Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans

Feng Lu and Evangelos Milios

Department of Computer Science, York University, North York, Canada M3J 1P3
{lufeng,eem}@cs.yorku.ca

Abstract

A mobile robot exploring an unknown environment has no absolute frame of reference for its position, other than features it detects through its sensors. Using distinguishable landmarks is one possible approach, but it requires solving the object recognition problem. In particular, when the robot uses two-dimensional laser range scans for localization, it is difficult to accurately detect and localize landmarks in the environment (such as corners and occlusions) from the range scans.

In this paper, we develop two new iterative algorithms to register a range scan to a previous scan so as to compute relative robot positions in an unknown environment, that avoid the above problems. The first algorithm is based on matching data points with tangent directions in two scans and minimizing a distance function in order to solve the displacement between the scans. The second algorithm establishes correspondences between points in the two scans and then solves the point-to-point least-squares problem to compute the relative pose of the two scans. Our methods work in curved environments and can handle partial occlusions by rejecting outliers.

1 Introduction

Previous research in mobile robotics has typically addressed several types of problems. The first is path planning in the case of a perfectly known environment and perfect sensing. The optimality criterion in this class of problems is the minimization of the cost for traversal between a start and an end position, while avoiding obstacles [18]. The second type of problem is exploration of an unknown world with perfect range sensing and odometry information [23]. Here the issues are primarily the complete coverage of the environment and the complexity of the algorithm as a function of the complexity of the environment (number of vertices and edges of objects).

The third type of problem is path execution within a known real environment. The focus here is typically on the sensing required to accurately execute a preplanned path, and the key problem is robot self-localization (the "where am I" problem). One primary issue in solving this problem is how to match sensed data (vision, sonar, laser, infrared etc.) against map information. A comprehensive collection of literature in this area can be found in [7,13]. The robot self-localization problem can be addressed either by metric means, by ensuring that the difference between the robot's actual position and the position where the robot thinks it is remains small and bounded, or by qualitative and topological means. It is well understood that odometry is not sufficient because it leads to unbounded position error [13]. A qualitative solution by tracking which side of landmark-defined lines the robot is on is proposed in [20].

If the robot is equipped with vision, then matching 3D models with 2D scene images is possible [9,16]. Because of the enormous computational requirements of using image data, using two-dimensional laser range scans has also been proposed and demonstrated successfully [5]. The robot "Blanche" assumes that a metric map of the environment consisting of polygonal obstacles is available, and it matches noisy range scans against the map frequently to keep the position error small [6]. At the heart of the method is an iterative least-squares algorithm that finds the congruence between a range scan and the map provided that the initial displacement is small. We can thus consider the problem of position estimation in a known (or partially known) two-dimensional polygonal environment as solved.

Finally, exploration of an unknown world with imperfect range sensing and odometry information is addressed [17]. Here the self-localization of the robot is still an important issue. Distinct locations or landmarks detected from sonar data are used in [19]. Matching local models of line segments obtained from sonar data with a cumulative global model is adopted in [8,24].

In the context of the above literature, our work addresses the robot self-localization problem by using laser range scans, similar to the work in [5]. However, we consider more general cases as (1) an arbitrary two-dimensional world, not necessarily polygonal; and (2) an unknown environment. We intend to address the localization problem for both path execution and exploration of unknown world.

The main issue in our problem is the consistent alignment of two data sets (range scans in particular). This is quite different from model-based matching [10]. Although a range scan represents a 2D shape (contour of the visible world), this shape is only represented by noisy discrete points instead of a high-quality model, which makes it very difficult to reliably define or extract features, as described in [10]. Another complication is due to the presence of occlusion, since the range scans from two different robot positions will only partially overlap. This implies that spatial correlation methods which use arclength to reference shape contours (such as [15]) can not be easily applied to scan-matching. The registration method based on fitting Spherical Attribute Images [12] also has difficulty in matching partial views.

Our approach is to start with an approximate alignment of the two scans (obtained from odometry), and then iteratively improve the alignment by defining and minimizing some distance between the scans. We present two algorithms. The first algorithm is based on matching data points with tangent directions on the two scans. It is formulated as searching a distance function to estimate the relative rotation between the scans and using an embedded least-squares procedure to solve for the relative translation. Previous methods which iteratively match points to curves or surfaces have also been reported [1,4]. They typically solve both rotation and translation from a least-squares procedure and use fixed-point iterations to improve the solution. To ensure convergence, an initial small rotation and translation must be assumed. Our method differs from previous methods in that we do not include the non-linear rotation component in the least-squares solution and we use a search procedure rather than fixed-point iterations to minimize the distance function. Thus we are able to solve larger rotational error than the previous methods.

Our second algorithm is based on iterative least-squares solutions using point to point correspondences, similar to the ICP algorithm proposed in [1]. But in addition to choosing the closest point on the model as the correspondence for a data point, we also choose the model point which has the same distance from the origin as the data point. Using two correspondence sets allows our method to estimate the rotation and translation very accurately and our method converges significantly faster than the ICP algorithm.

The rest of the paper is organized as follows: Section 2 defines the problem as the alignment of two range scans so as to recover the relative translation and rotation of the two scans. Section 3 describes the tangent-based matching method. Section 4 describes the point-based matching method. Section 5 summarizes the integration of the two algorithms and shows experimental results from both simulated and real laser range data. Finally, section 6 concludes the paper.

2 Problem Definition

2.1 Pose Estimation by Aligning Scans

A range scan is a list of points corresponding to the intersection points of a laser beam with objects in the robot's environment. The laser beam rotates in a horizontal plane and emanates from a sensor mounted on the robot. Thus a range scan describes a 2D slice of the environment. Points of the range scan can be specified in a polar coordinate system whose origin is the location of the sensor, and the reference axis for the laser beam direction is the home orientation of the rotating range sensor. Each scan point is represented by the laser beam direction, and the range measurement along that direction. We refer to $(O(x,y),\theta_0)$ as the scan pose, where O is the position of the sensor in a global coordinate system and θ_0 is the sensor home orientation.

Suppose that the robot starts at pose P_{ref} (which is our reference pose) and takes a scan (call it the reference scan S_{ref}). The robot then moves through a static environment to a new pose P_{new} and takes another scan (call it the new scan S_{new}). The approximate difference of pose P_{new} from pose P_{ref} (i.e. the relative translation and rotation) is usually known from odometry information. However, this information is often imperfect due to wheel slippage. Our task is to determine the exact difference of pose P_{new} with respect to pose P_{ref}, by aligning the two scans.

The matching problem is formulated as the following: Initially assuming that the pose of S_{new} is P'_{new}, find a rotation ω and a translation T for S_{new} such that, after the transformation, S_{new} is aligned with S_{ref}.

2.2 Criterion for Matching Scans

A scan is a sequence of points which represent the contour curve of the local environment. Due to the existence of random sensing noise and self-occlusion, it may be impossible to align two scans perfectly. There exist two types of discrepancies between the scans.

In the first type, there are small deviations of scan points from the true contour due to random sensing noise. These deviations can be characterized by an error distribution function which depends on the sensor. Another type of discrepancy is the gross difference between the scans caused by occlusion (i.e. some area is only visible in one scan but not in the other). We regard this kind of discrepancy as an outlier which cannot be described by the sensing error distribution.

Naturally, we adopt a scan matching criterion as to find the best alignment of the overlapping part in the sense of minimum least-squares error, while ignoring the outlier parts. We can formulate this matching criterion as the minimization of a distance measure between the two scans, as a function of the rotation and translation. To exclude outliers in defining the distance measure, we can apply the concept of robust statistics [11,2].

3 Search/Least-Squares Matching Algorithm

3.1 Method Overview

Our approach to the scan matching problem is to define a distance measure between the two scans and search for an appropriate rigid transformation which minimizes the distance. Although the search space is essentially a three-dimensional one (rotation and two-dimensional translation), we try to reduce the search problem to an efficient one-dimensional search plus an embedded least-squares solution, by carefully formulating the distance measure.

The idea of the matching method is briefly described below. First, we compute the tangent directions on the scans by fitting lines to sets of consecutive scan points. Then we associate approximate correspondences of scan points, assuming a known rotation angle ω (but not the translation T). From each pair of corresponding points, we formulate a linear equation about the unknown translation T. Then, by using all correspondence pairs, we define a least-squares model for T which also represents a matching distance as a function of ω. Outliers can be detected using gates and they contribute a constant cost to the distance measure. The final step is to search for a rotation ω which minimizes this distance function. The translation T is solved through the least-squares method. We refer to this method as rotation search/least-squares method to distinguish it from pure least-squares based fixed-point iteration type of methods.

The steps of the algorithm are summarized as the following:

1. Project the reference scan S_{ref} to the pose P'_{new} so that the two scans are represented in the same coordinate system. Discard those points on S_{ref} which are likely not to be visible from the new pose.
2. Compute the tangent directions on each scan by fitting lines to a neighborhood of sample points. Discard unreliable tangent lines (such as at corners or depth discontinuities).
3. Decide on a trial value of the rotation ω from a global search procedure.
4. For each point on S_{new}, use the rotation angle ω to define an approximate corresponding point on S_{ref} and compute the point through interpolation. Check the correspondence with some thresholds in order to reject outliers.
5. Use all the correspondence pairs to construct a least-squares model for T and find the least-squares solution. Define a matching distance as a function of ω from the least-squares residual and the outlier penalties.
6. Update the rigid transformation by the least-squares solution of T.
7. Repeat steps 3–6 and find the rotation ω which minimizes the matching distance function. Also obtain the overall translation by integrating the individual updates.

The algorithm will be explained in detail in the following sections.

3.2 Projection of Reference Scan

The reference scan S_{ref} is originally defined at pose P_{ref}. We would like to project it to the (approximate) new pose P'_{new} to emulate the world as viewed from P'_{new}. This is easily done by a change of coordinate systems for the points in S_{ref}.

After projection, we determine whether each point in S_{ref} is visible from P'_{new} based on the bounded obstacle assumption and opaque assumption. (1) Assume that initially the points in S_{ref} are ordered by their polar angles (say counterclockwise). After projecting the points to the new pose, if the new polar angles of some points are in the wrong order (i.e. become clockwise), then the surface containing these points is facing away from the sensor and thus the points are not visible. (2) Along the rays from the new origin P'_{new} to the points, if there are other points (either from S_{ref} itself or from the new scan S_{new}) close enough to the rays, then the points further from the origin are considered hidden (see Fig. 1 for illustration). The points determined as nonvisible are discarded.

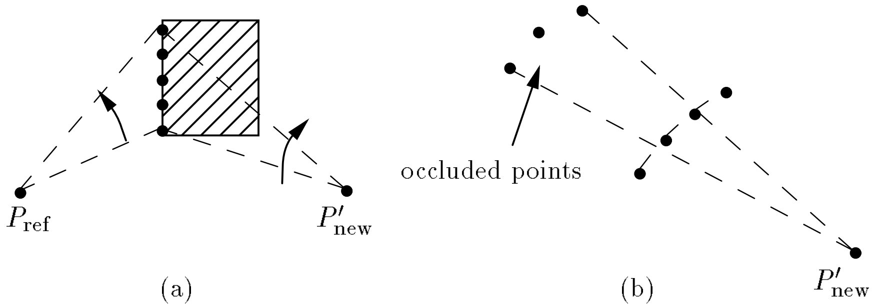


Figure 1: Scan points are considered invisible after projection if (a) their angles are in the wrong order or (b) there are other points blocking the rays.

Finally, we obtain a set of points in S_{ref} which are ordered by angles. The projected scan can be readily used as a reference for registering S_{new}.

3.3 Fitting Tangent Lines

At each sample point on a scan, we compute an approximated tangent line by fitting to a neighborhood of sample points centered at that point.

