

Scan-SLAM: Recursive Mapping and Localisation with Arbitrary-Shaped Landmarks

Tim Bailey and Juan Nieto

Abstract—Scan-SLAM is a simultaneous localisation and mapping algorithm that combines scan-matching methods with recursive estimation of landmark locations (using an EKF or other Bayesian filter). The scan-matching capability allows landmarks with arbitrary shapes to be modelled directly by sensed data and tracked within a conventional filter framework. This paper presents the essential Scan-SLAM algorithm, and implementation details for application with a scanning range-laser: segmentation, alignment, covariance estimation, data association, and landmark model augmentation.

I. INTRODUCTION

The earliest probabilistic versions of simultaneous localisation and mapping (SLAM) were built upon the extended Kalman filter (EKF) and tracked simple geometric features such as points or lines [28, 9]. Traditionally, landmarks conforming to these geometric models were referred to as “point targets”, (a term borrowed from the target-tracking literature), and a prevalent notion arose that the EKF formulation was only suited to environments with structure representable by these models. Furthermore, EKF-SLAM was seen as wasteful in discarding any data that did not correspond to an available model.

The Scan-SLAM algorithm [24, 25] was introduced as a response to this notion. We argue that the term “point target” is a misnomer in this context insofar as it does not refer to the shape or geometry of the landmark. Rather it should be understood as referring to the point *location* of the stationary landmark. Thus, EKF-SLAM is applicable to arbitrary shape models, and the estimation of landmark pose is separate from the definition of landmark shape. In particular, these models are not confined to geometric shapes like circles or lines, and Scan-SLAM proposes the generation of shape models directly from the sensed data. (It is worth noting that, while Scan-SLAM is presented here using the EKF, it is equally suited to other estimation algorithms, such as the information filter or particle filters. The key point is that landmark pose estimation and shape modelling are treated separately.)

SLAM based on scan matching is not new, with early work on the alignment of range-laser scans for map building now more than ten years old [19, 20]. The consistent pose estimation (CPE) algorithm [14, 17] builds directly on this work. More recently, scan-matching SLAM has been implemented using particle filters [16, 11, 13] and sparse information-form estimation [12, 8]. These methods have all successfully built large-scale maps in a variety of challenging environments. What these approaches have in common is that they all estimate the entire vehicle trajectory, rather than just its latest momentary pose, and they maintain a history of raw data scans. Essentially, their estimation process is trajectory-centric rather than landmark-centric as in traditional EKF-SLAM.

Scan-SLAM presents a landmark-centric version of scan-matching SLAM. Is it better than existing scan-matching methods? The short answer is no. This approach may be seen as alternative to, rather than superior to, trajectory-oriented SLAM. Each has tradeoffs in terms of implementation complexity, computational efficiency, and mapping accuracy and reliability. However, both paradigms have much in common. All scan-matching methods involve scan segmentation, registration, estimation of alignment uncertainty, and alignment validation. These common issues will be the focus of this paper.

The scope of this paper is to address issues related to SLAM in the small to medium scale. Large scale implementations rely on additional mechanisms to mitigate computational cost and nonlinearities [1, 4], and to facilitate the closing of large loops [1, 22, 23, 5]. These mechanisms are directly applicable to scan-matching SLAM, but are not examined here. Another topic beyond the scope of this paper is the important issue of map management. Any industrial-strength SLAM implementation will track tentative landmarks for a period before fusing them into the map, and remove existing landmarks that have become unreliable or disappear. These, and other map housekeeping issues, are not discussed further.

The following section presents the essential form of the Scan-SLAM algorithm. Section III discusses the properties of several alternatives for representing

shape models. Section IV discusses segmentation of laser scans. Section V addresses scan alignment and Section VI presents algorithms for estimating alignment covariance. The next section examines data association in regard to searching for candidate landmarks and validating alignment. Section VIII discusses the evolution of landmark shape models with successive observations, and the final section sums up with some concluding remarks.

II. ESSENCE OF SCAN-SLAM

The essential Scan-SLAM algorithm differs from traditional feature-based EKF-SLAM only in the way it generates its observation model. An EKF requires an observation model of the general form

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{r}, \quad (1)$$

where the measurement \mathbf{z} is a function of the state \mathbf{x} , and \mathbf{r} is zero-mean measurement noise with covariance \mathbf{R} . A typical conventional observation model for laser data is a range-bearing model,

$$\mathbf{z} = \begin{bmatrix} \sqrt{(x_L - x_v)^2 + (y_L - y_v)^2} \\ \arctan \frac{y_L - y_v}{x_L - x_v} - \phi_v \end{bmatrix} + \mathbf{r}. \quad (2)$$

However, when observing a landmark, a scanning laser never directly returns range-bearing of the landmark location. Rather it returns a series of measurements of the landmark surface, and this is converted to a location estimate using a shape model. For example, suppose the object were a tree trunk, then one would segment out the relevant laser measurements, fit a circle to these points, defining the landmark location as the circle centre, and estimate the observation uncertainty from the raw measurement uncertainties and the circle model. This then forms the observation according to (2). For more details on this example, see [1, Section 3.4.1].

Scan-SLAM does basically the same thing, but without the predefined geometric model of a circle (or line or point, etc.). Instead, the shape model is composed of raw sensor measurements, and the observation model is obtained by aligning this model with subsequent laser scans. Having generated an observation of the form $\{\mathbf{z}, \mathbf{R}\}$, the remaining algorithm—vehicle pose prediction, observation fusion, adding new landmarks—is identical to conventional SLAM.

The essential Scan-SLAM framework, therefore, is as follows. The SLAM state $\mathbf{x} = [\mathbf{x}_v, \mathbf{x}_1, \dots, \mathbf{x}_N]^T$ consists of the vehicle pose \mathbf{x}_v and the global pose of landmark coordinate frames \mathbf{x}_i as shown in Fig. 1(a). Each landmark estimate is a local coordinate frame, and the landmark’s shape model is defined in these local

coordinates as in Fig. 1(b). The shape information is separate from and auxiliary to the SLAM estimation process. On obtaining a new scan, the predicted location of the landmark relative to the vehicle is obtained by projecting the global landmark frame estimate into the vehicle coordinate frame, as in Fig. 1(c), and, in turn, projecting the shape model into the vehicle coordinate frame. This gives an initial prediction for scan-matching, aligning the shape model with the scan and computing the alignment covariance, as shown in Fig. 1(d).

It is important to realise that alignment is observation generation, analogous to the alignment with a circle model mentioned above. It is not a filter operation; not a prediction step followed by an update step. The prediction of the landmark frame in vehicle coordinates is merely a seed for the alignment process, typically necessary to achieve a basin of convergence and avoid local minima.

