Scan Registration for Autonomous Mining Vehicles Using 3D-NDT

Martin Magnusson and Achim Lilienthal
AASS
Department of Technology
Örebro University
SE-701 82 Örebro, Sweden
e-mail: martin.magnusson@tech.oru.se, achim@lilienthals.de

Tom Duckett
Department of Computing and Informatics
University of Lincoln
Brayford Pool
Lincoln LN6 7TS UK
e-mail: tduckett@lincoln.ac.uk

Received 24 September 2006; accepted 28 May 2007

Scan registration is an essential subtask when building maps based on range finder data from mobile robots. The problem is to deduce how the robot has moved between consecutive scans, based on the shape of overlapping portions of the scans. This paper presents a new algorithm for registration of 3D data. The algorithm is a generalization and improvement of the normal distributions transform (NDT) for 2D data developed by Biber and Strasser, which allows for accurate registration using a memory-efficient representation of the scan surface. A detailed quantitative and qualitative comparison of the new algorithm with the 3D version of the popular ICP (iterative closest point) algorithm is presented. Results with actual mine data, some of which were collected with a new prototype 3D laser scanner, show that the presented algorithm is faster and slightly more reliable than the standard ICP algorithm for 3D registration, while using a more memory-efficient scan surface representation.

1. INTRODUCTION

The main application considered in this paper is tunnel profiling (that is, measuring and building three-dimensional models) by using a range sensor mounted on drill rigs that are commonly used for tunnel excavation (see Figure 1). Profiling of mine tunnels is necessary to check that new tunnels have

Figure 1. An Atlas Copco drill rig in its natural environment. The vehicle is equipped with rock drills mounted on telescopic booms and is used for drilling holes in the rock face before blasting.

the desired shape, to measure the volume of material removed, to survey old tunnels and investigate whether they are still safe, and to build three-dimensional maps that can be used for autonomous operation of drill rigs and other mining vehicles.

Today's tools for tunnel profile scanning are either very slow or very expensive, and profiling currently needs to be performed separately from any other activity in the tunnel. The rock drill industry has been searching for tools that give a fast and cheap solution to this problem for a long time.

The long-term goal of this work is to make it possible for mining vehicles to operate with minimal human intervention, or completely autonomously. If underground operations could be performed by autonomous vehicles, the lives and health of thousands of mine workers could be saved in the future.

The paper is structured as follows. Section 2 briefly covers the basic algorithms for scan registration that provided the foundation for this work. Section 3 describes the three-dimensional normal distributions transform, a novel algorithm for registration of 3D surfaces, and Section 4 presents some variants and improvements to 3D-NDT. Section 5 gives the results of experiments performed using scan data from an underground mine, and shows a detailed comparison of the algorithms presented in the paper. Finally, Section 6 concludes and summarizes the paper.

2. EXISTING SCAN REGISTRATION ALGORITHMS

Pairwise scan registration is the process of aligning two overlapping scans, given an estimate of the relative transformation needed to match one with the other. When the scans are properly aligned, they are said to be in registration. Several algorithms for this purpose exist, the most common and well known of which is the ICP (iterative closest point) algorithm (Besl & McKay, 1992; Chen & Medioni, 1992). Following the nomenclature of Besl & McKay, the scan that serves as the reference is called the *model* and the scan that is moved into alignment with the model is called the *data* scan.

2.1. ICP

ICP works by iteratively searching for pairs of nearby points in the two scans and minimizing the sum of all point-to-point distances.

Two main problems of ICP are that it is point-based, and as such does not make use of the local surface shape around each point, and that the nearest-neighbor search in the central loop is rather time consuming. One way to speed up the search is to use an efficient search data structure, such as a *k*d-tree with approximate nearest-neighbor search (Greenspan & Yurick, 2003), but the search pass is still the main bottleneck for the algorithm's running time.

If the point pairs that are found in the first step of the algorithm indeed correspond to the same point on the scanned surface, the computed transformation will be exact. However, since the closest point is used as a guess for the corresponding point, it is desirable to detect and filter bad correspondences and keep only the best ones. One strategy is to assign different weights to the pairs, as a kind of "soft" outlier rejection (Rusinkiewicz, 2001). The strategy is to assign more weight to point pairs that are likely to contribute more to the end result and less weight to pairs that are more likely to be incorrect correspondences. One example of a weighting criterion is to use the relative distance between the points. The weight w of the correspondence between points $\mathbf{x}$ and $\mathbf{y}$ can be set proportional to the point pair with the largest distance so that

$$w = 1 - \frac{|\mathbf{x} - \mathbf{y}|}{\max|\mathbf{x}_i - \mathbf{y}_j|}. \quad (1)$$

We found that linear weighting based on distance did not improve the results on our data. For tunnel or corridor data, distance-based weighting can in fact degrade performance. Because most points along the walls and ceiling will generally be well-aligned, their influence will overwhelm point pairs with larger distances, which may correspond to corners and other features that are important. Therefore we weighted all point pairs equally.

2.2. 2D-NDT

The normal distributions transform (NDT) is a more recent method for registration developed for two-dimensional scan data (Biber & Strasser, 2003). The key element in this algorithm is its representation of the model. Instead of using the individual points of the model, it is represented by a combination of normal distributions, describing the probability of finding a surface point at a certain position. The normal distributions give a piecewise smooth representation of the model point cloud, with continuous first and second order derivatives. Using this representation, it is possible to apply standard numerical optimization methods for registration. Numerical optimization is a problem that has been studied for centuries, and fast and reliable methods for optimizing functions such as a sum of normal distributions have been developed and tested over time. Because the points in the model are not used directly for matching, there is no need for the computationally expensive nearest-neighbor search that is done in the central loop of ICP. Storing the NDT representation of scans instead of storing the point clouds themselves also requires much less memory. This is beneficial for all large maps, where storing the complete point cloud data is uneconomical. Another application where a compact map representation is needed is when using multiple time scales for mapping dynamic environments, where multiple copies of the same area are stored, representing different time scales. Computing the normal distributions is a quick one-off task that is done during a single pass through the points of the model.

The first step of the NDT algorithm is to subdivide the space occupied by the model into regularly sized cells (squares in the 2D case, or cubes in 3D). Then, for each cell b that contains more than some minimum number of points, the mean vector $\mathbf{q}$ of the points in the cell and the covariance matrix $\mathbf{C}$ are calculated as

$$\mathbf{q} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k, \quad (2)$$

$$\mathbf{C} = \frac{1}{n-1}\sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{q})(\mathbf{x}_k - \mathbf{q})^T, \quad (3)$$

where $\mathbf{x}_{k=1,\ldots,n}$ are the points contained in the cell.