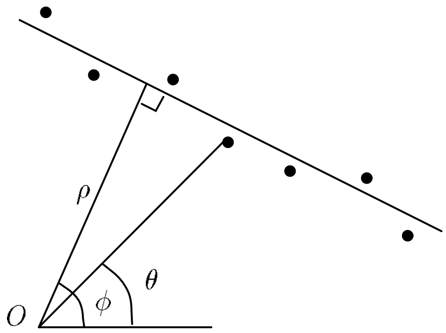


Figure 2: Parameters of a line fit to a set of points.

A line fit to a set of points (x_i, y_i)'s in neighborhood of size n is defined by the parameters ρ (normal distance from the origin to the line) and ϕ (direction of a normal to the line) (see Fig. 2) which minimize the following error:

$$E_{\mathrm{fit}} = \sum_{i=1}^{n}(x_i \cos\phi + y_i \sin\phi - \rho)^2. \tag{1}$$

There exists a closed-form solution for ρ, ϕ, and $min_{(\rho,\phi)}(E_{\mathrm{fit}})$ (see Appendix A).

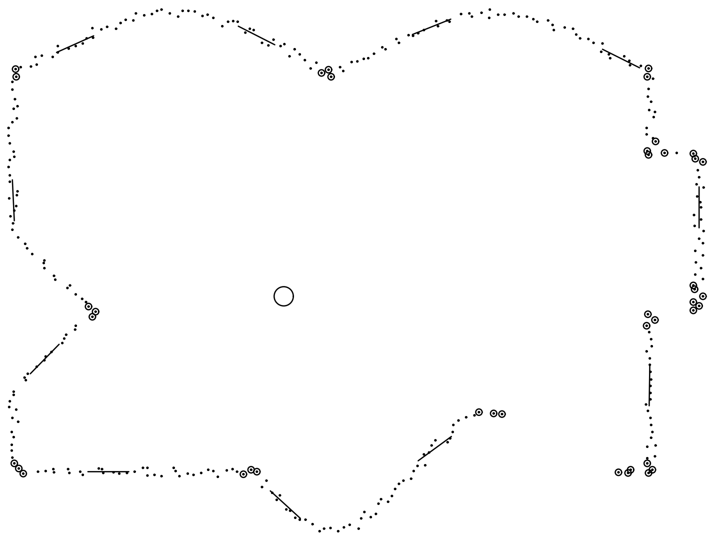


Figure 3: The points labeled by circles are discarded. Tangent lines are computed at all the other points. Only some of these tangent lines are shown for clarity.

Although we fit a line at every sample point, some of the lines should not be considered as tangent lines if the world contour itself is not smooth enough near these points. Specifically, we want to exclude sample points near corners or occlusion boundaries.

Two indicators can help us to recognize these non-smooth regions. First, we check the incidence angle $\theta - \phi$ (θ is the polar angle of the sample point and ϕ is the computed normal direction, see Fig. 2). A high incidence angle indicates that either the sensing direction is nearly parallel to a surface so that the range measurements are unreliable, or there is a depth discontinuity (occlusion boundary).

Another value we check is the fitting error $min(E_{\text{fit}})$ which indicates the co-linearity of the points or the smoothness of the local region. The error is usually small for straight lines or smooth curves, but large at high-curvature points or highly noisy regions. Therefore, a large fitting error usually means that the tangent line is poorly defined.

We use the above two indicators to reject unreliable fitting results. Only when both values are within predefined limits, we consider the fitted line as the tangent line at the current sample point. Otherwise, we regard the tangent line to be undefined. Fig. 3 shows an example of using incidence angle and fitting error to detect points where tangent lines are unreliable. We compute tangent lines at all the other scan points.

It is interesting to note that the points we discard for being unreliable in defining tangent (which are likely corners and occlusion boundaries) can be considered as features. It is possible to do feature-based matching using these points, provided that the correspondences between the features on the two scans can be determined. As the number of identifiable features in the scan is far less than the number of available data points, we believe that the feature-based solution is less accurate and less robust than our method which uses all the data points where a tangent is defined.

3.4 Correspondence Definition

We define the correspondence of scan points and set up an equation based on the values of a pair of corresponding points. For the convenience of analysis, we initially regard the scans as continuous curves rather than sets of discrete points and we also ignore sensing noise or occlusion, until we instantiate the equation for the actual scan points.

Once we have projected S_{ref} to the pose P'_{new}, we represent the two scans in the same coordinate system (defined by (T_0, θ_0)). The two scans differ only by a rotation ω and a translation $T = (T_x, T_y)^t$.

Let P_1 be a sample point on S_{new} and let P_2 be the true corresponding point on S_{ref} (i.e. they both represent the same physical point in the world). The two points are related by

$$P_2 = R_\omega P_1 + T, \tag{2}$$

where $R_\omega = \begin{pmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{pmatrix}$ is the rotation matrix. Consider the tangent line defined on S_{new} at P_1 and another tangent line on S_{new} at P_2. Let the normal directions of the two tangent lines be $\vec{n}_1$ and $\vec{n}_2$, respectively. Then we can derive from Eq. 2 that

$$\begin{aligned} \vec{n}_2 &= R_\omega \vec{n}_1 & (3) \\ \vec{n}_2 \cdot P_2 &= (R_\omega \vec{n}_1) \cdot (R_\omega P_1) + (R_\omega \vec{n}_1) \cdot T. & (4) \end{aligned}$$

Our strategy is to use Eq. 4 to estimate the translation T, given the rotation ω and the two scans. However, we note that for a point P_1 on S_{new}, the exact correspondence point P_2 on S_{ref} also depends on T. Therefore, we want to derive an approximated version of Eq. 4 which does not use the exact correspondence point P_2.

We will choose a point P^* on S_{ref} which is close to P_2. There are many choices of selecting P^* based on $R_\omega P_1$ (which is P_1 after correcting the rotation), such as the closest point from $R_\omega P_1$ or the intersection of the normal line at $R_\omega P_1$ with S_{ref}. For the convenience of searching, we choose P^* as the intersection of the extension of vector $R_\omega P_1$ with scan S_{ref} (see Fig. 4 for illustration). Let $\vec{n}^*$ be the normal direction of the tangent line at P^* on S_{ref}. If scan S_{ref} is smooth and if P^* is not too far away from P_2, we have the approximate relationships:

$$\begin{aligned} (R_\omega \vec{n}_1) \cdot T &\approx (R_\omega \vec{n}_1) \cdot (P^* - R_\omega P_1) & (5) \\ \vec{n}^* \cdot T &\approx \vec{n}^* \cdot (P^* - R_\omega P_1). & (6) \end{aligned}$$

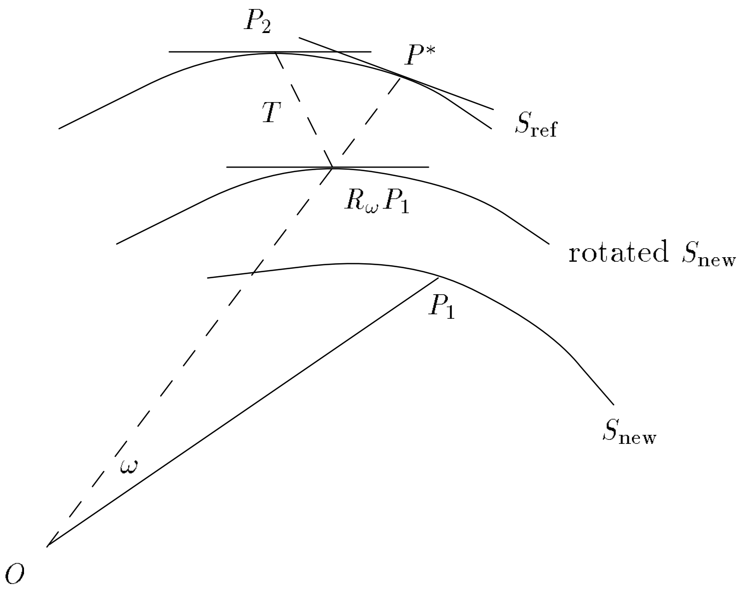


Figure 4: Illustration of point relationships.

The approximation errors of the above relationships are of the order $O(x^2)$ where x is related to $|T|/|P_1|$ (see Appendix B for derivation). Therefore, if $|T| \ll |P_1|$ and if the contour curve is smooth, the approximation error is small. We can combine the above two relationships to form a more accurate approximation whose error is of the order $O(x^3)$ (see Appendix B):

$$(R_\omega \vec{n}_1 + \vec{n}^*) \cdot T \approx (R_\omega \vec{n}_1 + \vec{n}^*) \cdot (P^* - R_\omega P_1). \tag{7}$$

Eq. 7 is a linear equation about T in the form:

$$C^x T_x + C^y T_y \approx D. \tag{8}$$

It is established based on the point P_1 and the rotation angle ω. Notice that, given P_1 and ω, we can determine P^* without using T. The coefficients C^x, C^y, D can be computed from ω and the parameters of the tangent lines.

Now if we instantiate Eq. 7 for n_p points on S_{new}, we can define an error function:

$$E(\omega, T) = \sum_{i=1}^{n_p} (C_i^x T_x + C_i^y T_y - D_i)^2. \tag{9}$$

This error indicates the "distance" between the two scans as a function of ω and T. For a given ω, we can solve for T which minimizes the above distance, in terms of ω. Moreover, the least-squares error $min_T\, E(\omega, T)$ can be considered as a matching distance function which is primarily defined by the rotation ω. Our final solution is based on searching the function for an optimal ω such that the least-squares error is minimum.

3.5 Correspondence Search

Considering the fact that S_{ref} is a discrete set of points, we can obtain the approximate correspondence point P^* for P_1 by interpolation. Let the polar angle of P_1 be θ, then the point P^* is at direction $\theta + \omega$. Note that the points in S_{ref} are ordered by their polar angles. We can locate two consecutive points on S_{ref} whose angles enclose $\theta + \omega$. Then we linearly interpolate for both the range and the normal direction of P^* across the two points. This correspondence search is very efficient. We can implement the search for all correspondence pairs in linear time with respect to the number of points in the two scans.

Correspondence points which differ greatly in their normal directions or positions are considered as outliers and they are not included in the least-squares solution. Typical causes of outliers are occlusion or poor tangent estimation. We empirically choose thresholds α and H_d. A correspondence pair is accepted only if both of the following conditions are met:

$$(R_\omega \vec{n}_1) \cdot \vec{n}^* \geq \cos\alpha; \quad |D_i| \leq H_d. \tag{10}$$

Otherwise, we consider the pair as outlier and discard it. We also exclude the points if they are near a corner or a depth discontinuity (as detected by the line fitting procedure).

3.6 Optimization

We define a total matching distance for a given rotation ω based on the least-squares error (Eq. 9) and the number of outliers. Let n_p be the number of matching pairs of points and n_o be the number of outliers, the total matching distance is defined as:

$$E_{\text{match}}(\omega) = \frac{1}{n_p + n_o}(\min_T E(\omega, T) + n_o H_d^2), \tag{11}$$

where H_d^2 is the constant cost of an outlier (note that H_d is the threshold that detects outliers). The effect of the thresholding and adding fixed cost for outliers to the least-squares error is approximately equivalent to using a truncated quadratic robust estimator to define the distance measure. The advantage of using a robust estimator is that a small number of arbitrarily bad outliers do not lead to an arbitrarily bad estimation [11,2].

The matching distance in Eq. 11 is a function of ω. Implicitly, it is also a function of the translation T (because the least-squares solution T may be difference from the true translation). But the variable ω is dominant. We expect that the distance function should have a global minimum at the true rotation angle and that the function is unimodal near the minimum. From the plot of a typical $E_{\text{match}}(\omega)$ (Figure 5), we can see that the distance function has a single prominent valley. Furthermore, if there is little translation, it has a well-defined lowest point.