The form of the observation model generated by scan alignment is the pose of the landmark frame as seen from the coordinate frame of the vehicle,

$$\begin{aligned} \mathbf{z} &= \mathbf{x}_L^v \\ &= \mathbf{F}(\mathbf{x}_L^G - \mathbf{x}_v^G) + \mathbf{r}, \end{aligned} \quad (3)$$

where the superscript denotes the base coordinate frame, G for global coordinates and v for vehicle coordinates, and the rotation matrix is given by¹

$$\mathbf{F} = \begin{bmatrix} \cos \phi_v^G & \sin \phi_v^G & 0 \\ -\sin \phi_v^G & \cos \phi_v^G & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

III. MODEL REPRESENTATION

The landmark shape model is represented in a format suited to scan-matching. Three such representations are Gaussian mixture models (GMMs), occupancy grids, and raw points. The first two perform alignment by cross-correlation, the latter by a process called iterative closest points (ICP) [3].

GMMs are weighted sums of Gaussian functions, and are applied to modelling laser scans in [1, Section 4.4]. This representation permits very accurate scan alignment, more accurate than ICP since it does not involve point-to-point correspondence assumptions. However, a straightforward implementation of GMM cross-correlation is expensive and, while efficient forms are possible (see [1, Section 4.4.4]), a quick and nearly-as-accurate solution based on ICP is simpler.

Occupancy (or evidence) grids convert a range-image into a probabilistic estimate of occupied space, where

¹Rotation matrices are usually denoted \mathbf{R} but we are already using this variable to denote observation uncertainty.

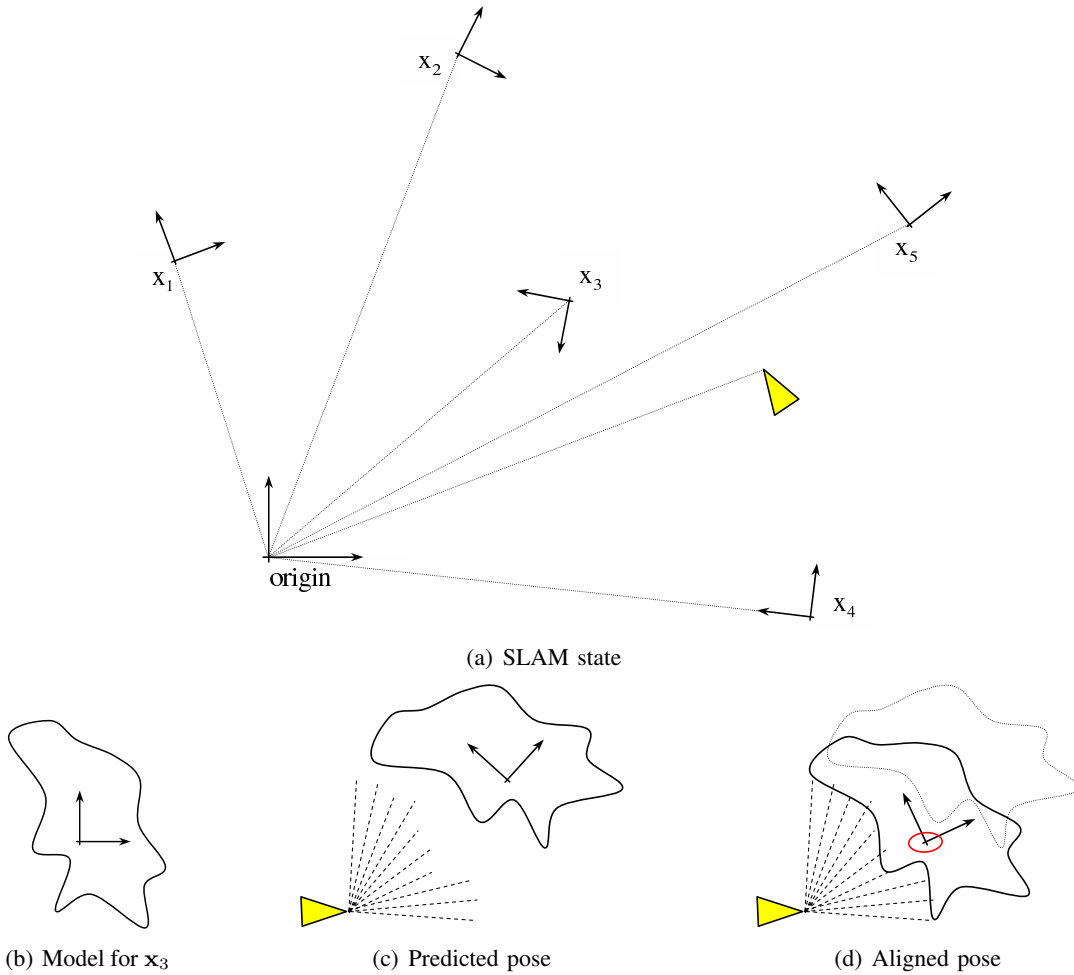


Fig. 1. The essential Scan-SLAM algorithm. The SLAM state (a) is composed of the vehicle pose and the global pose of landmark coordinate frames. For each landmark frame, there is an auxiliary shape model (b), which is maintained in the local coordinate system of the landmark frame. This embedded frame defines the “point *location*” of the landmark. When the vehicle obtains a new scan, (of landmark x_3 in this example), alignment is commenced from the predicted location $\hat{\mathbf{x}}_3^v$ in vehicle coordinates (c) and, after alignment, the resultant relative pose and estimated uncertainty forms the new observation (d).

the continuous space is approximated by regular discrete cells [10, 15, 27, 18]. Grids offer a superior representation, at least in principle, because they permit explicit representation of occupied space (about the endpoint of a range measurement), free-space (along the measurement ray) and unknown space (unobserved regions). This has the potential to provide more informative cross-correlation than the GMM method above or ICP, which only model occupied space. However, grids are also expensive, and there is a tradeoff between correlation accuracy and grid-cell size. They are particularly expensive for 3-D range images. Nevertheless, 2-D grids are popular for scan-matching SLAM [14, 16, 11, 13], and several efficient implementations are available as open-source at <http://openslam.org>.

Models composed of unprocessed range-image data-points may be aligned using ICP [3, 20]. The basic ICP algorithm is very simple. Let a scan of reference

points be denoted $\mathbf{p}^r = \{\mathbf{p}_1^r, \dots, \mathbf{p}_N^r\}$ in coordinate frame r and a second scan of points be denoted $\mathbf{q}^s = \{\mathbf{q}_1^s, \dots, \mathbf{q}_M^s\}$ in coordinate frame s . Given an initial estimate $\hat{\mathbf{x}}_s^r = [\hat{x}_s^r, \hat{y}_s^r, \hat{\phi}_s^r]^T$ of the pose of frame s with respect to frame r , the following three operations are performed iteratively: (i) transform the points \mathbf{q}^s into the frame of r ,

$$\hat{\mathbf{q}}_i^r = \begin{bmatrix} \hat{x}_s^r \\ \hat{y}_s^r \end{bmatrix} + \begin{bmatrix} \cos \hat{\phi}_s^r & -\sin \hat{\phi}_s^r \\ \sin \hat{\phi}_s^r & \cos \hat{\phi}_s^r \end{bmatrix} \mathbf{q}_i^s, \quad (4)$$