The probability that there is a point at position $\mathbf{x}$ in cell b can then be modeled by the normal distribution $N(\mathbf{q},\mathbf{C})$. The probability density function (PDF) is formulated as

$$p(\mathbf{x}) = \frac{1}{c}\exp\left(-\frac{(\mathbf{x}-\mathbf{q})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{q})}{2}\right), \quad (4)$$

where $\mathbf{q}$ and $\mathbf{C}$ are the mean vector and covariance matrix for the cell that contains point $\mathbf{x}$, and c is a normalizing constant that can be set to one for practical purposes. Setting the limit for which cells are considered occupied to five points per cell is reasonable, in order to get a sensible covariance matrix. A 2D laser scan and its corresponding normal distributions are shown in Figure 2.

The parameters to be optimized—that is, the rotation and translation of the current pose estimate—can be encoded in a vector $\mathbf{p}$. For 2D registration, there are three transformation parameters to optimize. Let $\mathbf{p} = [t_x, t_y, \phi]$ be the parameter vector, where t_x and t_y are the translation parameters and ϕ is the rotation angle. Using counter-clockwise rotation, the 2D transformation function is

$$T_3(\mathbf{p},\mathbf{x}) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}\mathbf{x} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (5)$$

The algorithm measures the fitness of a particular pose by evaluating the PDFs at all points of the data scan. Since optimization problems are generally formulated as minimization problems, the score function is defined so that good parameters yield a large negative number.

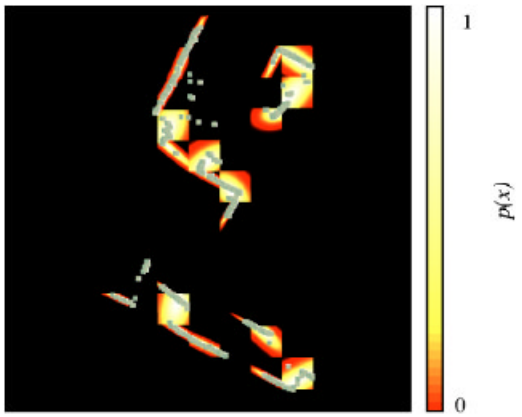

Figure 2. A 2D laser scan of part of a room and the NDT representation describing the surface shape. The original point cloud is shown with small squares, and the rounded shapes show the normal distributions of the occupied grid cells. Each cell is a square with 1 m side length. Brighter areas represent a higher probability.

Given a set of points $\mathcal{X}=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$, a pose $\mathbf{p}$, and a transformation function $T(\mathbf{p},\mathbf{x})$ to transform a point in space, the score $s(\mathbf{p})$ for the current set of parameters is defined as

$$s(\mathbf{p}) = -\sum_{k=1}^{n} p(T(\mathbf{p},\mathbf{x}_k)). \tag{6}$$

In other words, the score is the negated sum of probabilities that the transformed points of the data scan are actually lying on the model surface.

Given the transformation parameters $\mathbf{p}$, Newton's algorithm can be used to iteratively solve the equation $\mathbf{H}\Delta\mathbf{p}=-\mathbf{g}$, where $\mathbf{H}$ and $\mathbf{g}$ are the Hessian and gradient of s. The increment $\Delta\mathbf{p}$ is added to the current estimate of the parameters in each iteration, so that $\mathbf{p}\leftarrow\mathbf{p}+\Delta\mathbf{p}$.

For brevity, let

$$\mathbf{x}' \equiv T(\mathbf{p},\mathbf{x}) - \mathbf{q}. \tag{7}$$

In other words, $\mathbf{x}'$ is the transformed point $\mathbf{x}$, relative to the center of the point distribution of the cell to which it belongs. The entries for the gradient of the score function can be written as

$$g_i = \frac{\delta s}{\delta p_i} = \sum_{k=1}^{n} \mathbf{x}_k'^T \mathbf{C}^{-1} \frac{\delta \mathbf{x}_k'}{\delta p_i} \exp\left(\frac{-\mathbf{x}_k'^T \mathbf{C}^{-1} \mathbf{x}_k'}{2}\right). \tag{8}$$

The entries of the Hessian are

$$H_{ij} = \frac{\delta s}{\delta p_i \delta p_j} = \sum_{k=1}^{n} \exp\left(\frac{-\mathbf{x}_k'^T \mathbf{C}^{-1} \mathbf{x}_k'}{2}\right)\left(\left(\mathbf{x}_k'^T \mathbf{C}^{-1} \frac{\delta \mathbf{x}_k'}{\delta p_i}\right) \times \left(-\mathbf{x}_k'^T \mathbf{C}^{-1} \frac{\delta \mathbf{x}_k'}{\delta p_j}\right) + \mathbf{x}_k'^T \mathbf{C}^{-1} \frac{\delta^2 \mathbf{x}_k'}{\delta p_i \delta p_j} + \frac{\delta \mathbf{x}_k'^T}{\delta p_j} \mathbf{C}^{-1} \frac{\delta \mathbf{x}_k'}{\delta p_i}\right). \tag{9}$$

The first- and second-order partial derivatives of $\mathbf{x}'$ in Eqs. (8) and (9) depend on the transformation function. Using the 2D transformation function from Eq. (5), the first-order derivative $\delta\mathbf{x}'/\delta p_i$ is given by column i of the Jacobian matrix

$$\mathbf{J}_3 = \begin{bmatrix} 1 & 0 & -x_1' \sin\phi - x_2' \cos\phi \\ 0 & 1 & x_1' \cos\phi - x_2' \sin\phi \end{bmatrix}, \tag{10}$$

and the second-order derivatives are

$$\frac{\delta^2 \mathbf{x}'}{\delta p_i \delta p_j} = \begin{cases} \begin{pmatrix} -x_1' \cos\phi + x_2' \cos\phi \\ -x_1' \sin\phi - x_2' \cos\phi \end{pmatrix} & \text{if } i=j=3, \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{otherwise.} \end{cases} \tag{11}$$

The NDT algorithm for registering two point sets $\mathcal{X}$ and $\mathcal{Y}$ (finding the pose $\mathbf{p}$ that moves the data scan $\mathcal{X}$ into registration with the model $\mathcal{Y}$) is given in Algorithm 1.