An optimization procedure needs to be applied to minimize a distance function. Because the distance function is generally not a smooth one, gradient based search methods are not applicable.

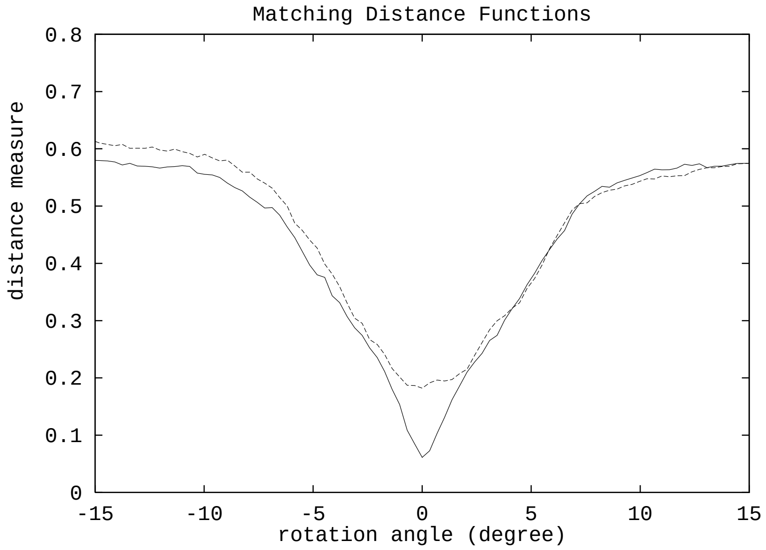


Figure 5: Matching distance measure as a function of rotation angle, in an example. The solid curve is the distance function when translation is zero. The dashed curve is the distance function when translation is (0.2, 0.1) meters (comparing with dimension of the environment of 10 meters).

One possible way of searching for a minimum is to sample the search space. Stochastic methods (such as simulated annealing) may also be applied to minimize the non-smooth distance function (such as [3]). These methods are usually computationally expensive.

We use the search by golden section method [14] to find the minimum along the rotation dimension in the distance function. We choose this method because of its efficiency in terms of required function evaluations. The search by the golden section method is stated as follows. Assume that the global minimum is enclosed in an interval $[\omega_1, \omega_2]$. The first trial point is set at $x_1 = \omega_1 + 0.618(\omega_2 - \omega_1)$. At every step, a new point is chosen at $x_2 = \omega_1 + \omega_2 - x_1$. Then depending on whether the function has a lower value at x_1 or x_2, the better point becomes the new x_1 and the worse one replaces ω_1 or ω_2 to form a new interval which still encloses the minimum. After sufficient iterations, the final interval will be narrow enough to localize the minimum point.

Due to the residue in the translation, the optimal rotation found by the one-dimensional search may be biased. To address this problem and reduce the bias, we correct the translation by the least-squares solution every time we evaluate the distance function. As the search narrows down the rotation, the translation residue is also getting smaller and so is the bias.

From the plot, we can see that the valley of the curve is within an interval of about 0.5 radians (30 degrees) in width. Beyond this interval, the curve is mostly flat. We require this initial interval to start the search procedure. However, in case the initial rotation error is very large or completely unknown, we can determine an initial interval by coarsely sampling the function at some rotation values. For example, we can sample the function at every 15 degrees (resulting in a total of 24 samples) and choose two adjacent points which have the lowest function values to form an interval. This effectively allows our method to handle arbitrarily large rotations.

The amount of translational error handled by our method depends on the threshold H_d. Usually the algorithm is good enough to handle the residuals of the odometry estimates.

4 Point Correspondence Based Matching

4.1 General Approach

We now present another method to align two range scans. It is an iterative algorithm and is based on associating point to point correspondence. The idea of the method is the following. For each point P_i on S_{new}, we use a simple rule (independent of the actual rotation and translation) to determine a corresponding point P_i' on S_{ref}. Then from all the correspondence pairs of points, we compute a least-squares solution of the relative rotation and translation. This solution is applied to reduce the pose error between the two scans. We repeat this process until it converges.

The least-squares solution is derived by minimizing the following distance function which is defined on n pairs of correspondence points:

$$E_{\text{dist}}(\omega, T) = \sum_{i=1}^{n} |R_\omega P_i + T - P_i'|^2. \tag{12}$$

Closed-form solutions for ω and T can be derived (see Appendix C).

The central issue of the algorithm is then to define a sensible rule to determine correspondences without knowing the actual rotation and translation.

4.2 Rules for Correspondence

We describe the correspondence rules in the following sections. For convenience, we will regard the reference scan S_{ref} as a continuous curve and refer to it as the model. We will consider the correspondence search on a discrete scan later.

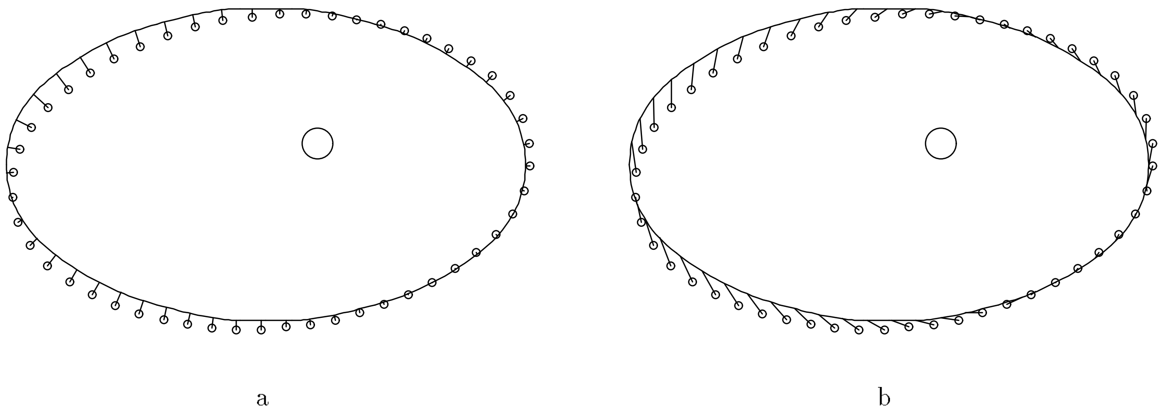


Figure 6: Use rules to determine correspondences (which are joined by line segments in the figure). (a) Use the closest-point rule; (b) use the matching-range-point rule. The model is the ellipse. The data points are labeled by small circles. The big circle in the center is the origin.

4.2.1 Closest-Point Rule

A commonly used rule is to choose the closest point on the model as the correspondence for a data point. We refer to this rule as the *closest-point rule*. Fig. 6(a) shows an example of finding the correspondences for a set of points in an elliptic model, using the closest-point rule. Besl and McKay described a general-purpose iterative closest point (ICP) algorithm for shape registration based on this rule and they proved that the ICP algorithm always converges monotonically to a local minimum with respect to the least-squares distance function [1]. We observe from experiments that, if the rotation is small, the ICP algorithm is good at solving the translation.

One disadvantage of the ICP algorithm is that it converges very slowly. Especially, when the model is curved, the correspondences found by the closest-point rule may contain little information about the rotation. As seen from Fig. 6(a), the vectors joining the correspondence pairs have very inconsistent directions and they tend to cancel out each other when used together to solve for the rotation. Moreover, regardless of the type of the model, the convergence speed of the algorithm is always very slow when the distance function approachs a local minimum.

To accelerate the ICP algorithm, Besl and McKay used a line search method to heuristically determine the transformation variables based on their values in two or three recent iterations. Although this improves the convergence speed near a local minimum, the problem of obtaining a poor solution for the rotation component still exists. Therefore, the improvement in convergence speed is limited. Moreover, in order to apply the line search method, an implicit assumption is made about the smoothness of the least-square distance function. But this is typically not true if the number of correspondence pairs changes during the iterations (as a result of rejecting outliers).

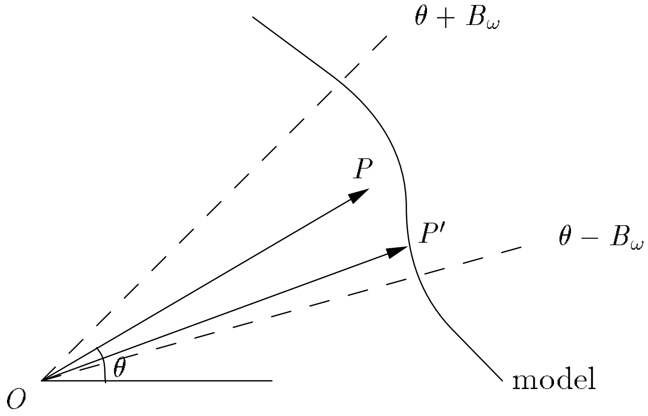


Figure 7: Matching-range-point rule: For a point P, the corresponding point P' on the scan lies within the sector and $|OP'|$ is closest to $|OP|$.

4.2.2 Matching-Range-Point Rule

We propose a different rule which finds the correspondences in such a way that they significantly reveal the rotation component.

Consider a data point P and its corresponding point $P' = R_\omega P + T$. If we ignore the translation, we have $|P'| \approx |P|$. On the other hand, the polar angle θ of P and the polar angle $\hat{\theta}$ of P' are related by $\hat{\theta} \approx \theta + \omega$. This implies that the correspondence of P under a rotation is a point which has the same polar range as that of P, and the polar angles of the corresponding points differ by the rotation angle ω. Now in the presence of a small translation, we can still expect that the point P' with the same range as that of P is possibly a good approximation to the true correspondence of P, and this approximate correspondence provides rich information about the rotation angle ω.

To ensure that the rule finds a unique and reliable correspondence, we only search for the matching-range point within a local region of the model near P. Suppose that we can estimate a bound B_ω for the rotation ω, i.e. $|\omega| \leq B_\omega$. We have: $\hat{\theta} \in [\theta - B_\omega, \theta + B_\omega]$. This means that P' should lie within the sector bounded by $\theta \pm B_\omega$. Therefore, we propose the *matching-range-point rule* as the following: *For a data point P, its corresponding point is P' on the model where P' satisfies $|\hat{\theta} - \theta| \leq B_\omega$ and $|P'|$ is closest to $|P|$.* The rule is illustrated in Fig. 7.

Fig. 6(b) shows an example of using the matching-range-point rule to find correspondences in the elliptic model. We can clearly see that the vectors joining the correspondence pairs consistently indicate the rotation direction. Therefore, the least-squares solution should be a good approximation to the true transformation, especially in the rotation component.

Based on this new rule, we design an iterative matching-range-point (IMRP) algorithm. In this algorithm, the parameter B_ω controls the size of the neighborhood to be searched for a correspondence and also the maximum rotation possibly solved in one iteration. Thus it will be best to choose B_ω to be close to the rotation residual at every iteration. We empirically generate B_ω using an exponentially decreasing function of the form: $B_\omega(t) = B_\omega(0)e^{-\alpha t}$. It is also possible to use the least-squares solution of ω in the current iteration to determine the value of B_ω for the next iteration.

A comparison of the performance of the ICP algorithm and the IMRP algorithm is illustrated in Fig. 8. We notice that the IMRP algorithm converges faster than the ICP algorithm in estimating the rotation. For the translation part, the IMRP algorithm is initially slower but it eventually converges faster than the ICP algorithm.

The reason that the translation residuals from the IMRP algorithm are initially reduced slowly is that the matching-range-point rule only tends to influence the translation component when the sector width B_ω becomes small enough. In the early iteration when B_ω is large, the algorithm tends to explain the displacement between the correspondence pairs by rotation rather than translation. This phenomenon may present a potential problem to the stability of the algorithm if we need to dynamically reject outliers using a threshold, as good correspondences may be falsely rejected as outliers due to the incorrect translation. We will study more about outlier detection later.