(ii) for each $\hat{\mathbf{q}}_i^r \in \hat{\mathbf{q}}^r$, find its nearest neighbour $\mathbf{p}_j^r \in \mathbf{p}^r$, and (iii) compute a new estimate of $\hat{\mathbf{x}}_s^r$ so as to minimise an error function $f(\{\mathbf{p}_j^r, \mathbf{q}_i^s\}_{i=1}^M, \hat{\mathbf{x}}_s^r)$. These three operations may be summarised by a short MATLAB function, wherein each set of points is stored as a matrix of column vectors. (Note, for least squares, the error function minimisation is implicit in the `transform_points` function.)

```

function x = icp(p,q,x,N)
for i = 1:N
    qr = transform_frame(q,x);
    j = nearest_neighbours(p,qr);
    x = transform_points(p(:,j),q);
end

```

Typically the nearest neighbour search is based on Euclidean distance between points, and the subsequent alignment transformation is an unweighted least-squares operation. This form carries the implicit assumption that the point location uncertainties are equal and isotropic, which is generally not the case, but this choice of error function has the distinct advantage of a cheap closed-form (non-iterative) solution [20, Appendix C]. A MATLAB implementation of the alignment algorithm is as follows.

```

function x = transform_points(p,q)
P = cov([q;p]',1);
S = P(1:2,3:4);
phi = atan2(S(1,2)-S(2,1), ...
            S(1,1)+S(2,2));
R = [cos(phi) -sin(phi);
     sin(phi)  cos(phi)];
xy = mean(p,2) - R*mean(q,2);
x = [xy; phi];

```

There are two significant problems with the simple ICP algorithm. First, the point-to-point correspondence between the two scans introduces an additional error into the relative pose estimate [26]. This may be alleviated somewhat by approximating a continuous surface from the reference scan [7, 29]. Surface approximation also tends to improve alignment when the density of points in the reference scan is unevenly spaced. A cheap but effective approximation is accomplished by a simple interpolation scheme, whereby for each $\hat{\mathbf{q}}_i^r \in \hat{\mathbf{Q}}^r$, the nearest neighbour \mathbf{p}_j^r is found, and two line segments are formed joining \mathbf{p}_j^r to its adjacent points \mathbf{p}_{j-1}^r and \mathbf{p}_{j+1}^r . (We presume the points \mathbf{p}^r are ordered along the surface of the reference scan, thereby forming a poly-line surface approximation.) The point correspondence then is computed as the closest point to $\hat{\mathbf{q}}_i^r$ on these two segments. An example of interpolated correspondence is shown in Fig. 2.

The second problem with the above ICP algorithm is the appearance of *outlier* points in the second scan. These may be due to viewpoint variation, dynamic objects, clutter, or sensor anomalies. A simple solution is to apply a distance threshold G to gate out correspondences where $\|\mathbf{p}_j^r - \hat{\mathbf{q}}_i^r\| > G$.

For the experiments performed in this paper, raw

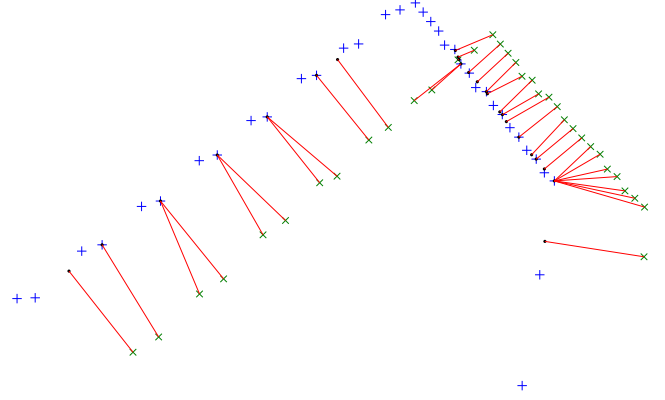


Fig. 2. Interpolated correspondence between two scans. For each \times , the nearest $+$ is found, and two segments are drawn connecting this $+$ to its adjacent $+$'s. (Of course, only a single segment is formed for an end-point $+$.) The correspondent point is computed as the closest point on these segments to the \times .

data points and an interpolating ICP algorithm are used to represent and align the landmark shape models. A graduated gating scheme is applied, with a 0.5 metre gate in the first iteration, reduced after each subsequent iteration until the final iteration has a gate of 0.05 metres.

IV. SCAN SEGMENTATION

Segmentation is the process of splitting a scan into several coherent clusters. For Scan-SLAM, segmentation is necessary to define new shape models or to augment existing models. The choice of segmentation method is rather arbitrary and will be dependent on other design choices such as the alignment and covariance estimation strategies.

One option is to consider measurements in angle-order and segment point clusters at range discontinuities as shown in Fig. 3(a). The problem here is that some segments are very linear and may be difficult to align properly. Either one chooses to discard these sort of segments or devises alignment and covariance estimation methods that can generate reliable results. Combining several orthogonal segments together is another option.

An alternative segmentation strategy is to place model origins at the centre of point clusters and associate all points within a certain radius of the cluster to that model, as in Fig. 3(b). Clearly there are many possible segmentation schemes.

Fig. 4 shows a map created with many small models. Here the segments for model creation are generated using the approach in Fig. 3(a), but model augmentation (see Section VIII) is performed so that new points are added to a model only if they lie within the model's original

