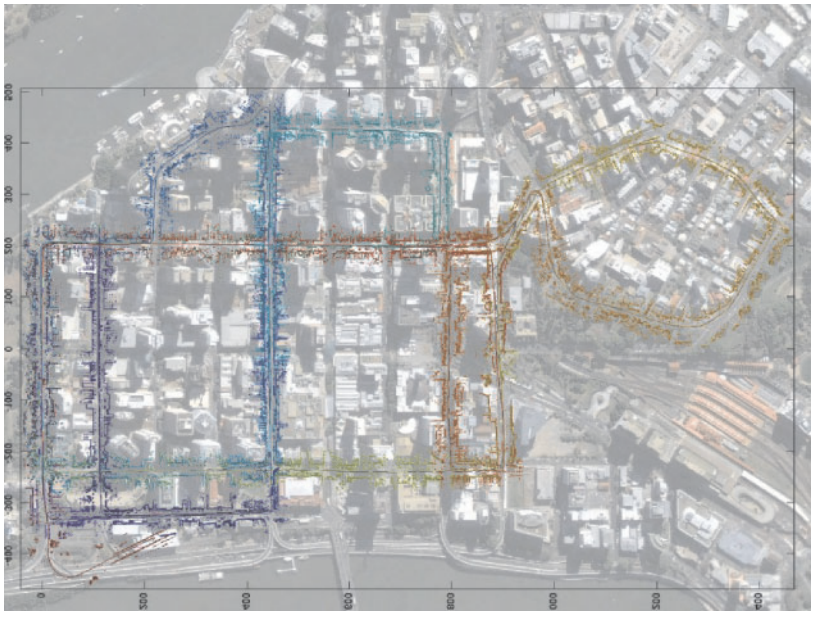
Novel data association techniques are required to achieve this result, both at the level of local map building and the global map matching. Local map building is achieved using an ICP-based scan matching algorithm taking into account surface normals and a robust weighting of scan surface points. Using robust optimization limits the detrimental effects of varying viewpoints, dynamic elements, and three-dimensional artifacts such as ground strikes in hilly terrain. The scan matching algorithm is incorporated into an EKF that includes state variables offset at a fixed time lag from earlier poses to reduce the growth of uncertainty arising from poor scan overlap and unexpected velocity changes.

At a more global scale, a map matching algorithm based on cross-correlation is able to align overlapping local maps without an initial guess of the registration. This technique is suitable for unstructured environments, where existing approaches based solely on orientation histograms fail. We have demonstrated that entropy sequences of projection histograms provide a reliable signal for rotational alignment in unstructured scenes.

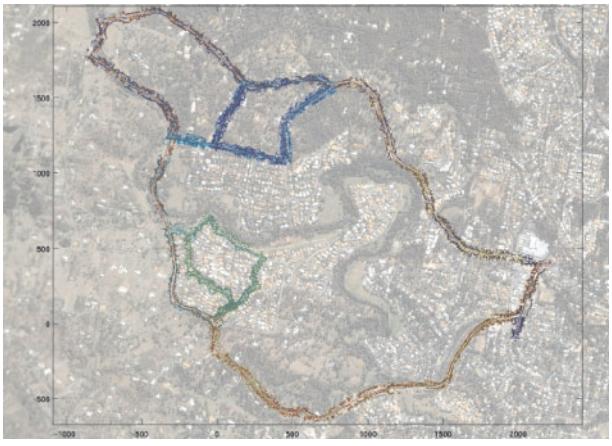
Further, an exhaustive correlations approach can be used in the event that the orientation histogram and entropy sequence both do not have a reliable signal. Although this is seldom the case when traveling on roads—since the road itself usually guarantees a sufficient signal—future work will investigate the utility of this approach in off-road environments.

Despite the success of our approach, there are still challenging situations requiring additional research. For example, when traveling at very high speeds, the amount of overlap between successive scans decreases, reducing the quality of the scan matches, or even failing to match in extreme cases. Likewise, in the case of extreme three-dimensional artifacts from hilly terrain or degenerate environments (such as bridges or smooth tunnels) map building will fail without help from additional sensors. As mentioned previously in Section 3, there is a graceful degradation as the number of outliers due to dynamic obstacles and ground strikes increases. When there are more scan points on moving objects (i.e. a fleet of buses surrounding the vehicle), the scan matching algorithm may confuse the inliers and outliers and corrupt the local map. Unexpected sensor outages also pose difficulties since they break the continuity of the mapping process, but can be rectified with global relocalization in applications where there are several repeated passes. The algorithm is also susceptible to missing loop closure events when attempting to close extremely large loops, since its estimate of the open-loop uncertainty can become over-confident due to unmodeled non-linear effects from the integrated rotation uncertainty. It is also possible for the uncertainty to encompass the entire map, resulting in exhaustive comparisons to detect map matches which can not be done in real-time.

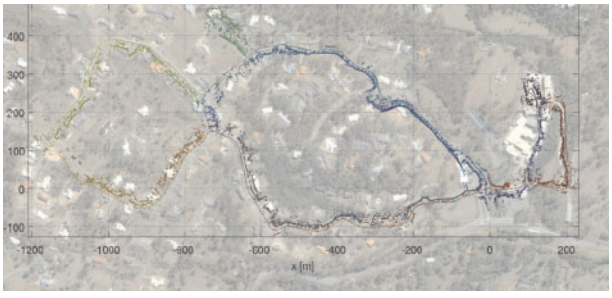
Current research focuses on solving the latter three problems using efficient methods for global relocalization, which can also be applied to the fusion of data sets from multiple runs or multiple robots. Further investigation will additionally consider extending our approach to three-dimensional laser maps and other map representations such as those generated by sonar, radar, or stereo vision.