4.2.3 Combining the Two Rules

It is desirable to combine the two rules in order to achieve both the convergence speed of the matching-range-point rule and the stability of the closest-point rule. We propose an *iterative dual correspondence* (IDC) algorithm which uses both rules, as the following:

1. In each iteration do the following steps:
2. For each data point P_i,
   (a) apply the closest-point rule to determine a correspondence point P_i' for P_i.
   (b) apply the matching-range-point rule to determine a correspondence point P_i'' for P_i.
3. Compute the least-squares solution (ω_1, T_1) from the set of correspondence pairs (P_i, P_i'), $i = 1, \ldots, n$ (which are obtained using the closest-point rule).
4. Compute the least-squares solution (ω_2, T_2) from the set of correspondence pairs (P_i, P_i''), $i = 1, \ldots, n$ (which are obtained using the matching-range point rule).
5. Form (ω_2, T_1) as the solution for the transformation in the current iteration.

The basic idea of the above algorithm is to take the translation component from the closest-point rule solution and the rotation component from the matching-range-point rule solution to form the current solution for the transformation.

The combination of the two rules appears to achieve significantly better results than each of the individual rules. It does not only ensure the stability of the iterations, but also increases the convergence speed significantly. In fact, the two rules reinforce each other by reducing different components of the transformation, so that each of them can be more effective in the following iteration. We notice that the IDC algorithm is insensitive to the choices of parameter B_ω.

Fig. 8 illustrates the residuals of the transformation components during the iterations of the three algorithms. The model and the data points in this example are the ones given in Fig. 6, where the initial rotation is -6 degrees; the initial translation is (-5, 5) units (the width of the elliptic model is 1000 units). Clearly, the IDC algorithm reduces the residuals much more quickly than the other two single-rule algorithms.

We experimentally estimate the rate of convergence for each of the three algorithms using the above example. The iterative closest point algorithm appears to have a sublinear convergence rate. The ratio of error residuals ($C = e_{i+1}/e_i$) gets larger and larger and it approaches to 1. In fact, after 30 iterations, C is at 0.998. For both the iterative matching-range point algorithm and the iterative dual correspondence algorithm, the rate of convergence seems to be linear. The error ratio for the iterative matching-range point algorithm is $C = 0.875$; the error ratio for the iterative dual correspondence algorithm is $C = 0.685$.

4.3 Correspondence Search

We will use the IDC algorithm to register a new scan S_{new} to the reference scan (model) S_{ref}. Considering that S_{ref} is discrete, we need to interpolate in order to locate the correspondence point for each of the two rules.

First, we consider the matching-range-point rule. Let $P(r, \theta)$ be a point on S_{new}. We want to find another point $P'(\hat{r}, \hat{\theta})$ on S_{ref} according to the matching-range rule. Since the points in scan S_{ref} are parameterized in a polar coordinate system, we choose to linearly interpolate $1/r$ from θ between two points.

Let $P_1(\theta_1, r_1)$, $P_2(\theta_2, r_2)$ be two adjacent points, the interpolation function $\hat{r}(\hat{\theta})$ is:

$$\begin{aligned} \hat{r} &= 1 \Bigg/ \left(\frac{1}{r_1} + \frac{\hat{\theta} - \theta_1}{\theta_2 - \theta_1} \left(\frac{1}{r_2} - \frac{1}{r_1} \right) \right) \\ &= \frac{r_1 r_2 (\theta_2 - \theta_1)}{r_1(\hat{\theta} - \theta_1) + r_2(\theta_2 - \hat{\theta})} \end{aligned} \tag{13}$$

This interpolation scheme is approximately equivalent to connecting the two points with a line

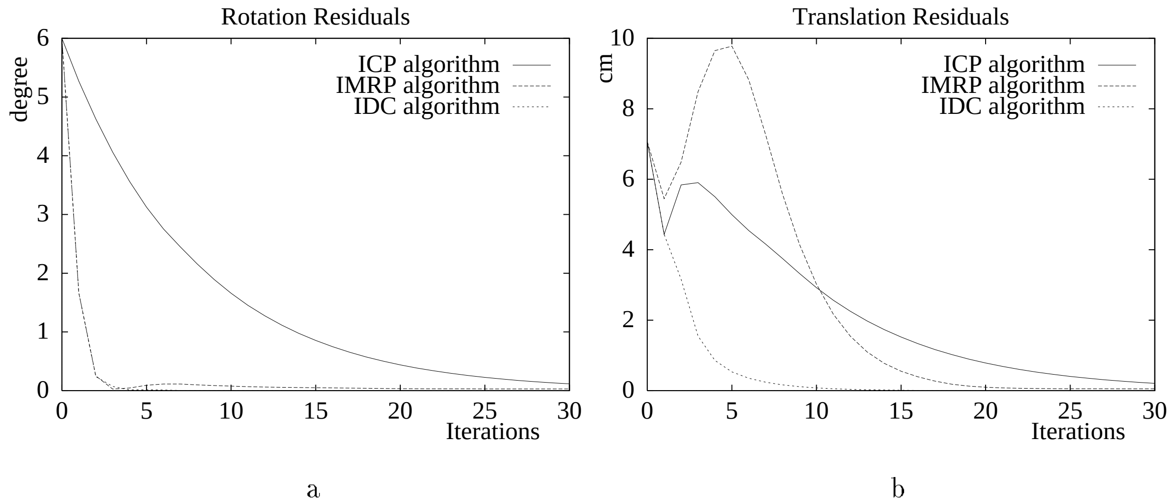


Figure 8: Comparison of iterative algorithms. (a) Rotation residuals from the three algorithms over 30 iterations (note the curve for the IDC algorithm is barely visible as it is almost zero after three iterations). The initial rotation angle is 6 degrees. (b) Magnitudes of translation residuals. The magnitude of initial translation is 7.07 units. The model and data set are given in Figure 6.

segment if $|\theta_2 - \theta_1|$ is small. This can be seen by comparing Equation 13 with the equation of a straight line passing through P_1 and P_2:

$$\hat{r} = \frac{r_1 r_2 \sin(\theta_2 - \theta_1)}{r_1 \sin(\hat{\theta} - \theta_1) + r_2 \sin(\theta_2 - \hat{\theta})}. \tag{14}$$

The interpolation in Equation 13 also has the advantage that the interpolated $\hat{r}$ is a monotonic function of $\hat{\theta}$; it is easy to inversely compute $\hat{\theta}$ for a given $\hat{r}$.

The region where we need to search for P' consists of pairs of points whose angular intervals overlap with $[\theta - B_\omega, \theta + B_\omega]$. Let $[\theta_1, \theta_2]$ be one such interval. We find the intersection $[\hat{\theta}_1, \hat{\theta}_2] = [\theta_1, \theta_2] \cap [\theta - B_\omega, \theta + B_\omega]$. Then we use Eq. 13 to interpolate two points $P_1'(\hat{\theta}_1, \hat{r}_1)$ and $P_2'(\hat{\theta}_2, \hat{r}_2)$ at the ends of the new interval. If $\hat{r}_1 \leq r$ and $\hat{r}_2 \leq r$, or $\hat{r}_1 \geq r$ and $\hat{r}_2 \geq r$, one of P_1' and P_2' (whose polar range is closer to r) is a candidate for P'. In this case, we still need to check other intervals for a potentially better candidate. But if $\hat{r}_1 \leq r \leq \hat{r}_2$, or $\hat{r}_1 \geq r \geq \hat{r}_2$, we can (inversely) interpolate for a P' which has the same range as P.

Now we consider the closest-point rule. For consistency with the matching-range rule, we slightly modify the closest-point rule such that it chooses the closest point P' within the sector $[\theta - B_\omega, \theta + B_\omega]$ only as the correspondence of P. The interpolation here is simply connecting two adjacent points with a line segment. In a similar way as in the search for a matching-range correspondence, we search all the angular intervals and find a P' by checking the distance from P to the line segments (or end-points).

The search required to determine all the correspondence pairs can be efficiently implemented in linear time with respect to the number of points on the two scans, by considering the fact that all the points in the scans are ordered by their angles.

Due to occlusion, we may not find appropriate correspondence points for some data points. We need to identify these outliers and exclude them from the least-squares distance function. Assume that we have a bound B_r for the maximum displacement resulting from translation for a pair of corresponding points P and P', i.e. $||P'| - |P|| \leq B_r$. We use this condition to test for outliers. For a pair of points found by one of the rules, we accept them as a correspondence pair only if the above condition is met. The threshold B_r can be selected as the kth largest distance among all the correspondence pairs, according to a predetermined constant fraction (p-tile).

4.4 Convergence of Iterative Algorithm

The iterative algorithm terminates when the change in the least-squares errors in sufficiently small. In practice, we find it sufficient to fix the number of iterations to about 15–20.

The iterative process in the algorithm resembles fixed-point iterations. Since the zero pose error is a stable fixed-point for the process, we expect that once the iteration converges that the final solution should be very accurate. To ensure convergence a good initial estimate (small initial pose error) is required by the algorithm. Usually, we can use odometry to provide this initial estimate. Another possibility is to first apply the rotation search/least-squares method which we presented in Section 3 to obtain a relatively good registration before applying the point-based method. We find from experiments that this strategy usually guarantees the convergence of the point-based method and the point-based method also usually improves the accuracy of the solution.

5 Scan Matching Experiments

5.1 Combining The Two Algorithms

We have presented two algorithms, the rotation search/least-squares algorithm and the iterative point correspondence algorithm, for matching range scans. The rotation search/least-squares uses a search procedure in the optimization process. This allows the algorithm to robustly solve for the transformation even in the presence of a large initial pose error. However, because the matching distance function is not smooth and we only use a simple one-dimensional search procedure to do the minimization, the solution given by this algorithm may not be highly accurate. On the other hand, the iterative point correspondence based algorithm gives a more accurate solution through its fix-point iterations, as long as it converges. This natually leads to a two-staged algorithm to solve a general matching problem by sequentially combining the two algorithms. We apply this algorithm in the following experiments.

5.2 Sensing Strategy

If we have control over the sensing directions (as is the case with the ARK sensor [25]), we can choose to take a range scan in such a way that the sampling points are evenly distributed in space. In other words, we want the distances between adjacent points to be approximately equal. This is more efficient than a uniform sampling of orientation where the interval width is proportional to the ranges of the points from the sensor.

Assume that the current sample point is $P(\theta, r)$. We will decide the direction of the next shot, $\theta + \Delta\theta$, such that the new point is a fixed distance d away from P. If we approximate the contour curve near P by its tangent line (which has parameter (ρ, ϕ)), the increment in sensing direction is:

$$\Delta\theta \;=\; \arctan \frac{d \cos(\theta - \phi)}{r + d \sin(\theta - \phi)}, \tag{15}$$

where the angle $\theta - \phi$ can be computed by the following:

$$\theta - \phi \;=\; \arctan(\frac{1}{r}\frac{dr}{d\theta}). \tag{16}$$

Note that the derivative $\frac{dr}{d\theta}$ can be estimated from recent range measurements. Combine the above two equations and take a first order approximation (for a small d), we get:

$$\Delta\theta \;\approx\; \frac{d}{\sqrt{r^2 + (\frac{dr}{d\theta})^2}}. \tag{17}$$

To avoid infinitely small increments, we can predefine a minimum value for $\Delta\theta$. Now $\theta + \Delta\theta$ is the direction to take the next measurement.

This sensing strategy is used to generate both simulated and real scan data in our experiments.