In recent work carried out independently, a semi-3D version of NDT was used to register large high-resolution outdoor scans (Ripperda & Brenner, 2005). In the work of Ripperda and Brenner, each 3D scan was divided into several horizontal slices and 2D-NDT was applied on each pair of slices. Using N

slices, and denoting the score for slice n by $s_n(\mathbf{p})$, the score function was defined as the sum over all slice pairs

$$s(\mathbf{p}) = \sum_{n=1}^{N} s_n(\mathbf{p}). \tag{12}$$

The approach used by Ripperda and Brenner can only perform registration in the horizontal plane, and only works under the assumption that the local coordinate systems of all scans are aligned in the plane, meaning that the scanner must have the same orientation at each scan pose. This assumption does not hold for the majority of mobile robot applications.

Algorithm 1 Register data scan $\mathcal{X}$ with model $\mathcal{Y}$ using NDT

Build cell structure $\mathcal{B}$
for all points $\mathbf{y}_i \in \mathcal{Y}$ **do**
  Find the cell b_k that contains $\mathbf{y}_i$
  Store $\mathbf{y}_i$ in b_k
end for
for all cells $b_i \in \mathcal{B}$ **do**
  $\mathcal{Y}' = \{\mathbf{y}'_1, \ldots, \mathbf{y}'_n\} \leftarrow$ all points in b_i
  $\mathbf{q}_i \leftarrow \frac{1}{n}\Sigma_{j=1}^{n}\mathbf{y}'_j$
  $\mathbf{C}_i \leftarrow$ covariance of all points in $\mathcal{Y}'$
end for
while not converged **do**
  $score \leftarrow 0$
  $\mathbf{g} \leftarrow 0$
  $\mathbf{H} \leftarrow 0$
  for all points $\mathbf{x}_i \in \mathcal{X}$ **do**
    Find the cell b_k that contains $T(\mathbf{p}, \mathbf{x}_i)$
    $\mathbf{x}'_i \leftarrow T(\mathbf{p}, \mathbf{x}_i)$
    $score \leftarrow score - p(\mathbf{x}'_i)$ (see Eq. (4))
    Update $\mathbf{g}$ (see Eq. (8))
    Update $\mathbf{H}$ (see Eq. (9))
  end for
  Solve $\mathbf{H}\Delta\mathbf{p} = -\mathbf{g}$
  $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$
end while

2.3. Registration with Approximants to the Distance Function

Mitra et al. presented an approach to 3D scan registration that is similar to NDT (Mitra, Gelfand, Pottmann & Guibas, 2004). The idea behind their algorithm is to describe the model surface implicitly, using quadratic approximants to the squared distance function from the target surface, instead of the normal distributions used by NDT or the original point cloud data used by ICP. Registration then becomes the task of minimizing the sum of the distance functions when evaluated at the points of the data scan. Because the approximants used in their algorithm are second-order approximations of the local surface shape that are valid within an interval around each point, and not just at the points where they are computed, it is possible to use Newton iteration to solve the registration problem with this surface representation, too. One way to use the approximants is to compute them on demand for each point in the model, using the normal vector and the two principal curvature directions at that point. The normal and principal curvature vectors are computed in a preprocessing step, and the distance functions are computed at each step of the registration process. The other method presented by the authors is to subdivide the space occupied by the model into a grid. For each grid cell (both cells that are occupied by the surface and empty cells), a quadratic patch is fitted to the squared distance to the scan surface. The second method is quite similar to the NDT versions described in this paper. For any point in the data scan, the algorithm queries the cell structure for the corresponding approximant to the squared distance function to the surface and uses these values as the "score" of the current pose.

The squared distance function used by Mitra et al. is in fact a generalization of the error metrics used by the most common versions of ICP: the point-to-point distance mentioned in Section 2.1, and the point-to-plane distance, which measures the distance from a point in the data scan to the closest point on the tangential plane of its closest neighbor in the model. In their paper, they showed that the suggested functions lead to more reliable registration from a larger number of initial pose estimates than point-to-plane ICP. The algorithm behaves like point-to-point ICP (stable with regard to the initial error, but slower) when the scans are far from each

other, and like point-to-plane ICP (faster, but less stable with regard to the initial error) when the scans are almost registered.

The quadratic patches approximate both the position and the curvature of the surface, while the normal distributions used in NDT only give an estimate of the position. As long as the surface is smooth and the cells are small enough so that the surface is approximately uni-modal within each cell, quadratic patches are a more descriptive representation of the surface than the normal distribution of points within the cell. Mitra et al. use the fitting error of the quadratic patch to deal with the problem of choosing a good cell size, by building an octree cell structure that has small cells where required and large cells where that is sufficient. Neighboring cells are merged if a patch fitted to the surface in the larger cell has an acceptable fitting error. A similar method has also been implemented for NDT (see Section 4.3.2).

For very noisy data, we hypothesize that surface patches are an inappropriate model of the scan data, compared to the more context-free normal distribution representation. The quadratic patches assume that the scan points are sampled from a piece-wise smooth surface, which is not always the case. In the mine mapping application, the walls of the tunnels are quite rough, and the the sample spacing is at a larger scale than the surface roughness for areas of the tunnel far away from the scanner. Using only the scan points or an approximated surface fitted to the scan points is likely to lead to misalignment of scans proportional to the roughness of the walls. The uneven walls will in this case behave like noisy measurements. Smoothing the surface with the proposed normal distributions is a good alternative in that case. Mitra et al. did not report the execution times of their algorithm, but it would be interesting to compare the speed and accuracy of their approach to that of NDT. We did not compare the two algorithms for the work presented here, because of time constraints and the lack of a publicly available implementation. Though the storage requirements for the quadratic fit representation are smaller than storing the point clouds themselves—at least for densely sampled point clouds—they are somewhat larger than for NDT, because distance approximants are stored for all cells (requiring nine parameters per cell), and not just the occupied ones.