(a) Brisbane CBD.



(b) Brookfield map.



(c) Pinjarra Hills map.

Fig. 20. Maps created using the *Atlas* SLAM framework overlayed onto aerial images. (a) Brisbane central business district (CBD). 12.7 km through downtown city streets. (b) Brookfield map. 29.6 km through rural roads. (c) Pinjarra Hills map. 6.7 km through suburban streets. This experiment used a single side-mounted SICK LMS laser providing 180° field of view centered to the left.

Acknowledgments

This work was funded by the CSIRO ICT Centre under the ROVER and Dependable Field Robotics projects. The authors would like to thank the Autonomous Systems Laboratory team for their support of this work. Special thanks go to Michael Brünig, Peter Corke, Paul Flick, Navid Nourani, Cédric Pradalier, Jonathan Roberts, Ashley Tews, Kane Usher, John Whitham, and Graeme Winstanley who all contributed to this work.

References

Bailey, T. (2002). *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. *Ph.D. Thesis*, The University of Sydney, Sydney, Australia.

Bar-Shalom, Y. and Fortmann, T. E. (1988). *Tracking and Data Association*. Academic Press.

Biber, P. and Straßer, W. (2003). The normal distributions transform: A new approach to laser scan matching. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). Simultaneous localization and map building in large-scale cyclic environments using the Atlas Framework. *International Journal of Robotics Research*, 23(12): 1113–1139.

Bosse, M. and Roberts, J. (2007). Histogram matching and global initialization for laser-only SLAM in large unstructured environments. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Bosse, M. C. (2004). *ATLAS: A Framework for Large Scale Automated Mapping and Localization*. *Ph.D. Thesis*, Massachusetts Institute of Technology.

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2724–2729.

Diosi, A. and Kleeman, L. (2005). Laser scan matching in polar coordinates with application to SLAM. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, pp. 3317–3322.

Eustice, R. M., Singh, H., and Leonard, J. J. (2006). Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6): 1100–1114.

Gutmann, J. and Konolige, K. (1999). Incremental mapping of large cyclic environments. *Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, California, pp. 318–325, URL citeseer.ist.psu.edu/gutmann00incremental.html.

John A. Volpe National Transportation Systems Center (2001). Vulnerability assessment of the transportation infrastructure relying on the global positioning system.

Leonard, J. J. and Rikoski, R. J. (2001). Incorporation of delayed decision making into stochastic mapping. *Lecture Notes in Control and Information Sciences: Experimental Robotics VII*. Berlin/Heidelberg, Springer, pp. 533–542.

Lu, F. and Milios, E. (1994). Robot pose estimation in unknown environments by matching 2D range scans. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, USA, pp. 935–938.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *International Joint Conference of Artificial Intelligence*.

Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2): 61–74.

Neira, J. and Tardós, J. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6): 890–897.

Neira, J., Tardós, J. D., and Castellanos, J. A. (2003). Linear time vehicle relocation in SLAM. *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, pp. 427–433.

Newman, P. (ed.) (2007). *Journal of Field Robotics Special Issue on SLAM in the Field*, volume 24(1/2).

Newman, P., Cole, D., and Ho, K. (2006). Outdoor SLAM using visual appearance and laser ranging. *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, Florida, pp. 1180–1187.

Pfister, S. T., Kriechbaum, K. L., Roumeliotis, S. I., and Burdick, J. W. (2002). A weighted range sensor matching algorithm for mobile robot displacement estimation. *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, D.C., USA.

Potmesil, M. (1983). Generating models for solid objects by matching 3D surface segements. *International Joint Conference of Artificial Intelligence*, pp. 1089–1093.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C*, 2nd edition, chapter 15. Cambridge, Cambridge University Press, pp. 699–706.

Pulli, K. (1999). Multiview registration for large data sets. *International Conference on 3D Digital Imaging and Modeling*, pp. 160–168.

Ramos, F. T., Nieto, J., and Durrant-White, H. (2007). Recognising and modelling landmarks to close loops in outdoor SLAM. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Rencken, W., Feiten, W., and Zöllner, R. (1998). Relocalisation by partial map matching. *Lecture Notes in Computer Science*, volume 1724, pp. 21–35.

Röfer, T. (2002). Using histogram correlation to create consistent laser scan maps. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp. 625–630.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, pp. 145–152.

Schultz, A. C. and Adams, W. (1998). Continuous localization using evidence grids. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2833–2839.

Silver, D., Ferguson, D., Morris, A., and Thayer, S. (2006). Topological exploration of subterranean environments. *Journal of Field Robotics Special Issue on Field and Service Robotics*, 23(6/7): 395–415.

Smith, R. and Cheeseman, P. (1987). On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4): 56–68.

Thrun, S., Burgard, W., and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Tomono, M. (2004). A scan matching method using euclidean invariant signature for global localization and map building. *Proceedings of the IEEE International Conference on Robotics and Automation*.

Wang, C.-C. and Thorpe, C. (2004). A hierarchical object based representation for simultaneous localization and mapping. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Weiss, G., Wetzler, C., and von Puttkamer, E. (1994). Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Munich, Germany, pp. 595–601.

Williams, S. and Mahon, I. (2004). Simultaneous localisation and mapping on the great barrier reef. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1771–1776.