5.3 Experiments with Simulated Data

Range measurements are simulated for various environments in order to test the matching algorithms. We model the environments with lines and spline curves. Sensing noises are modeled as additive random variables. For a given environment model, we can simulate a range scan from any robot pose.

We present a series of experiments using simulated range data. In each experiment, we choose an environment model and two poses, P_{ref} and P_{new}, from which to take the scans S_{ref} and S_{new}. The pose error is generated randomly where ω is uniformly distributed in [-0.25, 0.25] radians

($\pm 14.3°$) and T is uniformly distributed in a disk of radius 50 centimeters (comparing with the dimension of the environment of 10 meters). Sensing noise is assumed to be uniformly distributed. We vary the maximum sensing noise in each experiment.

In each experiment setting, the matching process was run 1000 times (with randomly generated initial pose error and sensing noise as described above). We computed the standard deviations (from the theoretical zero means) of the residuals of the variables (ω, T_x, and T_y) resulting from each matching process. The standard deviations of the residuals after the first stage (rotation search algorithm) and after the second stage (iterative point correspondence algorithm) were recorded. In the matching process, the number of iterations was fixed at 15 in each of the two stages.

The results from six experiments are listed in Table 1. In each experiment, we list the environment model, the maximum sensing noise, and the standard deviations of the residuals. We also plot all the translation residuals from each of the six sets of experiments (Fig. 9). Each translation residual is shown as a dot in the x-y plane. The dots form a cluster around the origin.

We have some observations from the statistical results. (1) The pose error residuals are very small compared with the sensing noise. The standard deviation of translation residuals is about 15% to 25% of the standard deviation of sensing noise. The rotation residuals are well within one degree. With a typical sensor accuracy of 5cm, the rotation residuals are within one tenth of a degree. (2) The algorithm behaved robustly over thousands of runs. (3) The second algorithm (iterative point correspondence algorithm) gave significantly more accurate estimates of rotation than the first algorithm. But the two methods are equally good in finding the translations (the second algorithm is slightly better when the sensing noise is relatively small or when the environment is mostly linear (experiments 1, 2, 3). (4) The residuals appear to be normally distributed with approximately zero means. (5) When the sensing noise becomes large (experiments 5 and 6), the algorithm degrades gracefully. In experiment 6 (where the sensing noise is set to 20cm), it appears that the limit of the algorithm is reached. We experienced a few failure cases where the iterative algorithm did not converge. There were about 10 such cases among the 1000 runs. The failure is partly caused by the step of rejecting outliers using thresholds. Raising the threshold level may improve convergence. But it may result in bias in the estimation because of outliers. High sensing noise also leads to bad estimation of local contour directions. Thus the correspondence association step (which involves interpolation) is subject to large error.

We plot three typical matching examples from the above experiments. In each example, we show the two scans which are misplaced with respect to each other. Then the same two scans after alignment are shown. In the figures, the reference scan is plotted with x's. The big X at the center is the actual pose P_{ref} of the reference scan. The new scan to be registered is plotted with circles. The big circles with arrows are the poses of the new scan before and after registration (that is, P'_{new} and P_{new}).

Example 1 (Fig. 10) is a typical case from experiment 3. Here the rotation is 0.25 radians (14.3°)

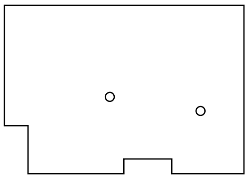
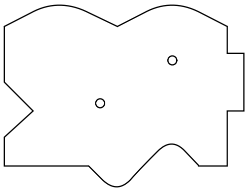
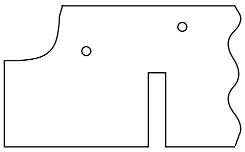
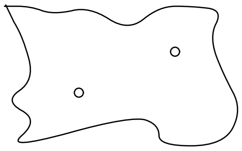
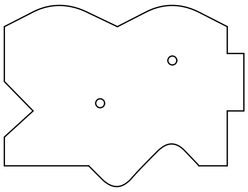
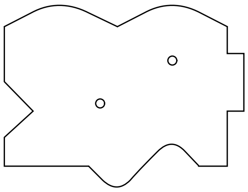
No.	Simulated Environments and Maximum Sensing Noise	Residual Standard Deviations		
			After Stage 1	After Stage 2
1	 noise: $\pm$5cm	σ_ω	0.2786 °	0.0547 °
		σ_x	0.3909 cm	0.3418 cm
		σ_y	0.4531 cm	0.2702 cm
2	 noise: $\pm$5cm	σ_ω	0.3668 °	0.0754 °
		σ_x	0.4150 cm	0.3592 cm
		σ_y	0.3886 cm	0.3146 cm
3	 noise: $\pm$10cm	σ_ω	0.4414 °	0.1876 °
		σ_x	1.3449 cm	1.0436 cm
		σ_y	1.5336 cm	0.8532 cm
4	 noise: $\pm$10cm	σ_ω	0.3970 °	0.1824 °
		σ_x	0.8723 cm	0.9535 cm
		σ_y	0.7836 cm	0.8446 cm
5	 noise: $\pm$15cm	σ_ω	0.6090 °	0.3027 °
		σ_x	1.2268 cm	1.2604 cm
		σ_y	1.1269 cm	1.1438 cm
6	 noise: $\pm$20cm	σ_ω	1.1517 °	0.6230 °
		σ_x	2.2832 cm	2.5478 cm
		σ_y	2.1961 cm	2.1811 cm

Table 1: Statistics of experiments in simulated environments. Maximum initial rotation and translation are set at $\pm 14.3°$ and 50cm, respectively.

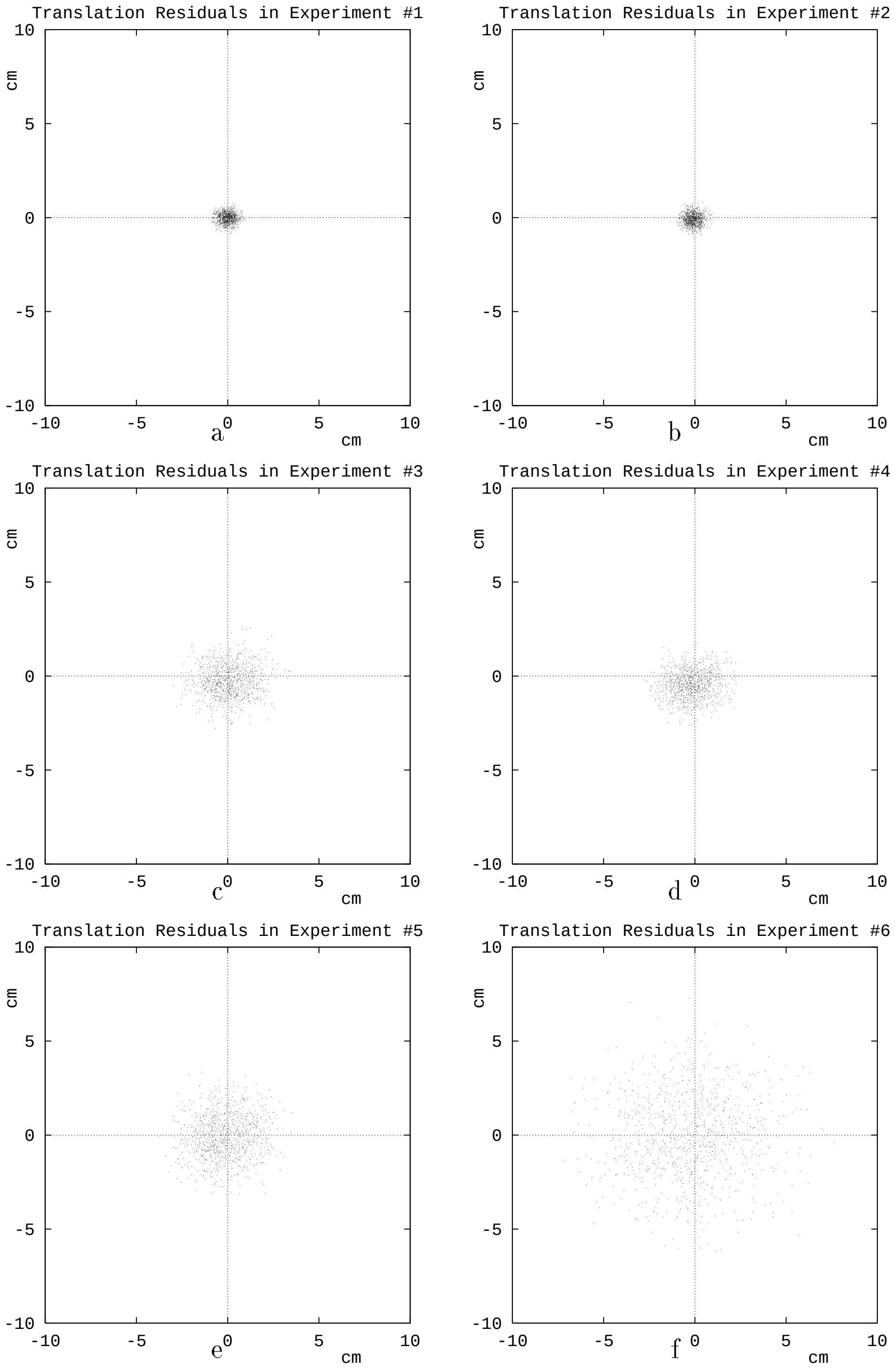
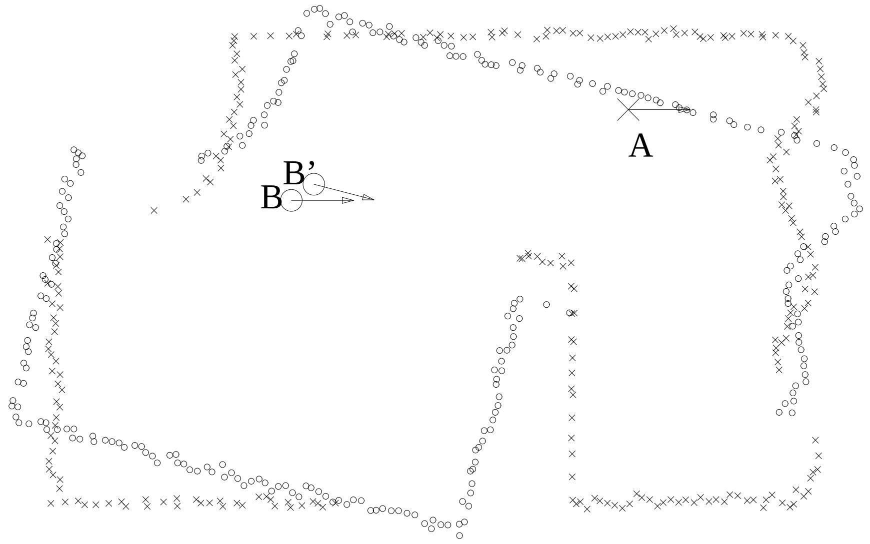
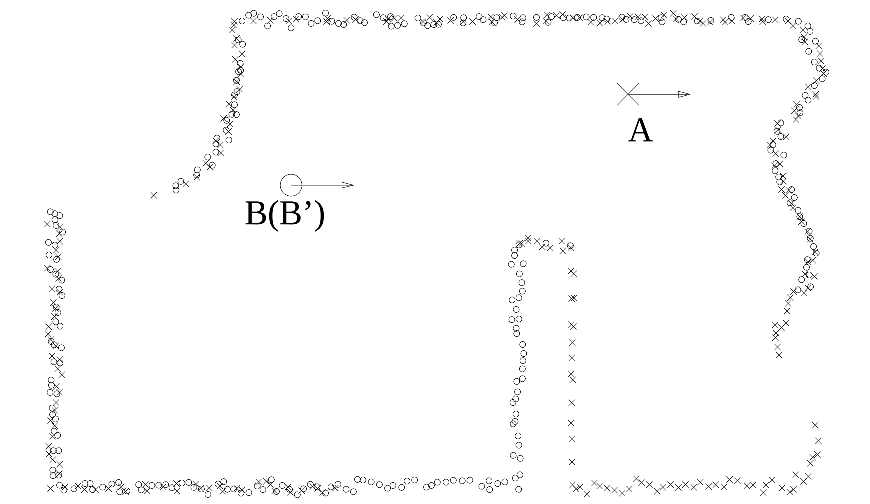


Figure 9: Translation residuals after apply the matching algorithm. (a) to (f) correspond to experiments 1 to 6 in table 1. There are 1000 samples in each experiment. Larger spread of dots in the plot corresponds to higher levels of sensing noise.



a



b

Figure 10: Example 1 (from experiment 3, maximum sensing noise is 10cm). The robot went from pose A to pose B in reality, but due to odometry errors, it thinks it went to pose B'. Points on scans from pose A and B are labeled as x's and small circles respectively. The poses are indicated by arrows. Part (a) shows the alignment error due to the difference between pose B' and true pose B. Part (b) shows the result of aligning the two scans. The pose B' is corrected to the true pose B by the same transformation. Notice the large occlusion in the lower right corner.