NDT之后最显著的特点就是连续、可微；可以进行泰勒展开，从某种角度来说，对函数进行了"平滑"

3. 3D-NDT

The main difference between 2D and 3D registration with NDT lies in the spatial transformation function $T(\mathbf{p},\mathbf{x})$ and its partial derivatives. In two dimensions, rotation is represented with a single value for the angle of rotation around the origin, and the most obvious transformation function is the one from Eq. (5). General rotation in 3D is more complex. A robust 3D rotation representation requires both an axis and an angle. A straightforward way to represent a general 3D transformation is to use seven parameters (three parameters for the translation, three for the rotation axis, and one for the rotation angle). Using a right-handed coordinate system and counter-clockwise rotations, the transformation of a 3D point $\mathbf{x}$ using a parameter vector $\mathbf{p}$ can then be formulated as

$$T_7(\mathbf{p},\mathbf{x}) = \begin{bmatrix} er_x^2 + c & er_x r_y - sr_z & er_x r_z + sr_y \\ er_x r_y + sr_z & er_y^2 + c & er_y r_z - sr_x \\ er_x r_z - sr_y & er_y r_z + sr_x & er_z^2 + c \end{bmatrix} \mathbf{x} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \tag{13}$$

where $\mathbf{p}=[\mathbf{t}|\mathbf{r}|\phi]$, $\mathbf{t}=[t_x,t_y,t_z]$ is the translation, $\mathbf{r}=[r_x,r_y,r_z]$ is the axis of rotation, $s=\sin\phi$, $c=\cos\phi$, $e=1-\cos\phi$, and ϕ is the rotation angle.

A common way to represent 3D rotation in computer graphics is to use quaternions, which are a generalization of complex numbers. Quaternions have favorable properties when used for rotation, most notably when composing several rotations. A normalized quaternion always represents a valid rotation. A combination of rotation matrices, on the other hand, may become nonorthogonal as rounding errors increase over time, and using a nonorthogonal transformation matrix for rotation has undesired effects. The axis-angle rotation $\mathbf{r}$, ϕ can be represented by the quaternion $\cos\phi+(r_x\cos\phi)i+(r_y\cos\phi)j+(r_z\cos\phi)k$.

The partial derivatives that are needed for Eqs. (8) and (9) when using T_7 can be found in the Jacobian and Hessian matrices (17) and (18). The Hessian is presented as a block matrix with 7×7 blocks, where each block is a three-element vector. Similarly to Eq. (7), define

$$\mathbf{x}' \equiv T_7(\mathbf{p},\mathbf{x}) - \mathbf{q}, \tag{14}$$

where $\mathbf{x}$ is a 3D scan point, $\mathbf{q}$ is the mean vector of the cell in which it lies, and $\mathbf{p}$ is a vector of transformation parameters. Then

$$\frac{\delta \mathbf{x}'}{\delta p_i} = \text{the } i\text{th column of } \mathbf{J}_7, \tag{15}$$

$$\frac{\delta^2 \mathbf{x}'}{\delta p_i \delta p_j} = \mathbf{H}_{ij}, \tag{16}$$

$$\mathbf{J}_7 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ e(2r_x x_1 + r_y x_2 + r_z x_3) & er_y x_1 - sx_3 & er_z x_1 + sx_2 \\ er_x x_2 + sx_3 & e(r_x x_1 + 2r_y x_2 + r_z x_3) & er_z x_2 - sx_1 \\ er_x x_3 - sx_2 & er_y x_3 + sx_1 & e(r_x x_1 + r_y x_2 + 2r_z x_3) \\ sA - cB & sC - cD & sE - cF \end{bmatrix}^T \tag{17}$$

$$A = (r_x^2 - 1)x_1 + r_x r_y x_2 + r_x r_z x_3, \quad B = r_z x_2 - r_y x_3,$$

$$C = r_x r_y x_1 + (r_y^2 - 1)x_2 + r_y r_z x_3, \quad D = -r_z x_1 + r_x x_3,$$

$$E = r_x r_z x_1 + r_y r_z x_2 + (r_z^2 - 1)x_3, \quad F = r_y x_1 - r_x x_2$$

$$\mathbf{H}_7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} \\ 0 & 0 & 0 & \mathbf{b} & \mathbf{e} & \mathbf{f} & \mathbf{g} \\ 0 & 0 & 0 & \mathbf{c} & \mathbf{f} & \mathbf{h} & \mathbf{i} \\ 0 & 0 & 0 & \mathbf{d} & \mathbf{g} & \mathbf{i} & \mathbf{j} \end{bmatrix}, \tag{18}$$

$$\mathbf{a} = \begin{bmatrix} 2ex_1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} ex_2 \\ ex_1 \\ 0 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} ex_3 \\ 0 \\ ex_1 \end{bmatrix},$$

$$\mathbf{d} = \begin{bmatrix} s(2r_x x_1 + r_y x_2 + r_z x_3) \\ sr_y x_1 - cx_3 \\ sr_z x_1 + cx_2 \end{bmatrix},$$

$$\mathbf{e} = \begin{bmatrix} 0 \\ 2ex_2 \\ 0 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} 0 \\ ex_3 \\ ex_2 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} sr_x x_2 + cx_3 \\ s(r_x x_1 + 2r_y x_2 + r_z x_3) \\ sr_z x_2 - cx_1 \end{bmatrix},$$

$$\mathbf{h} = \begin{bmatrix} 0 \\ 0 \\ 2ex_3 \end{bmatrix}, \quad \mathbf{i} = \begin{bmatrix} sr_x x_3 - cx_2 \\ sr_y x_3 + cx_1 \\ s(r_x x_1 + r_y x_2 + 2r_z x_3) \end{bmatrix},$$

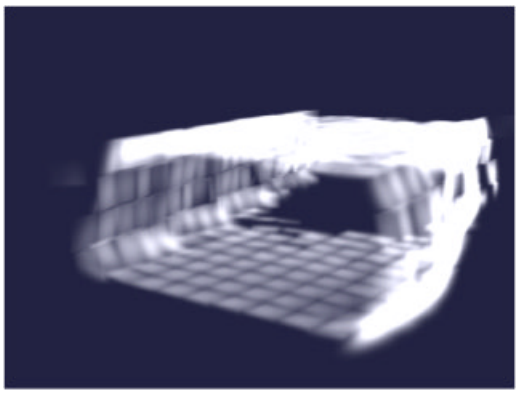

Figure 3. The probability functions used by 3D-NDT for a tunnel section. Brighter, denser parts represent higher probabilities. The cells have a side length of 1 m.

$$\mathbf{j} = \begin{bmatrix} cA + sB \\ cC + sD \\ cE + sF \end{bmatrix}.$$

In (17) and (18), x_n denotes the scalar nth component of the 3D vector $\mathbf{x}_k$. Figure 3 illustrates the 3D normal distributions for a mine tunnel scan.