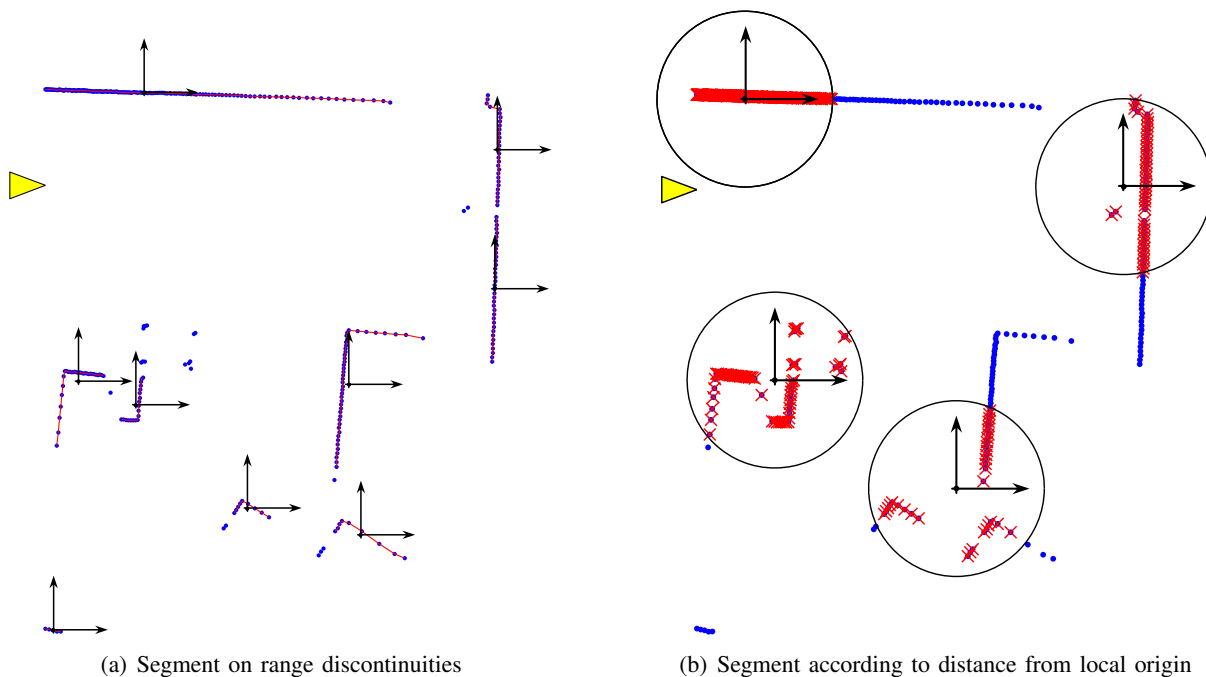


Fig. 3. Two of many possible segmentation strategies. Segmentation is required for two operations: creation of new models, and augmentation of existing models. The segmentation method will determine the size, distribution and evolution of the models.

segment bounds. The result is many small models, with different parts of the surface of a single physical object being represented by multiple models. On the other hand, Fig. 5 treats the whole scan as a single segment, in which case Scan-SLAM becomes equivalent to trajectory-oriented SLAM.

V. SCAN ALIGNMENT

When a new range-scan is obtained, models are aligned with the scan according to the process described in Section II. Each candidate model is aligned against the *entire* scan; the new scan is not segmented prior to alignment.

Given an ideal correlation function, and unlimited computation, scan alignment and subsequent estimation of uncertainty (see Section VI) should produce similar results regardless of the choice of segmentation strategy. However, in practice, smaller landmarks tend to be less reliable than models defined by whole scans. The problem is that smaller models contain less alignment information, and so will tend to give larger less-Gaussian alignment uncertainties. Furthermore, for efficiency, alignment and covariance estimation are typically performed as two separate operations, not as a full Bayesian solution. First, the model and scan are given a maximum-likelihood alignment (using ICP, say), and then covariance is computed by a local approximation about the maximum. Of particular concern are models,

such as lines, whose alignment is insufficiently constrained. While the alignment uncertainty of such models should be captured by the covariance estimate, it is difficult to obtain an accurate covariance approximation when the uncertainty is very large or if the alignment falls into a local minima. Also, approximation errors from multiple small models will tend to compound. One solution might be to reject all models that do not generate a confident pose alignment (determined, perhaps, by analysis of the determinant or trace of the alignment covariance estimate).

Small models are least reliable if they are aligned individually, since a single local minima may irrevocably skew the entire SLAM estimate. A better, though more complicated and expensive, implementation would align groups of models simultaneously, accounting for their joint cross-correlations. Therefore, while smaller sub-scan shape models can, in principle, be as good as matching whole scans, Scan-SLAM will generally be simpler to implement, and more accurate and reliable with larger models.

A separate practical issue encountered with any scan alignment algorithm is the problem of visible surfaces. When approaching an object from one direction, the vehicle will observe the surface of the side facing it. If the vehicle later approaches the object from another direction, observing the opposite side, there is the potential for scan-matching to inadvertently align the opposing surfaces. This is particularly so for office walls (see

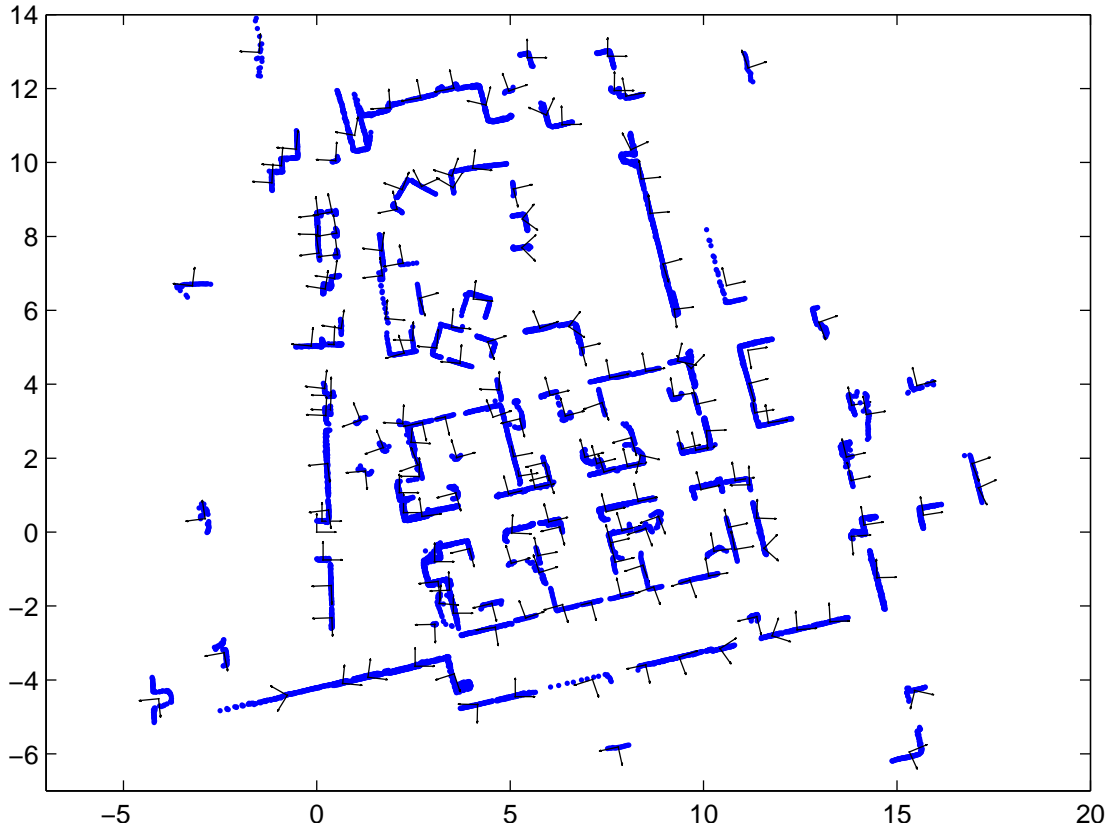


Fig. 4. Scan-SLAM map composed of many small models. Models are created with data segmented at range discontinuities. Models are augmented with points that fall within the region of the original model surface.