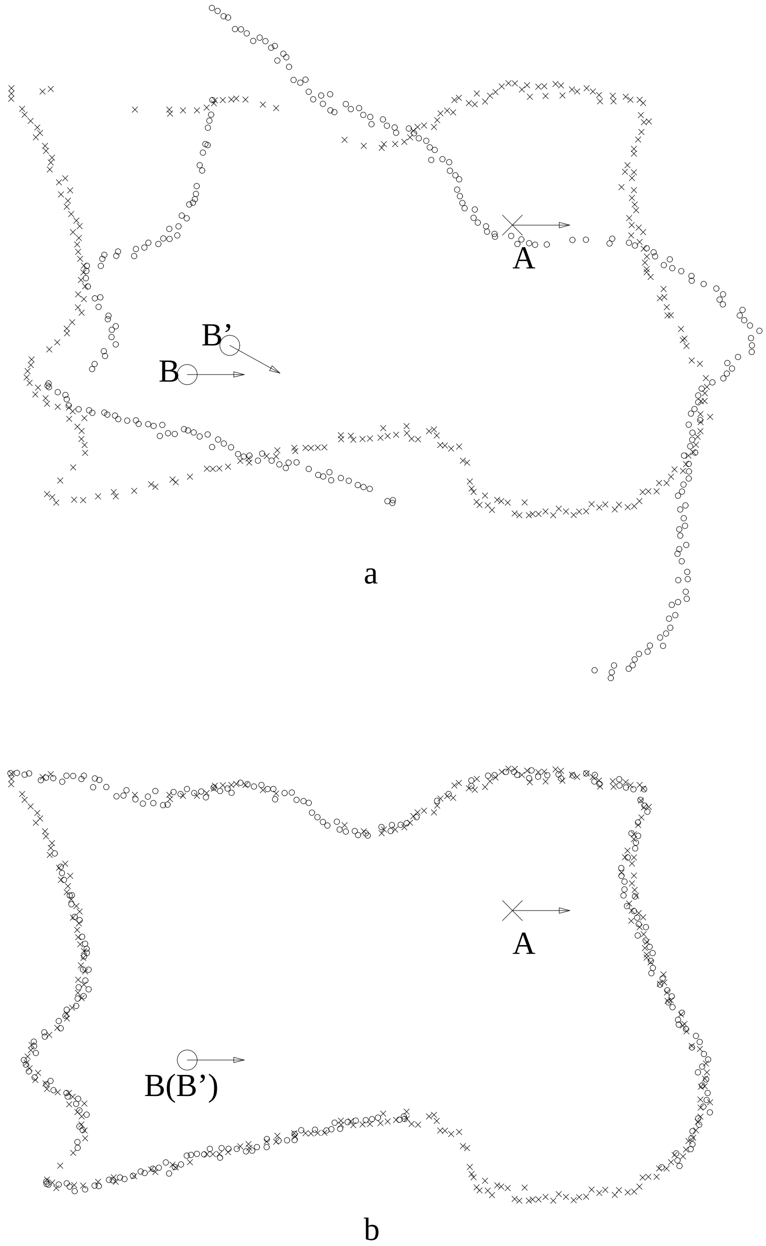
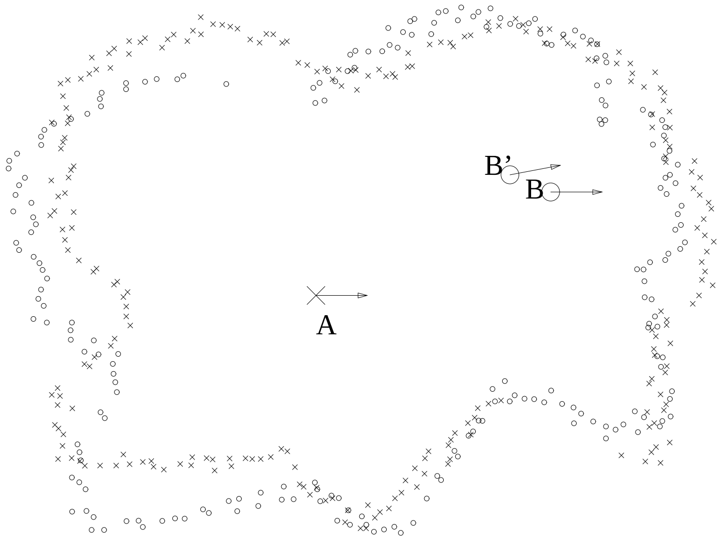
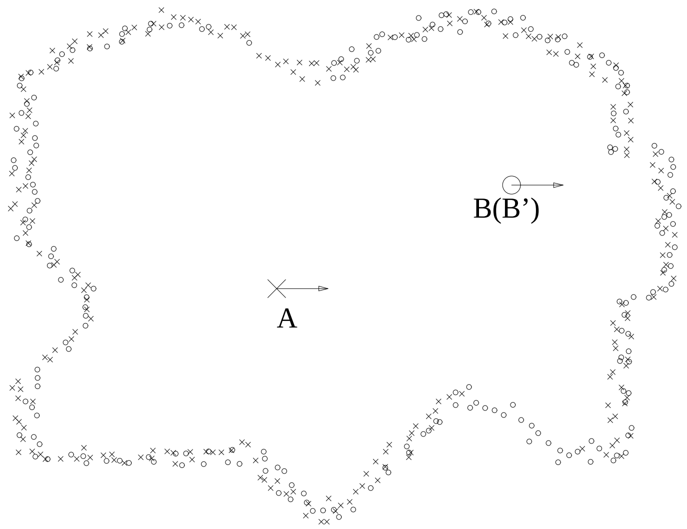


Figure 11: Example 2 (from experiment 4, maximum sensing noise is 10cm). Part (a) shows the pose error (indicated by the difference from B' to B) and the resulted alignment error of the scans. Part (b) shows the result of correcting the pose error and aligning the two scans. The large rotation and translation are corrected successfully. There is occlusion in the scan.



a



b

Figure 12: Example 3 (from experiment 6, maximum sensing noise is 20cm). Part (a) shows the pose error (indicated by the difference from B' to B) and the resulted alignment error of the scans. Part (b) shows the result of correcting the pose error and aligning the two scans. This level of sensing noise is about the limit of algorithm capacity.

and the translation is $(-30, -20)$ centimeters. Notice that there is a substantial portion of the world (about one third) which is visible only in one scan but not in the other. Our algorithm is able to ignore the unmatched parts but successfully perform the matching based on the remaining part of the world which is visible in both scans.

Example 2 (Fig. 11) is from experiment 4, in which the environment consists of mostly curves (rotation 0.5 radians ($28.6°$), translation $(-60, 40)$ centimeters). There also exist occlusions in the scans. Our matching method is successful.

Example 3 (Fig. 12) is chosen from experiment 6, in which the maximum sensing noise is as large as ± 20 centimeters. Rotation is -0.2 radians (-11.5 degrees) and translation is $(-40, 30)$ centimeters. The correct alignment is found by the algorithm in this example. Note this level of sensing noise is close to the limit of algorithm capacity.

In the above experiments, the second algorithm (point correspondence based method) is not fairly tested as its input (from the output of the first algorithm) is too good. Here we present another experiment which examines the iterative point correspondence algorithm without first applying the rotation search method. We use the same environment as the one in experiment 2, but set the maximum sensing noise to ± 10cm. The initial pose error is randomly generated so that ω is uniformly distributed over [-0.1, 0.1] radians and T is uniformly distributed in a disk of radius 20 centimeters. We ran this experiment 1000 times. The standard deviations of the residuals are listed in Table 2. For comparison, we also run the rotation search algorithm using the same input set. We can see that the second algorithm is still successful despite of the relatively larger initial pose errors. Again, the second algorithm performs significantly better than the first one in finding the rotation, but only slightly better in finding the translation.

Residual		First Algorithm	Second Algorithm
Rotation	σ_ω	0.5375 °	0.1599 °
Translation x	σ_x	0.7652 cm	0.7827 cm
Translation y	σ_y	0.7998 cm	0.6514 cm

Table 2: Standard deviations of the residuals from the two algorithms when they are run independently with the same set of input. Maximum sensing noise is ± 10cm. Maximum initial rotation and translation are $\pm 5.7°$ and 20cm, respectively.

5.4 Experiments with Real Data

In this section, we present the experimental results using real range sensor data in two different environments. In the first experiment, we used the ARK robot [25] to collect range scans in the Vision Lab at the Department of Computer Science, University of Toronto. The ARK robot is equipped with a laser rangefinder (model Optech G150) mounted on a pan and tilt unit on the robot platform. The accuracy of the rangefinder is 4 centimeters.

We present two examples of matching scans from this environment (Figures 13 and 14). Relatively large pose errors between the two scans were intentionally introduced. Unfortunately, the ground truth for this experiment is not available as the robot was driven by hand. Our matching algorithm successfully aligned the scans in both examples. Notice that in performing the matching the algorithm is able to ignore the cluttered areas at the right-side wall and the bottom left corner (which are bookshelves). The algorithm also handled the occluded area well.

Another testing environment is the cafeteria and nearby corridor in FAW, Ulm, Germany. The laser rangefinder used was a Ladar 2D IBEO Lasertechnik mounted on the AMOS robot. The laser sensor has a maximum viewing angle of 220 degrees, and an accuracy of $\pm 20mm$. We obtained 30 scans which were taken by the robot along its path at a pace of approximately 2 meters between scan poses. We then selected 84 pairs of scans which have sufficient overlap and applied the scan matching algorithm to each pair. Our matching algorithm successfully aligned the scans in all 84 cases. All of these scan matching results were used by a global scan registration algorithm which is discussed in [22,21].

Two examples of the matching results are shown in Figures 15 and 16. In both examples, we exaggerated the pose errors for testing the algorithm. The odometry error from the actual robot is much smaller. Note that this robot samples orientation uniformly when taking the scans, as opposed to our early strategy of making the sample points uniformly dense. But the matching algorithm still works well with this kind of scan.

We also include another example using real range data from a hallway (Figure 17). Notice that the constraints along the hallway direction is weak (the only reference are several shallow doorways). Our algorithm is able to derive the correct matching.

It appears that the real sensor data are more accurate than our typical simulated data. After thoroughly testing our algorithm with simulations, we believe that it should also work well in real indoor environments of this type. Moreover, in a typical indoor robot course where range measurements are taken at short discrete steps, the pose errors are usually much smaller than the ones we simulated. As a conclusion from the experiments, we believe that our algorithm should work robustly in real-life robotic applications.