The equations above were implemented for the experiments in Section 5. However, the angle parameter in the seven-element parameter vector is redundant. The angle can also be encoded implicitly in the three axis parameters, so that the length of the rotation axis corresponds to the angle of rotation, instead of maintaining a normalized rotation axis separately. In that case, only six parameters are needed.

4. ALTERNATIVE METHODS IMPLEMENTED

Several choices need to be made for a practical implementation of 3D-NDT. This section describes different methods and parameters that were tested, and their influence on the basic algorithm.

4.1. Sampling Method

When using 3D-NDT, the model is converted to a set of normal distributions. The points of the data scan are then aligned to these functions. Usually, a large number of scan points are redundant for the purpose of describing the scanned surface shape. Therefore, it is normally desirable to subsample the data scan in order to improve running time. In many cases, not least when scanning in corridors and tunnels, as well as in unstructured outdoor environments, the distribution of points is very much denser near the scanner location than farther out. If points are sampled in a uniformly random manner, the sampled subset will have a similar distribution. Consequently, parts that are further from the scanner contribute less to the registration. This is not specific to NDT, but is common to all point-based registration methods.

Spatially distributed sampling is a sensible alternative method, that is, making sure that the spatial distribution of points in the subsample is as even as possible. This can be done by grouping the points into equally sized cells, similar to what is done when the normal distributions are generated for the model. Then, a number of points are drawn from each cell. If the distribution of cells is adequate, this strategy will give an even distribution of points.

It is also possible to implement subsampling methods that consider the normals as well as positions of points, either making the distribution of normals as spread out as possible or primarily choosing points with "unusual" normals (Rusinkiewicz, 2001). The normal at each surface point must then be computed from a sufficient number of its neighbors. Gelfand et al. developed an improved sampling method for ICP, mainly for cases when the data consist of mostly planar regions with a few important "lock and key" features (Gelfand, Ikemoto, Rusinkiewicz & Levoy, 2003). Such data are notoriously difficult to register correctly, since the scans can "slide" along the planar regions without any big changes in the error function. The stable sampling method of Gelfand et al. requires that normals are computed for all sample points. They reported that the algorithm takes about three times longer to execute than ICP with uniform subsampling. In the work covered by this paper, point clouds without normal or connectivity information have been used, so these kinds of sampling methods have not been investigated in detail. Though it was not tested, we believe that most of the mine tunnel scans do not have the kind of shape that the stable ICP sampling method was designed for. While many of the scan pairs used in Section 5 are difficult to register for the same reasons, namely that the large-scale features are not enough for accurate registration, the important small-scale features are not generally as distinct as in the incised plane data sets used by Gelfand et al., but are more evenly distributed over the rough surface and have characteristics similar to Gaussian noise.

4.2. Cell Size

Choosing a good cell size is important when using NDT. Any feature that is much smaller then the size of a cell will be blurred out by the PDF that describes the local surface shape around it. Choosing a cell size that is too large therefore often leads to less accurate registration. On the other hand, the region of influence of a cell only extends as far as its boundaries. That is, the cell will only contribute to the score function for scan points within its bounds. The consequence of this is that if the cells are too small, the two scans must be close together before registration for the algorithm to succeed. Using smaller cells

also requires more memory. The optimal size and distribution of cells depend on the shape of the input data and on the application.

4.3. Discretization Methods

Using a fixed lattice of square or cubic cells burdens the user with the task of choosing a good cell size. A more adaptive cell structure would be preferable, using finer subdivision in places where a single normal distribution cannot describe the surface satisfyingly. This section presents a number of alternative methods for handling the cells and their PDFs.

4.3.1. Fixed Subdivision

The benefit of using a fixed lattice of cells is that the overhead for initializing the cell structure is very small. Only one set of PDF parameters needs to be computed for each cell, and the positioning of each cell is straightforward. Even more important for the performance of the algorithm is that point-to-cell look-up is also a very quick operation that can be done in constant time, as the cells can be stored in a simple array.

4.3.2. Octree Subdivision

An octree is a tree structure that can be used to store a hierarchical discretization of 3D space. In an octree, each node represents a bounded partition of the space. Each internal node has eight children that represent congruent and non-overlapping subdivisions of the space partition that corresponds to their parent node. When creating an octree, the root node is sized to encompass the whole model. The tree is then built recursively, splitting all nodes containing more than a threshold number of points. All data points are contained in the leaf nodes of the octree.

The "octree" version of 3D-NDT starts with fixed regular cells, as described before, with the difference that each cell is the root node of an octree. Each cell in which the spread of the distribution is larger than a certain threshold is then recursively split, thus making a forest of octrees. It is important for the efficiency of the algorithm that the point-to-cell look-up is fast, and this is the main reason why a forest of octrees was implemented, rather than having a single octree with a root node that spans the whole scan. For many types of scan data, a reasonable cell size can be specified, so that only a few cells in parts where the scan surface is particularly uneven need to be split. Thus, for most points, finding the correct cell only needs a single array access, while traversing a large octree once for each point would take more time. Using a forest gives a very slight increase in memory consumption, since a few unnecessary cells need to be stored, but the effect of this is negligible.

When traversing the cell structure looking for the corresponding cell to a point in the data scan, the leaf node that contains the point is chosen and its PDF is used to compute the score function.

4.3.3. Additive Subdivision

Using octree subdivision gives a better representation of the surface shape in areas where large cells would hide many details, while keeping large cells where the surface is largely planar and further subdivision is unnecessary. However, the problem that small cells have a smaller region of influence remains: if corresponding points of the two scans are not within the same cell, the extra fidelity is of no use.

A slight change to the octree subdivision scheme can mitigate this limitation. Instead of using only one leaf of the octrees, each point from the data scan has its score function evaluated for all of the distributions in the leaf cells. This effectively increases the support size of the leaf cells to that of their root cell, without sacrificing the extra refinement of the surface description that they give. This is illustrated in Figure 4.

4.3.4. Iterative Subdivision

Another option is simply to perform a number of NDT runs with successively finer cell resolution, so that the start pose for each iteration other than the first one is the end pose of the previous run. The first runs are good for bringing badly aligned scans closer together, and later runs improve the rough initial match.

If the different cell structures are computed from the smallest cell size to the largest and the cell sizes are changed by a factor 2 in each iteration, the larger cells do not need to be computed from scratch, but can be updated efficiently using the data from their subcells. This method is a potential improvement to how the implementation shown in Section 5.2 was done.