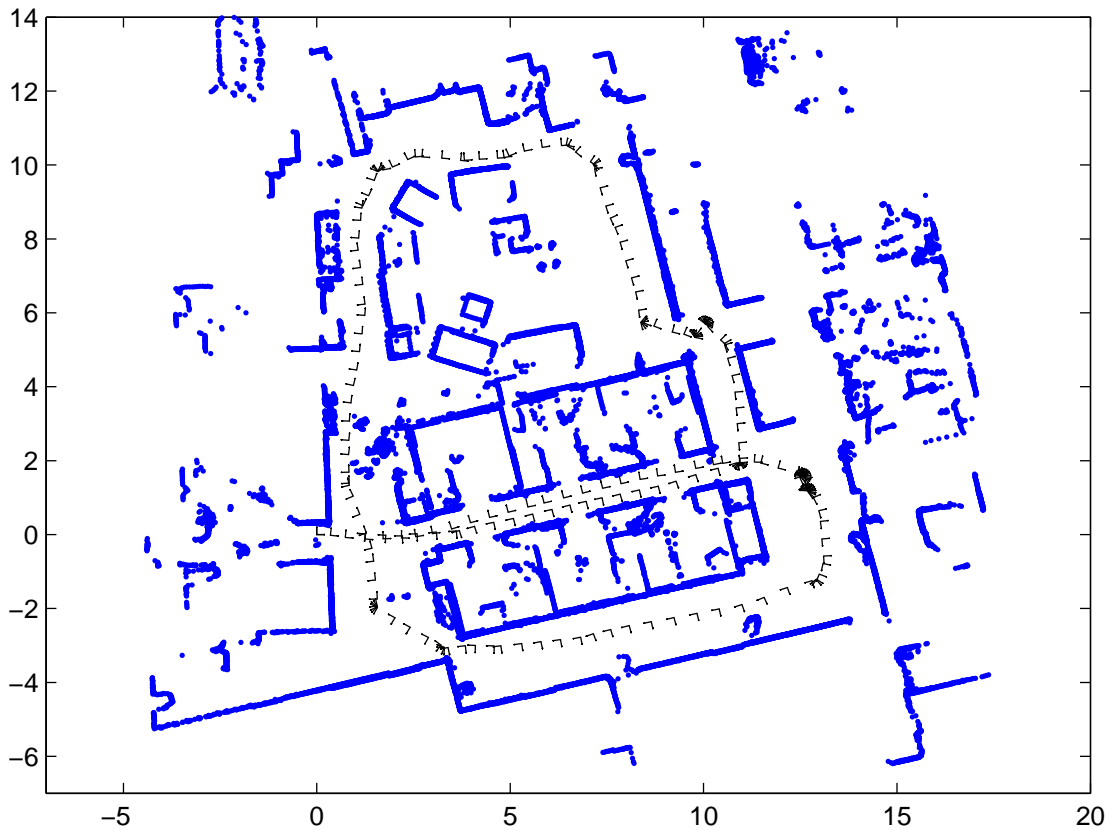


Fig. 5. Scan-SLAM map composed of full-scan models. The model coordinate frames are the vehicle pose estimates.

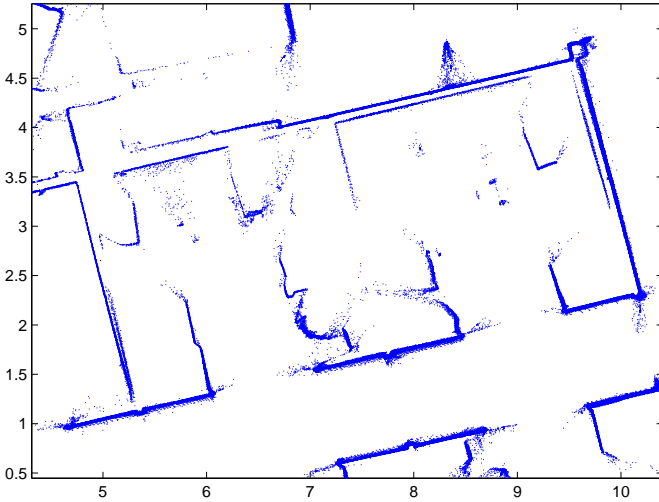


Fig. 6. Closeup view of the office environment from Fig. 5. The scan matched map shows surfaces from both sides of cubical walls. Point angle order can be used here to help avoid misalignment.

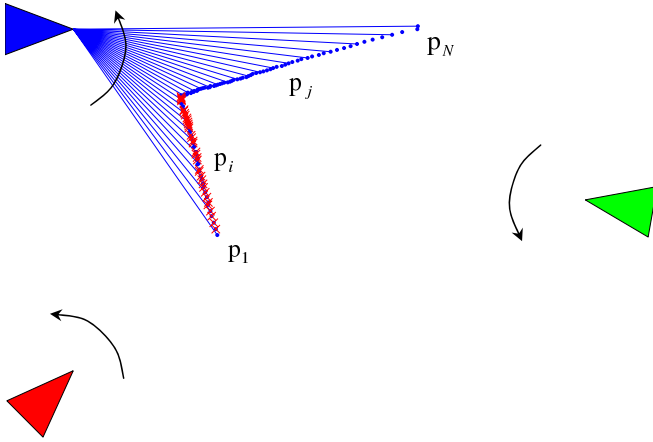


Fig. 7. Angle ordered points for visible surface discrimination. The original scan is obtained from the blue vehicle (top-left), and stored in counter-clockwise order. From the pose of the red vehicle (bottom-left), the algorithm predicts that only the marked points (red \times 's) are visible. From the pose of the green vehicle (right), all points are predicted to lie on a non-visible surface.

Fig. 6) and rectilinear objects (boxes, parked cars, etc.), whose opposite sides have similar shape. A solution is to record the angle-order of scan-points in the coordinate frame from which the scan was taken.

Consider the scenario in Fig. 7. The blue vehicle obtains a scan of an object, and stores the points $\mathbf{p}^b = \{\mathbf{p}_1^b, \dots, \mathbf{p}_N^b\}$ in counter-clockwise² order,

$$\theta_i^b < \theta_j^b, \quad \forall i < j, \quad (5)$$

where $\theta_i^b = \arctan \frac{y_i^b}{x_i^b}$. Suppose, at some later time, the vehicle moves to the pose of the red vehicle, with estimated pose $\hat{\mathbf{x}}_r^b = [\hat{x}_r^b, \hat{y}_r^b, \hat{\phi}_r^b]^T$ relative to the blue

²The points might equally have been stored in clockwise order, but counter-clockwise is chosen because it is the ordering of common scanning lasers.

vehicle. The blue vehicle scan is transformed to the coordinate frame of the red vehicle as³

$$\hat{\mathbf{p}}_i^r = \begin{bmatrix} \cos \hat{\phi}_r^b & \sin \hat{\phi}_r^b \\ -\sin \hat{\phi}_r^b & \cos \hat{\phi}_r^b \end{bmatrix} \left(\mathbf{p}_i^b - \begin{bmatrix} \hat{x}_r^b \\ \hat{y}_r^b \end{bmatrix} \right). \quad (6)$$

The angle of these points $\theta_i^r = \arctan \frac{y_i^r}{x_i^r}$ indicates whether the surface is facing towards $\theta_i^r < \theta_j^r$ or away from $\theta_j^r < \theta_i^r$ the new observer. In practice, it is sufficient to compare adjacent points, such that $\hat{\mathbf{p}}_i^r$ is considered a facing surface point if $\theta_{i-1}^r < \theta_i^r < \theta_{i+1}^r$.

While this heuristic is deterministic, since it does not consider the uncertainty of $\hat{\mathbf{x}}_r^b$ or of the measurements, it gives a good indication of non-visible surfaces when recomputed after each ICP iteration.