6 Conclusion

We have developed two methods for matching two range scans and correcting the relative pose error. The two methods can either be used individually or applied in sequence. Our matching methods have the following desirable properties:

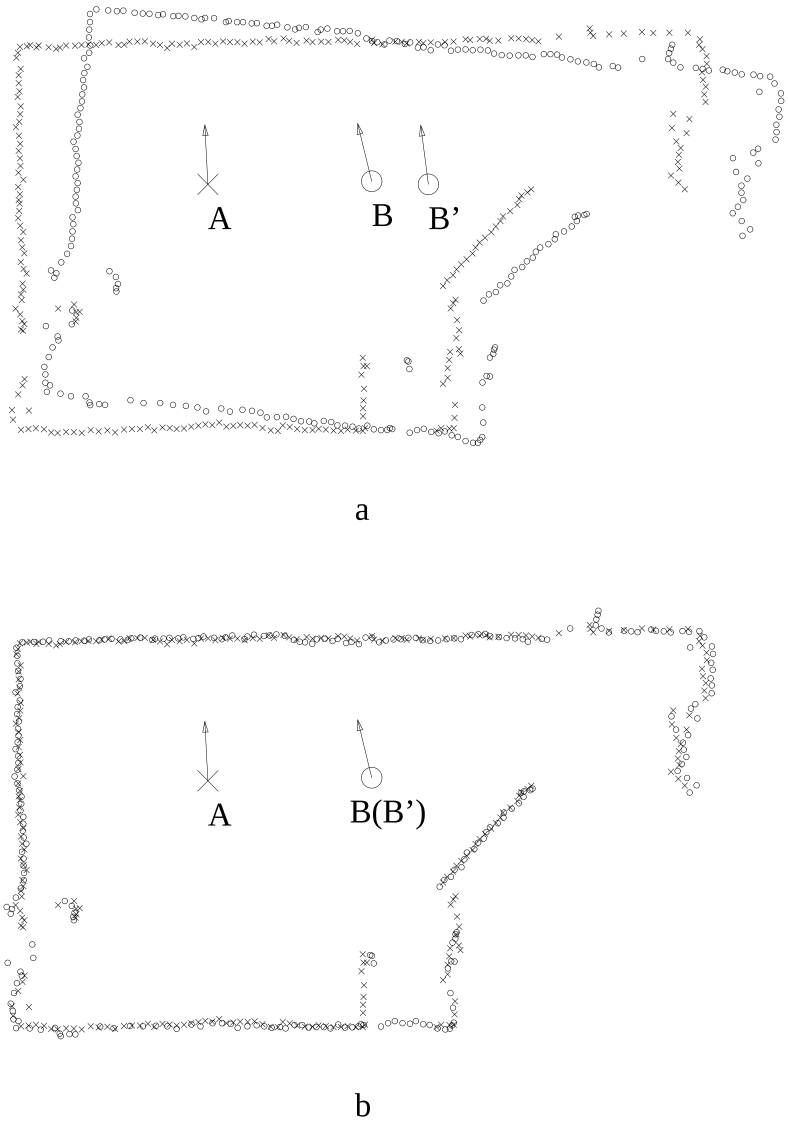
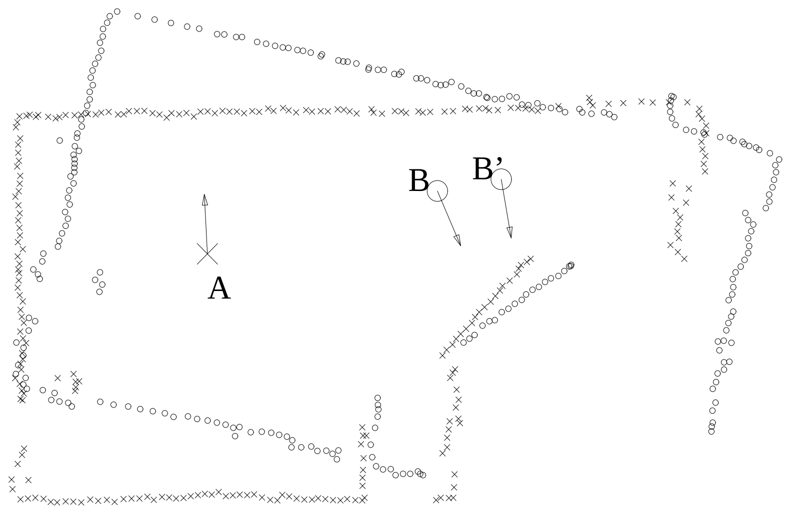
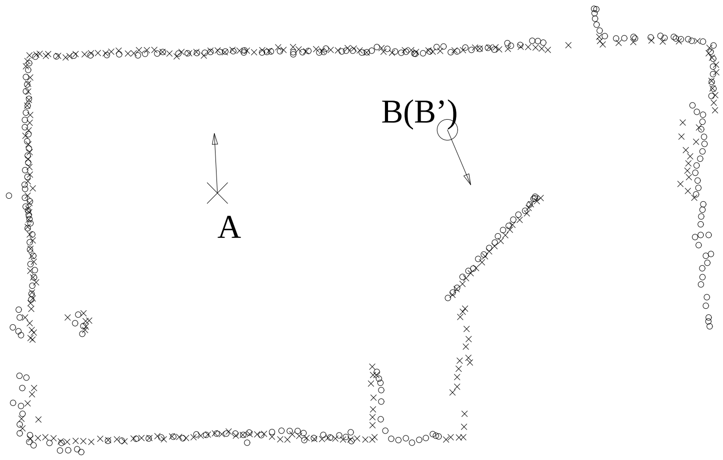


Figure 13: Example of matching scans taken by the ARK robot at U of T.



a



b

Figure 14: Another example of matching scans taken by the ARK robot.

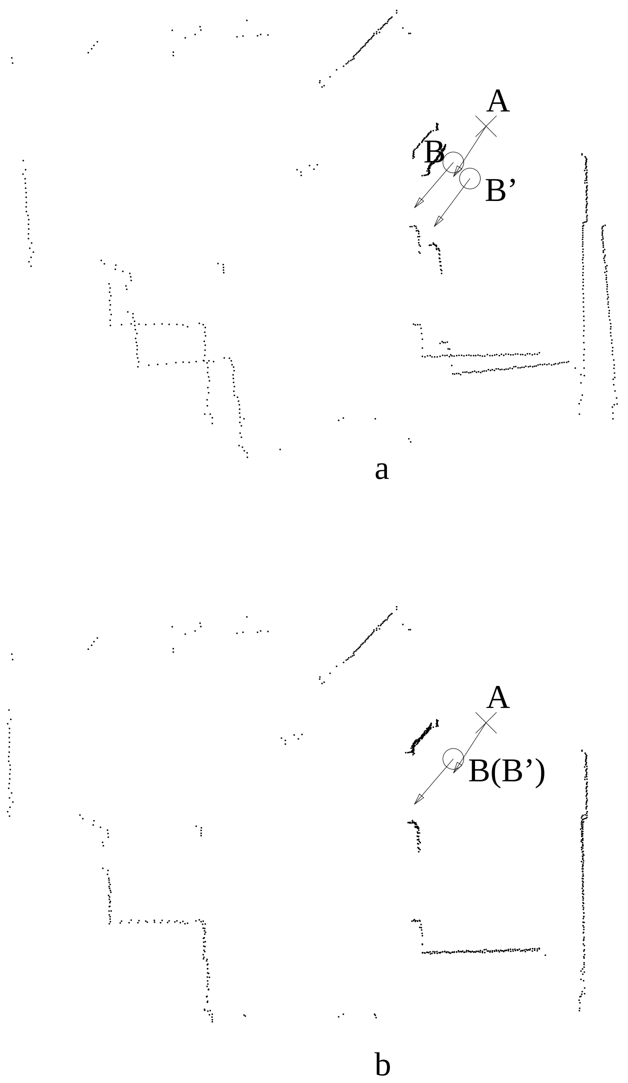


Figure 15: Example of matching real scans at FAW, Ulm, Germany. The scans are shown in dots. Part (a) shows the pose error (indicated by the difference from B' to B) and the resulted alignment error of the scans. Part (b) shows the result of correcting the pose error and aligning the two scans.

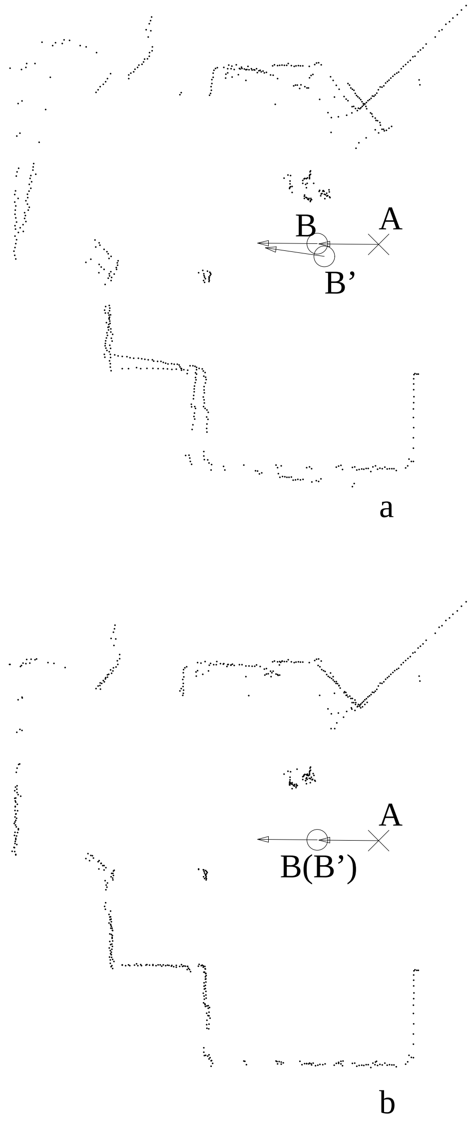
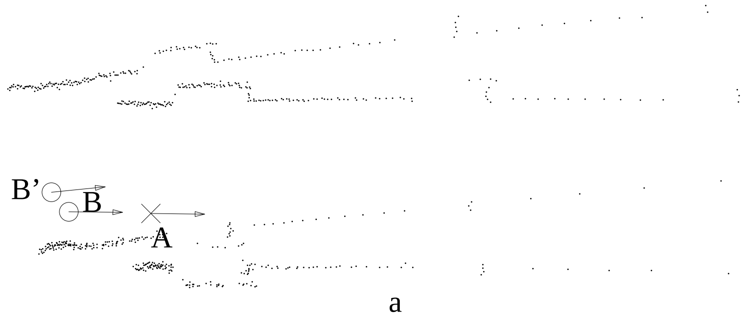


Figure 16: Another example of matching real scans at FAW, Ulm, Germany.



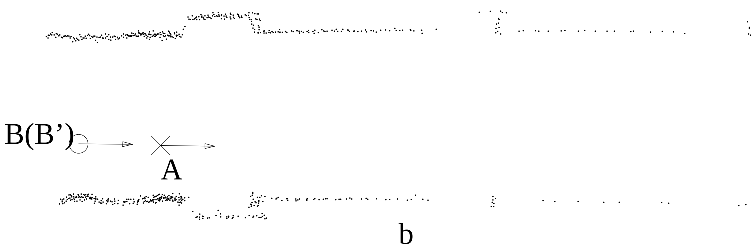


Figure 17: Example using real data from a hallway. Notice that the constraints are weak along the hallway direction. Alignment still can be made using our algorithm.