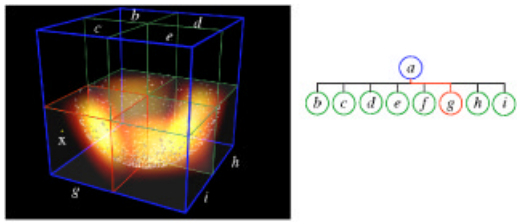

Figure 4. Comparing octree and additive subdivision. A subdivided grid cell is shown on the left, and the tree structure is shown on the right. The PDF of cell a has a large spread, because the points within the cell are not aligned along a planar region. Therefore it is split, and the PDFs of eight subregions b–i are computed instead. Point $\mathbf{x}$ is within cell a, and, more specifically, within subcell g. Using octree subdivision, $\mathbf{x}$'s contribution to the score function is computed from g alone. Using additive subdivision, the score is a sum computed from nodes b–i. In this example, nodes b–e are empty and will not add anything to the score.

During the preparation of this paper, Takeuchi and Tsubouchi presented another way of using NDT for 3D scan registration (Takeuchi & Tsubouchi, 2006). Their implementation is rather similar to the version described in this paper in that they also use an iterative subdivision scheme similar to that described here. An important difference is that they used smaller cells in the space that is near the sensor location and larger cells farther away in the early iterations, and used only the smaller size in the later iterations, when the scans were almost aligned. The reasoning behind this is that error in the rotation estimate caused larger displacements further from the sensor location, so larger cells are needed there to make sure that more points from the data scan are used. The linked cells strategy described in Section 4.3.5 is another solution to the same problem. Takeuchi and Tsubouchi tested their algorithm on data from a computer lab with good results, though they did not make a direct comparison of their algorithm with other registration algorithms.

4.3.5. Linked Cells and Infinite Outer Bounds

Using the discretization methods described so far, points from the data scan lying in unoccupied cells are discarded, thus rendering large parts of the input space "dead." Instead of doing so, the PDF from the closest occupied cell can be used for those points.

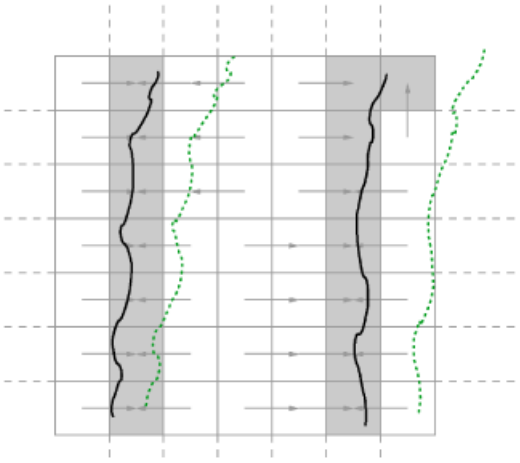

Figure 5. Matching two 2D scans of a tunnel section. The dotted scan is being registered to the solid scan. Occupied cells are shaded. If linked cells are not used, the parts of the scan that are in unshaded cells will be skipped. Otherwise the linked cell (shown with arrows) will be used. If using infinite outer bounds, the outer cells extend as shown with dashed lines.

This increases the region of influence of cells and is illustrated in Figure 5. Even though the value of the PDF of many cells is almost zero outside the cell bounds, so that it makes no substantial contribution to the score anyway, for cells with a very elongated point distribution, the influence outside the cell can also make a difference.

The same idea can also be applied to points falling outside of the cell lattice altogether. The score for those points can be computed using the closest cell on the edge of the lattice, so that the outer cells in effect have infinite outer bounds. However, doing so introduces a certain "drag" bias, as points from non-overlapping regions of the data scan will be attracted to border regions of the model.

Linked cells can be implemented either by letting each cell store a pointer to the nearest occupied cell, or by storing only occupied cells and putting them in a kd-tree. The latter should be preferable if there are many unoccupied cells.

5. EXPERIMENTS

This section covers experiments performed with underground mine data to compare the performance of different varieties of 3D-NDT and ICP.

There are many parameters that can be changed, both for ICP and 3D-NDT. To avoid a combinatorial explosion in the number of possibilities, a default "baseline" combination of variants was chosen that incorporates the following features:

1. ICP parameters
   - Euclidean point-to-point distance error metric,
   - outlier rejection using a 1 m fixed distance threshold,
   - least squares optimization (Besl & McKay, 1992),
   - approximate *k*d-tree search structure with 10 points per leaf node in the tree (to minimize the amount of back-tracking needed) and 1 cm error bound (for each search, a point that is no more than 1 cm from the true nearest neighbor is returned),
   - constant weighting of point pairs.
2. NDT parameters
   - fixed cells with 1 m side length,
   - Newton's method with line search for optimization, with a maximum step length of 0.05 ($|\Delta\mathbf{p}|=0.05$) so that the maximum change in the pose vector is 5 cm or 0.05 rad at each iteration,
   - neither linked cells nor infinite outer bounds.
3. Common parameters
   - convergence threshold when the change of $|\mathbf{p}|$ is less than 0.0001.

The times reported in this paper include all required preprocessing: creating a *k*d-tree (for ICP), building the cell structure and computing all PDFs (for NDT), and subsampling the data scan (for both algorithms).

Moderate effort was made to optimize the efficiency of the programs. The algorithms were implemented in C++. The ICP implementation uses the quite efficient approximate nearest neighbor library ANN. The numerical optimization code used in 3D-NDT makes use of the C linear algebra library newmat. This library claims to be most efficient for large matrices, but the matrices involved in the computations for 3D-NDT are no larger than 7×7. It is likely that the numerical optimization can be performed faster. The experiments were run on a computer with an AMD Athlon processor running at 1950 MHz and 512 MB of memory.

Figure 6. One of the tunnels in the Kvarntorp mine.

5.1. Data

Three mine data sets were used in the comparison and evaluation of the registration algorithms. They were collected in the Kvarntorp mine, south of Örebro in Sweden. This mine is no longer in production, but was once used to mine sandstone. The mine consists of more than 40 km of tunnels, all in one plane. Parts of the mine are currently used as archives and storage facilities, while others are used as a test bed for mining equipment.