VI. COMPUTING AN APPROXIMATE COVARIANCE

The ideal covariance obtained from scan alignment is the second moment of the correlation likelihood function. In computing a covariance estimate, there are three levels of approximation. First, the likelihood function itself is approximate. A grid-based representation may approach the optimal likelihood function using cross-correlation, as the grid-cells become infinitesimal. An ICP-based likelihood function, on the other hand, does not capture information about free-space, and also introduces an artificial point-to-point correspondence. The second level of approximation is that the covariance represents only the second moment of the likelihood function, which may be a poor approximation if the likelihood is very non-Gaussian. The third approximation is the covariance estimate, since computing an accurate second moment can be expensive. We address these three forms of approximation in turn.

A. Scan Correlation Likelihood Function

For the ICP algorithm with isotropic point uncertainty, linear interpolation and gating, the likelihood function for the pose of frame s relative to frame r is,

$$\Lambda(\mathbf{x}_s^r) = \prod_i^A \mathcal{N}(\mathbf{p}_j^r - \mathbf{q}_i^r, 2\mathbf{P}) \prod_i^B \mathcal{N}(G, 2\sigma^2), \quad (7)$$


where \mathbf{q}_i^r is the scan point projected according to (4), \mathbf{p}_j^r is the corresponding point interpolated from the reference scan, $\mathbf{P} = \sigma^2\mathbf{I}$ is the isotropic point uncertainty, and G is the gate distance threshold. Of the total points in $\{\mathbf{q}_i^r\}_{i=1}^M$, there are A points with corresponding neighbours and $B = M - A$ points gated out. More

³Notice that this equation is the same as (3), but for points not poses, and is the inverse operation to (4).

numerically stable computation is possible using the log-likelihood,

$$\log \Lambda(\mathbf{x}_s^r) = \sum_i^A (C_1 - \nu_i^T \mathbf{P}^{-1} \nu_i) + \sum_i^B \left(C_2 - \frac{G^2}{\sigma^2} \right), \quad (8)$$

where $\nu_i = \mathbf{p}_j^r - \mathbf{q}_i^r$. Since it is only necessary to evaluate a likelihood function to proportionality, the constant offsets C_1 and C_2 can be ignored in the log-likelihood,



$$\begin{aligned} \log \Lambda(\mathbf{x}_s^r) &= - \sum_i^A (\nu_i^T \mathbf{P}^{-1} \nu_i) - \frac{2BG^2}{\sigma^2} \\ &= - \frac{1}{\sigma^2} \left(\sum_i^A \nu_i^T \nu_i + BG^2 \right) \end{aligned} \quad (9)$$

The full likelihood function algorithm should also incorporate the assignment of point-to-point correspondences as in the following MATLAB function.

```
function L = loglikelihood(p, q, x, G, S)
qr = transform_frame(q, x);
[pj, i] = neighbour_interp(p, qr, G);
B = size(qr, 2) - length(i);
v = pj - qr(:, i);
D = sum(dot(v, v, 1));
L = -(D + B*G^2) / S;
```

Different likelihood functions presented in the literature account for unequal and non-isotropic sensor noise, and give a more realistic projection of the sensor's range-bearing uncertainty into Cartesian space [26, 6]. However, for a high accuracy sensor like a scanning laser, the sensor uncertainty modelling is less critical than accounting for correspondence uncertainty. In the above algorithm, the reevaluation of correspondence each time the function is called with a different \mathbf{x}_s^r incorporates this aspect. This likelihood function generates reasonable covariance estimates even though the point uncertainty model is isotropic. A better likelihood function would incorporate both sensor uncertainty and correspondence uncertainty.

B. Moment Approximation of the Likelihood Function

Approximating a non-Gaussian likelihood function by its first two moments is reasonable when there is plenty of data in both scans, allowing unambiguous maximum-likelihood pose alignment. When the true uncertainty is compact, it is usually well represented by a Gaussian. On the other hand, when there is little overlap between scans or lack of orthogonal constraints, the likelihood function can have quite non-Gaussian shape and be ill-represented by its second moment. The consequences of

this problem will depend on the covariance estimation method; if the estimate is able to reliably detect very uncertain alignments, then they can be safely rejected and will not cause further damage. If however, the estimate merely underestimates uncertainty, the resultant observation model can spoil the entire SLAM estimate.

C. Estimating Covariance

Laplace's method [21, Chapter 27] is a popular technique for approximating a function by a Gaussian. Several variations of this approach have been presented for scan-matching covariance estimation [20, 26, 2, 24, 6]. The basic implementation is to first perform maximum-likelihood alignment, and then compute the second-derivative (i.e., the curvature or Hessian) of the likelihood function about the max-likelihood point.

One problem with using the Hessian is that it does not naturally account for uncertainty in correspondence, and an analytical solution will compute the local curvature assuming the max-likelihood point-to-point associations are correct. A better solution is to approximate the Hessian numerically, using samples drawn sufficiently widely about the max-likelihood location to exercise correspondence changes. However, this can lead to wide variation in the covariance estimate depending on the chosen sample set.

Another problem with Laplace's method is that it doesn't account for the possibility of multiple local minima. The only way to address local minima is to adequately search the relevant space of the likelihood function. One way to search the space is to draw N samples from the prior distribution of \mathbf{x}_s^r , and use each sample to initialise a separate run of ICP, each converging to a perhaps different local minima. The covariance estimate is then computed as a combination of the N solutions [31, Section 3.3] [25].

Monte Carlo importance sampling is an alternative, though rather expensive, method for estimating the first and second moments. With this approach, N samples are drawn from a proposal distribution, $\mathbf{x}_i \sim q(\mathbf{x}_s^r)$, and are weighted according to the ratio of the likelihood function and the proposal, $w_i = \Lambda(\mathbf{x}_i) / q(\mathbf{x}_i)$. The weights are normalised, such that $w_i^* = w_i / \sum_{j=1}^N w_j$, and the weighted samples are then used to compute the sample mean and covariance.

$$\hat{\mathbf{x}}_s^r = \sum_{i=1}^N w_i^* \mathbf{x}_i \quad (10)$$

$$\mathbf{P}_s^r = \sum_{i=1}^N (w_i^* \mathbf{x}_i \mathbf{x}_i^T) - \hat{\mathbf{x}}_s^r \hat{\mathbf{x}}_s^{rT} \quad (11)$$

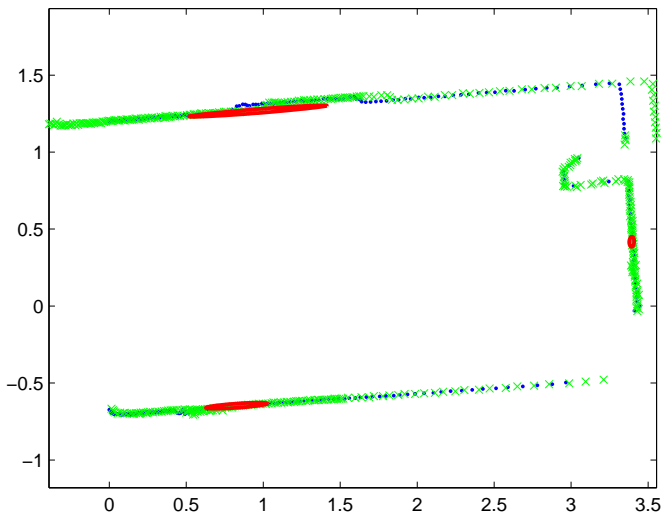


Fig. 8. Covariance estimates for the likelihood function of nearest neighbours with interpolation and gating. The reference scan is shown as blue dots and the models as green \times 's. The estimated covariances are depicted by red $3\text{-}\sigma$ ellipses.