1. Both methods can handle sensor noise and partial occlusion. By checking against thresholds, the methods can effectively reject possible outliers. The methods are not very sensitive to the choice of threshold values.
2. The methods do not rely on distinguishable features in the environment. Thus we avoid the difficult process of feature detection and feature correspondence. The availability of the number of features should not concern us either. In fact, we use most of the sensor data in the matching process. This suggests that our algorithms are robust.
3. We do not require an *a priori* world model. We can match a scan directly to another scan. Therefore our method can be used for exploration and map building in unknown environments.
4. Our method can work in a general curved environment. We do not require the world contours to be linear, as many other methods do.
5. The algorithm is fast (linear time in the number of sensor data points per iteration, for a small fixed number of iterations). There is no exponential searching needed in finding the primitive correspondence.
6. Our algorithm has the ability to correct arbitrarily large errors in rotation.
7. Our algorithms are demonstrated to be superior over similar existing methods in robustness and efficiency.

We have tested our algorithm thoroughly with simulated data under various adverse conditions. We also tested the algorithm with real data from two different environments using different sensors.

Acknowledgements

Funding for this work was provided by NSERC Canada and by the ARK project which receives its funding from PRECARN Associates Inc., the Department of Industry, Science and Technology, NRC Canada, the Ontario Technology Fund, Ontario Hydro, and AECL. The authors would like to thank Steffen Gutmann, Joerg Illmann, Thomas Kaempke, Manfred Knick, Erwin Prassler, and Christian Schlegel from FAW, Ulm for collecting range scans and making the data available for our experiments.

References

[1] 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.

[2] M. J. Black. *Robust Incremental Optical Flow.* PhD thesis, Yale University, September 1992.

[3] G. Blais and M. D. Levine. Registering multiple range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1995.

[4] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *International Journal of Image and Vision Computing*, 10(3):145–155, 1992.

[5] I. J. Cox. Blanche: Position estimation for an autonomous robot vehicle. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles.* Springer-Verlag, 1990.

[6] I. J. Cox. Blanche: An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, 1991.

[7] I. J. Cox and J. B. Kruskal. On the congruence of noisy images to line segments models. In *Second International Conference on Computer Vision*, pages 252–258, 1988.

[8] J. L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *IEEE International Conference on Robotics and Automation*, pages 674–680, 1989.

[9] C. Fennema, A. Hanson, E. Riseman, J. R. Beveridge, and R. Kumar. Model-directed mobile robot navigation. *IEEE Transaction on Systems, Man, and Cybernetics*, 20(6):1352–1369, 1990.

[10] W. E. L. Grimson, T. Lozano-Pérez, and D. P. Huttenlocher. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, Cambridge, MA, 1990.

[11] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics: The Approach Based on Influence Functions.* John Wiley and Sons, New York, NY, 1986.

[12] K. Higuchi, M. Hebert, and K. Ikeuchi. Building 3-D models from unregistered range images. In *IEEE International Conference on Robotics and Automation*, pages 2248–2253, 1994.

[13] S. Iyengar and A. Elfes. *Autonomous Mobile Robots, Vols. 1 and 2.* IEEE Computer Society Press, Los Alamitos, CA, 1991.

[14] D. Kahaner, C. Moler, and S. Nash. *Numerical Methods and Software.* Prentice Hall, Englewood Cliffs, NJ, 1989.

[15] A. Kalvin, E. Sconberg, J.T. Schwartz, , and M. Sharir. Two-dimensional, model-based, boundary matching using footprints. *International Journal of Robotics Research*, 5(4):38–55, 1986.

[16] A. Kosaka and A. C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image Understanding*, 56(3):271–329, 1992.

[17] B. Kuipers and Y. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.

[18] J-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, Mass., 1991.

[19] J. Leonard, H. Durrant-Whyte, and I. J. Cox. Dynamic map building for an autonomous mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1990.

[20] T.S. Levitt and D. T. Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44:305–360, 1990.

[21] F. Lu. *Shape Registration Using Optimization for Mobile Robot Navigation.* PhD thesis, University of Toronto, Department of Computer Science, 1995.

[22] F. Lu and E. Milios. Optimal global pose estimation for consistent sensor data registration. In *IEEE International Conference on Robotics and Automation*, pages 93–100, Nagoya, Japan, 1995.

[23] V. Lumelsky. A comparative study of the path length performance of maze-searching and robot motion algorithms. *IEEE Transactions on Robotics and Automation*, 7(1), 1991.

[24] P. MacKenzie and G. Dudek. Precise positioning using model-based maps. In *IEEE International Conference on Robotics and Automation*, 1994.

[25] S. B. Nickerson, M. Jenkin, E. Milios, B. Down, P. Jasiobedzki, A. Jepson, D. Terzopoulos, J. Tsotsos, D. Wilkes, N. Bains, and K. Tran. Design of ARK, a sensor-based mobile robot for industrial environments. In *INTELLIGENT VEHICLES*, Tokyo, Japan, 1993.

Appendix

A Line Fitting

Let $(x_i, y_i), i = 1, \ldots, n$ be a set of points in Cartesian coordinates. The line fitted to these points has parameter (ρ, ϕ) which minimizes the following error:

$$E_{\mathrm{fit}} = \sum_{i=1}^{n} (x_i \cos\phi + y_i \sin\phi - \rho)^2.$$

A closed-form solution can be derived as the following:

$$\begin{aligned} \phi &= \frac{1}{2} \arctan \frac{-2S_{xy}}{S_{y^2} - S_{x^2}} \\ \rho &= \bar{x}\cos\phi + \bar{y}\sin\phi, \end{aligned}$$

and

$$\min_{(\phi,\rho)}(E_{\text{fit}}) = \frac{1}{2}\left(S_{x^2} + S_{y^2} - \sqrt{4S_{xy}^2 + (S_{y^2} - S_{x^2})^2}\right),$$

where

$$\begin{aligned}
\bar{x} &= \frac{1}{n}\sum_{i=1}^{n} x_i, & \bar{y} &= \frac{1}{n}\sum_{i=1}^{n} y_i \\
S_{x^2} &= \sum_{i=1}^{n}(x_i - \bar{x})^2, & S_{y^2} &= \sum_{i=1}^{n}(y_i - \bar{y})^2 \\
S_{xy} &= \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).
\end{aligned}$$

B Error from Approximate Point Correspondence

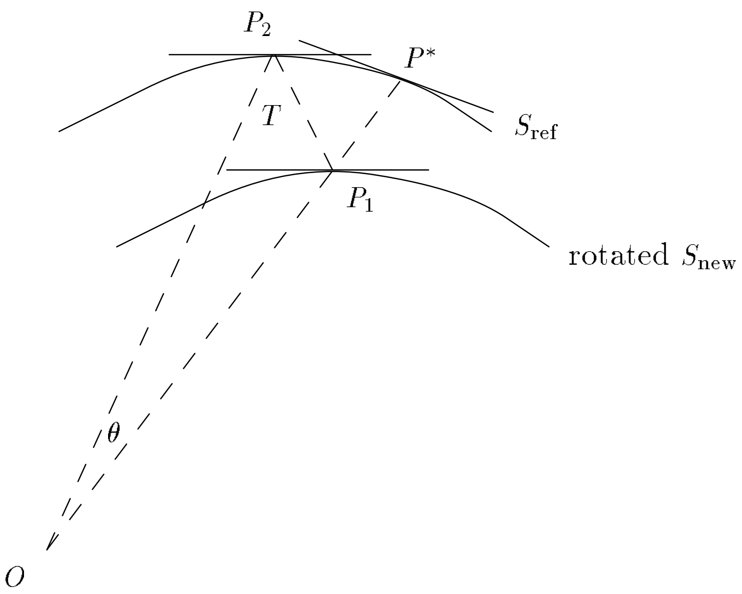


Figure 18: Illustration of approximate point correspondence.

In Section 3.4, we choose an approximate correspondence P^* to replace P_2 as the corresponding point for P_1 (see Fig 18). The pair P_1, P^* is used to form a constraint on the translation vector T as (assuming the rotation is already corrected out):

$$\vec{n}_1 \cdot T \approx \vec{n}_1 \cdot (P^* - P_1). \tag{18}$$

The error in the above approximation is derived as the following. We regard that the reference scan S_{ref} is parameterized with the angle θ as the index parameter. Let $P_2 = (x(0), y(0))$,

$P^* = (x(\theta), y(\theta))$. Then the approximation error is:

$$\begin{aligned} & (P^* - P_1 - T) \cdot \vec{n}_1 \\ = & \ (P^* - P_2) \cdot \vec{n}_2 \\ = & \ (x(\theta) - x(0), y(\theta) - y(0)) \cdot (-y'(0), x'(0)) / \sqrt{x'(0)^2 + y'(0)^2} \\ = & \ (\theta x'(0) + O(\theta^2), \theta y'(0) + O(\theta^2)) \cdot (-y'(0), x'(0)) / \sqrt{x'(0)^2 + y'(0)^2} \\ = & \ O(\theta^2). \end{aligned}$$

Notice that $\theta \leq |T|/|P_1|$. The above error is bounded by $O((|T|/|P_1|)^2)$, which is small when $|T| \ll |P_1|$.

Similarly, we can show that

$$\vec{n}^* \cdot T \approx \vec{n}^* \cdot (P^* - P_1) \tag{19}$$

with an approximation error of $O((|T|/|P_1|)^2)$.

Furthermore, we can combine Eq. 18 and Eq. 19 and yield a better approximated constraint:

$$(\vec{n}_1 + \vec{n}^*) \cdot T \approx (\vec{n}_1 + \vec{n}^*) \cdot (P^* - P_1) \tag{20}$$

whose approximation error can be similarly derived as

$$\begin{aligned} & (P^* - P_2) \cdot (\vec{n}^* + \vec{n}_2) \\ = & \ O(\theta^3) \ = \ O((|T|/|P_1|)^3). \end{aligned}$$

Note that in the above derivation, we assume that the curve (reference scan) is smooth.

C Closed-form Solution for Point-based Matching

For n pairs of points: $P(x_i, y_i), P'(x'_i, y'_i), i = 1, \ldots, n$, a distance function between the transformed P's and the P''s is defined as the following:

$$\begin{aligned} E_{\text{dist}}(\omega, T) &= \sum_{i=1}^{n} |R_\omega P + T - P'|^2 \\ &= \sum_{i=1}^{n} \Big((x_i \cos\omega - y_i \sin\omega + T_x - x'_i)^2 + (x_i \sin\omega + y_i \cos\omega + T_y - y'_i)^2 \Big) \end{aligned}$$

By minimizing E_{dist}, we can obtain a closed-form solution for T_x, T_y, and ω, as given below:

$$\begin{aligned} \omega &= \arctan \frac{S_{xy'} - S_{yx'}}{S_{xx'} + S_{yy'}} \\ T_x &= \bar{x}' - (\bar{x} \cos\omega - \bar{y} \sin\omega) \\ T_y &= \bar{y}' - (\bar{x} \sin\omega + \bar{y} \cos\omega) \end{aligned}$$

where

$$
\begin{aligned}
\bar{x} &= \frac{1}{n}\sum_{i=1}^{n} x_i, & \bar{y} &= \frac{1}{n}\sum_{i=1}^{n} y_i \\
\bar{x}' &= \frac{1}{n}\sum_{i=1}^{n} x_i', & \bar{y}' &= \frac{1}{n}\sum_{i=1}^{n} y_i' \\
S_{xx'} &= \sum_{i=1}^{n}(x_i - \bar{x})(x_i' - \bar{x}'), & S_{yy'} &= \sum_{i=1}^{n}(y_i - \bar{y})(y_i' - \bar{y}') \\
S_{xy'} &= \sum_{i=1}^{n}(x_i - \bar{x})(y_i' - \bar{y}'), & S_{yx'} &= \sum_{i=1}^{n}(y_i - \bar{y})(x_i' - \bar{x}').
\end{aligned}
$$