Because of the natural layers of sandstone, the tunnels have a rather characteristic shape, with flat ceilings and relatively straight walls. Even though the floor and ceiling are flat compared to many other mines and natural environments, the unevenness of the floor makes a wheeled vehicle tilt considerably while driving over it. The roughness is comparable to that of a gravel road, and if scans were being registered with only three degrees of freedom (disregarding tilt and changes in floor height), there would be large discrepancies between many scans. Figure 6 shows a photo from one of the tunnels.

The JUNCTION data set (Figure 7) consists of two scans from the end section of a tunnel. At the far end of the tunnel, there is a flat cast concrete structure, and on one of the side walls there is a passage to a neighboring tunnel. Both the end face and this passage are salient and large-scale features. These two scans were taken from the same pose, and only differ in resolution. In other words, the ground truth pose for the data scan with respect to the model is $\mathbf{t}=\mathbf{0}$, $R=([0,0,1],0)$. The data scan contains 139 642 points and the model contains 72 417 points.

The TUNNEL data set (Figure 8) was collected further down the same tunnel. Two scans were taken

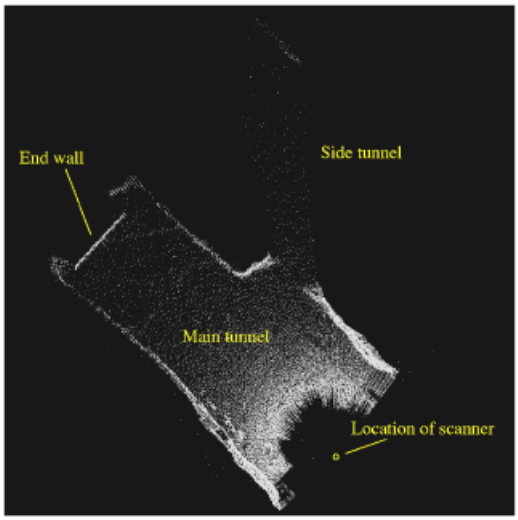

Figure 7. The data scan scan from the JUNCTION data set, seen from above.

approximately 4 m apart. The scans contain around 27 500 points each. The scans in this set have much less obvious features. The only large-scale features—the walls and ceiling—are not enough to give accurate registration, as the scans can "slide" along the direction of the tunnel, and still have a large amount of overlap and close proximity of all surfaces, which are the usual criteria for a good match. The small-scale features, such as bumps on the walls and light fixtures in the ceiling, need to be matched in order to properly register these scans. For these two scans, the ground truth was determined visually, by running a number of registration attempts and picking one that looked like the closest match. When collecting this data set, we tried to measure the relative displacement between the scans using a so-called total station (that is, a tripod mounted laser measurement device). The total station can be seen in Figure 6 (on a yellow tripod near the left wall). Three points were marked on the scanner, and the total station was set up at a fixed position further down the tunnel. The distances to the three points on the scanner were measured from each scanner pose, and the transformation from each scanner pose to the next was determined from these data. The resulting measurements were not accurate enough to use as a ground truth measurement, but they were good enough to provide an initial estimate for the registration algorithms.

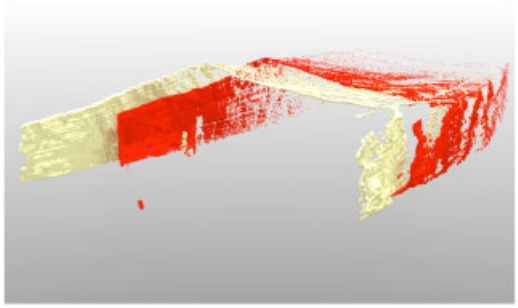

Figure 8. The two scans of the TUNNEL data set. The free-floating points in the middle of the tunnel are noise.

Both the JUNCTION and TUNNEL data sets were collected with an early prototype of a 3D laser range finder, built by Optab Optronikinnovation AB. The Optab scanner is based on a modulated infra-red laser that is projected onto a rotating mirror. The range is measured by investigating the phase-shift of the reflected laser beam. The configuration of the scanner was changed between the two data sets. For the TUNNEL data set, the scanner was oriented so that the first scan plane was horizontal. The scanner was then tilted upwards. This is a so-called pitching scan. Because of this configuration, the floor is not visible in the TUNNEL data. For the JUNCTION data set, the scanner was mounted so that each scan plane was vertical, and the scanner was rotated around the vertical axis. This is known as a yawing scan (Wulf & Wagner, 2003). The Optab scanner is shown in Figure 9.

A larger data set, KVARNTORP-LOOP, was collected at a later date, using a SICK LMS 200 laser scanner mounted on our mobile robot platform called Tjorven (shown in Figure 10). The SICK scanner is a 2D scanner, but was mounted on a pan-tilt unit in order to collect 3D data sets. For the KVARNTORP-LOOP data set, the robot was driven manually along two tunnels, forming a loop, with 3D scans being taken about 4 to 5 m apart. The robot was kept stationary during the scans, so that all points in each scan were taken at the same physical location. The first 65 scans from this set are shown in Figure 11. The scans contain around 95 000 points each. The scanner on Tjorven is configured for pitching scans.

The scans of the KVARNTORP-LOOP data set are more accurate than those of the JUNCTION and TUN-

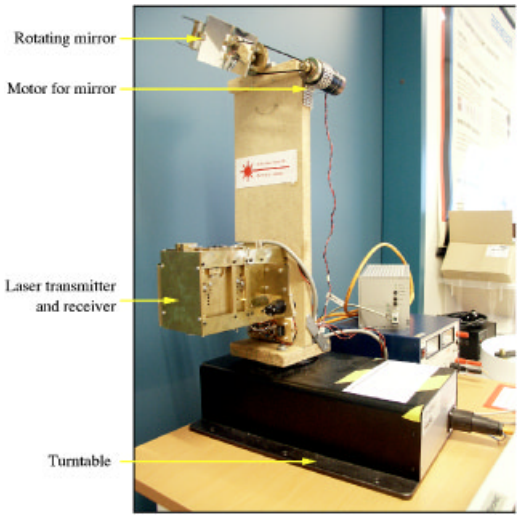

Figure 9. The Optab scanner lab prototype.

NEL sets, which show some disturbances due to the somewhat unstable experimental state of the scanner.