Given many samples, and an appropriate proposal distribution, importance sampling can approximate moments of the likelihood function to arbitrary accuracy, but in practice there is a tradeoff with computational expense.

An example of an importance sampling estimate using the likelihood function described above is shown in Fig. 8. This figure shows a single reference scan and three models, each aligned separately. The covariance estimates are computed to high accuracy using 3000 samples. Notice that the mean of the top estimate is misaligned; also the second moment probably does not fully capture the shape of the likelihood function. The best option here might be to compute the trace of the covariance and, if it exceeds a threshold, reject that model observation. The right-hand model exhibits accurate alignment estimate, while the bottom model is fairly uncertain. This is an example where full-scan alignment would be preferable to smaller segmented models.

VII. DATA ASSOCIATION

The entire process of scan registration is fundamentally a data association operation. However, in this section we are interested in two specific aspects of data association apart from scan alignment itself: the search for candidate models, and validation gating. These two operations occur before and after scan alignment, respectively.

A. Candidate Search

When the vehicle obtains a new scan, it must determine which of its existing models might possibly overlap with this scan, and are therefore candidates for matching.

This operation is particularly hard because the landmark shape models may extend arbitrarily far from their local coordinate frames. So, while comparing the Mahalanobis distance of the landmark frame location to the vehicle frame is a cheap option, it does not account for model extent and will be a very suboptimal test. On the other hand, exhaustive tentative alignments of all models with the scan might work reasonably well, but is prohibitively expensive.

For small-scale environments, it is sufficient to assume that the errors are small, and simply choose those landmarks whose frames are predicted to fall within the vehicle field-of-view. A projection of their model points into the vehicle frame, and a gated nearest-neighbours test of the model points to the scan points, will give a reasonable indication of the amount of overlap.

However, this approach will not work reliably in larger environments where there is significant pose uncertainty. Fortunately, existing methods for large-error association are applicable to initialising scan alignment. One option is to extract features, such as foreground points or corner points, from the models and the current scan, and apply batch data association on these features [1, 22]. The discussion in [1, Section 6.3] gives an efficient multi-pass strategy for this approach. An alternative solution is to generate appearance signatures for each model, and perform signature matching with the current scan to generate candidate model proposals [23, 5].

B. Validation Gating

After scan alignment and covariance estimation has generated an observation $\{z, \mathbf{R}\}$, and before performing data-fusion with this observation, it is essential to perform validation to determine that the observation is not spurious. The standard Mahalanobis distance gate should be applied, preferably as a batch validation of all current observation hypotheses at once [1, 22].

However, landmark shape models also contain size and shape information that may be used as further discrimination measures. This is important when scanning lasers are used in realistic environments, since each laser scan measures a 2-D slice of the environment. When the vehicle pitches or travels over non-flat terrain, it takes slices of different planes, which can give rise to varying shapes appearing in close proximity over a sequence of scans.

A simple, though not particularly effective, shape matching gate can be applied by counting the number (or percentage) of points in the model that find acceptable nearest-neighbour associations in the current scan, as shown in Fig. 9. Superior shape matching techniques may be found in the literature (see, e.g., [30, 32]).

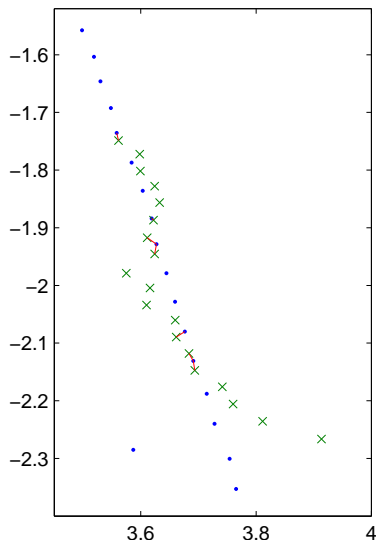


Fig. 9. Shape matching by counting the number of points with close nearest-neighbours after scan alignment. In this image, 7 out of 15 (i.e., 47%) of the \times points find neighbours closer than a 2cm gate. (We count only those \times 's from the first to last associated points, to allow for viewpoint variation.) The requirement of such a tight gate to discriminate two rather different shapes indicates that this method is rather suboptimal.

VIII. LANDMARK MODEL AUGMENTATION

As a landmark is viewed multiple times, possibly from different vantage points, the shape model may be augmented with the new data. While, strictly speaking, the model estimate is dependent on the SLAM estimate and should form part of the joint state vector, it is reasonable to treat model updates separately when the entire landmark lies within the sensing range. That is, if the error in model shape estimate is small, the bias introduced into the SLAM estimate by assuming independence is also small. By analogy with conventional SLAM, it is no different to the bias introduced by using a geometric circle model for a tree trunk that is not quite circular. An example of an object model augmented with multiple scans is shown in Fig. 10. After the alignment of many scans, the model could be subsampled to reduce its computational cost and to achieve a more even point distribution over its surface.

Arguably one of the advantages of the landmark-centric SLAM formulation is that it does not require estimation of the vehicle trajectory or a record of past scans. This allows it to operate reasonably efficiently using an EKF. However, the sparse information-form implementations of trajectory-centric SLAM have shown that keeping past pose estimates can actually increase estimate efficiency, by avoiding the information fill-in caused by marginalisation [12, 8]. Furthermore, the model augmentation process described above may be applied at the level of whole scans, wherein nearby

neighbourhoods of scans are merged to a single common coordinate frame, forming large multi-scan landmarks. (Although, it is possible that treating whole-scan merging as independent of the SLAM estimate may introduce more significant bias than for augmentation of smaller landmark models.) Therefore, whole-scan matching may also be formulated as landmark-centric SLAM and be performed in an EKF framework with relative efficiency.

IX. CONCLUSION

Scan-SLAM is a landmark-centric SLAM algorithm, with the same basic estimation structure as the earliest stochastic SLAM solutions. Its key difference is the definition of landmark shape models from raw sensor data, and defining the joint state vector of the SLAM estimate as the poses of model coordinate frames. Landmark-centric scan-matching SLAM shares much in common with existing trajectory-centric scan-matching SLAM algorithms, and the latter may be seen as a particular case wherein the landmarks are defined as whole scans. As such, there is significant overlap in the implementation issues for both forms.

Reliable estimation of scan alignment and its uncertainty is a non-trivial problem. The choice of representation, such as grids or points, presents tradeoffs in terms of computation, storage, and alignment assumptions. Maximum-likelihood alignment is prone to local minima, which can only be addressed by either having a good initial estimate or by adequately searching the likelihood space. And, since the likelihood function is typically very non-Gaussian, it can be difficult or expensive to compute a reliable covariance estimate. Nevertheless, the richness of a scan-based shape model may permit stronger observation validation than traditional models.

This paper also presents several key components for scan-matching SLAM in the small-to-medium scale. An efficient interpolating variant of ICP is used for scan alignment, and its covariance is found using a likelihood function that accounts for correspondence uncertainty. Non-visible surfaces are detected by examining the angle-order of model points; validation gating is assisted by shape comparison; and landmark models are augmented with successive scans. These components are a useful part of the toolbox necessary for an ICP-based implementation of scan-matching SLAM.

REFERENCES

- [1] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2002.

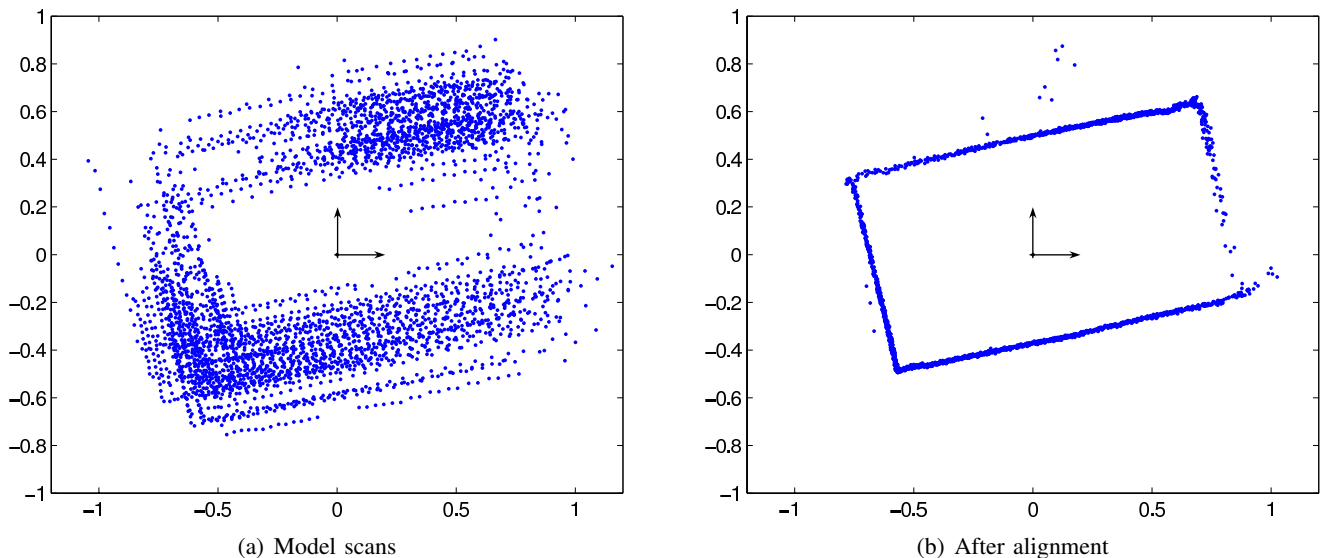


Fig. 10. Model augmentation with multiple scan data. Given an initial scan, which embeds the model’s coordinate frame, (a) subsequent scan segments are associated with the landmark, and (b) are aligned with the original scan to form an updated model estimate.

- [2] O. Bengtsson and A.-J. Baerveldt. Robot localization based on scan-matching—estimating the covariance matrix for the IDC algorithm. *Journal of Robotics and Autonomous Systems*, 44:29–40, 2003.
- [3] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [4] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *International Journal of Robotics Research*, 23(12):1113–1140, 2004.
- [5] M. Bosse and J. Roberts. Histogram matching and global initialization for laser-only SLAM in large unstructured environments. In *IEEE International Conference on Robotics and Automation*, pages 4820–4826, 2007.
- [6] A. Censi. An accurate closed-form estimate of ICPs covariance. In *IEEE International Conference on Robotics and Automation*, pages 3167–3172, 2007.
- [7] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation*, pages 2724–2729, 1991.
- [8] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006.
- [9] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
- [10] A. Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. In *Sixth Conference on Uncertainty in AI*, 1990.
- [11] A.I. Eliazar and R. Parr. DP-SLAM 2.0. In *IEEE International Conference on Robotics and Automation*, pages 1314–1320, 2004.
- [12] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Robotics: Science and Systems*, 2005.
- [13] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *IEEE International Conference on Robotics and Automation*, pages 667–672, 2005.
- [14] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318–325, 1999.
- [15] J.S. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *1st Euromicro Workshop on Advanced Mobile Robots (Eurobot’96)*, pages 61–67, 1996.
- [16] D. Hähnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 206–211, 2003.
- [17] K. Konolige. Large-scale map-making. In *National Conference on AI (AAAI)*, pages 457–463, 2004.
- [18] K. Konolige and K. Chou. Markov localization using correlation. In *International Joint Conference on Artificial Intelligence*, pages 1154–1159, 1999.
- [19] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
- [20] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1997.
- [21] D.J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [22] J. Neira, J.D. Tardós, and J.A. Castellanos. Linear time vehicle relocation in SLAM. In *IEEE International Conference on Robotics and Automation*, 2003.
- [23] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *IEEE International Conference on Robotics and Automation*, 2006.
- [24] J. Nieto, T. Bailey, and E. Nebot. Scan-SLAM: Combining EKF-SLAM and scan correlation. In *International Conference on Field and Service Robotics*, 2005.
- [25] J. Nieto, T. Bailey, and E. Nebot. Recursive scan-matching SLAM. *Journal of Robotics and Autonomous Systems*, 55(1):39–49, 2007.

- [26] S.T. Pfister, K.L. Kriechbaum, S.I. Roumeliotis, and J.W. Burdick. Weighted range sensor matching algorithms for mobile robot displacement estimation. In *IEEE International Conference on Robotics and Automation*, 2002.
- [27] A.C. Schultz and W. Adams. Continuous localization using evidence grids. In *IEEE International Conference on Robotics and Automation*, pages 2833–2839, 1998.
- [28] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *International Symposium of Robotics Research*, pages 467–474, 1987.
- [29] A.J. Stoddart, S. Lemke, A. Hilton, and T. Renn. Estimating pose uncertainty for surface registration. *Image and Vision Computing*, 16(2):111–120, 1998.
- [30] R.C. Veltkamp. Shape matching: similarity measures and algorithms. In *International Conference on Shape Modeling and Applications*, pages 188–197, 2001.
- [31] C.C. Wang. *Simultaneous Localization, Mapping and Moving Object Tracking*. PhD thesis, Carnegie Mellon University, Robotics Institute, 2004.
- [32] D. Wolter and L.J. Latecki. Shape matching for robot mapping. In *Pacific Rim International Conference on Artificial Intelligence*, pages 693–702, 2004.