5.2. Results

5.2.1. Results with Single Scan Pairs

To test the performance of the algorithms with respect to different parameter values the two scan pairs of the JUNCTION and TUNNEL data sets were used. A number of registration attempts were run from a set of start poses evenly distributed around the ground truth pose. The magnitudes of the translation and rotation components of the initial pose estimates were kept constant for each batch of tests, but the directions were different for each run. In other words, the translation displacement for each test run was a point on a sphere with a fixed radius. The added rotation error had its axis pointing in a random direction for each run and the angle (that is, the amount of rotation) was fixed for each batch of runs. The pose offsets were taken from a set of points evenly distributed on the unit sphere. The translation error of the initial pose estimate is denoted e_t and the rotation error is denoted e_r.

For these experiments, the following settings were used in the "baseline" setup:

- 10% of the points were sampled from the data scan with even spatial distribution,
- no subsampling of the model (all points were used),

Figure 10. Tjorven, our mobile robot platform. In addition to the laser scanner used for 3D mapping, it is also equipped with a digital camera, an array of sonars, an omnidirectional camera (not shown here), and a differential GPS system.

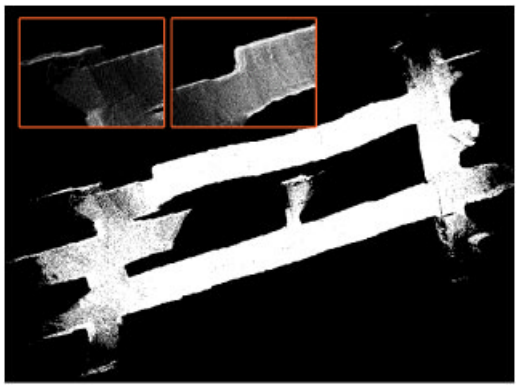

Figure 11. The first 65 scans from the KVARNTORP-LOOP data set, seen from above, after registration with manual intervention where the registration algorithms failed and for the scans without odometry information. The map measures approximately 55 m by 155 m, and is around 6 m high. The traversed distance around the loop is around 330 m. The top left corner shows the accumulated error after coming back to a previously visited location after completing the loop. The error there is about 2.7 m. To the right of this section is a clear "offset" in the tunnel. This is not a registration error, but shows the actual shape of the tunnel. That shape is probably due to a mistake on part of the excavation crew when they were trying to physically "close the loop."

- initial translation error of 1 m,
- initial rotation error of 0.1 rad,
- 100 tests for each set of parameters.

Table I shows the parameters that were manipulated in the experiment. The results of these experiments are presented with box plots, with a line connecting the median values of each set of runs. The box extends to the upper and lower quartile of the data, and the "whiskers" extend to the maximum and minimum values of the sequence. The limits for what is considered a "good match" are shown with dashed horizontal lines. These are not hard limits, but were chosen according to what was considered acceptable for the application and the accuracy with which the ground truth pose could be estimated. If only the median registration errors were shown, 3D-NDT would appear to be far superior to ICP in all cases. Even though the median error was lower when using 3D-NDT for scan registration, there were problems with some outlier poses for which the algorithm did not converge. The box plots show a more complete description of the distribution of the results, showing both how the majority of the runs behaved and the extreme values.

On a similar note, the mean squared point-to-point error is commonly used as a measure of the quality of registration. We did not include these numbers here, as they are not an objective measure of the registration accuracy. The mean squared point-to-point error is exactly the objective function that ICP tries to minimize, and, if that were indeed the best measure of registration accuracy, ICP would be an optimal algorithm and would never fail. We chose instead to determine a ground truth pose for each scan pair and measure the deviance from that pose, with some allowance for what is a close enough match, as described above. The ground truth pose for the JUNCTION data set was zero rotation and translation, since the scanner did not move between the two scans. For the TUNNEL data set the ground truth was determined by running and inspecting a number of registration attempts, and an average of the best matches was used as the ground truth pose.

Sample ratio: To test the sensitivity to the number of samples being used for registration, a number

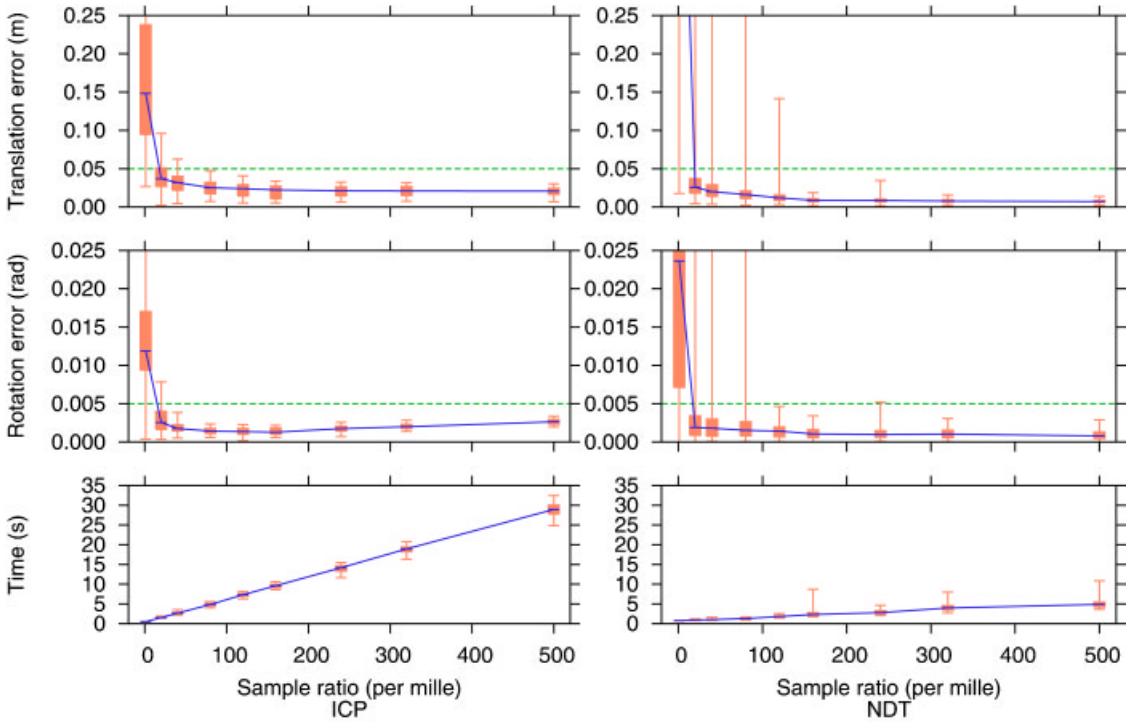

Figure 12. Sample ratio tests for the JUNCTION set.

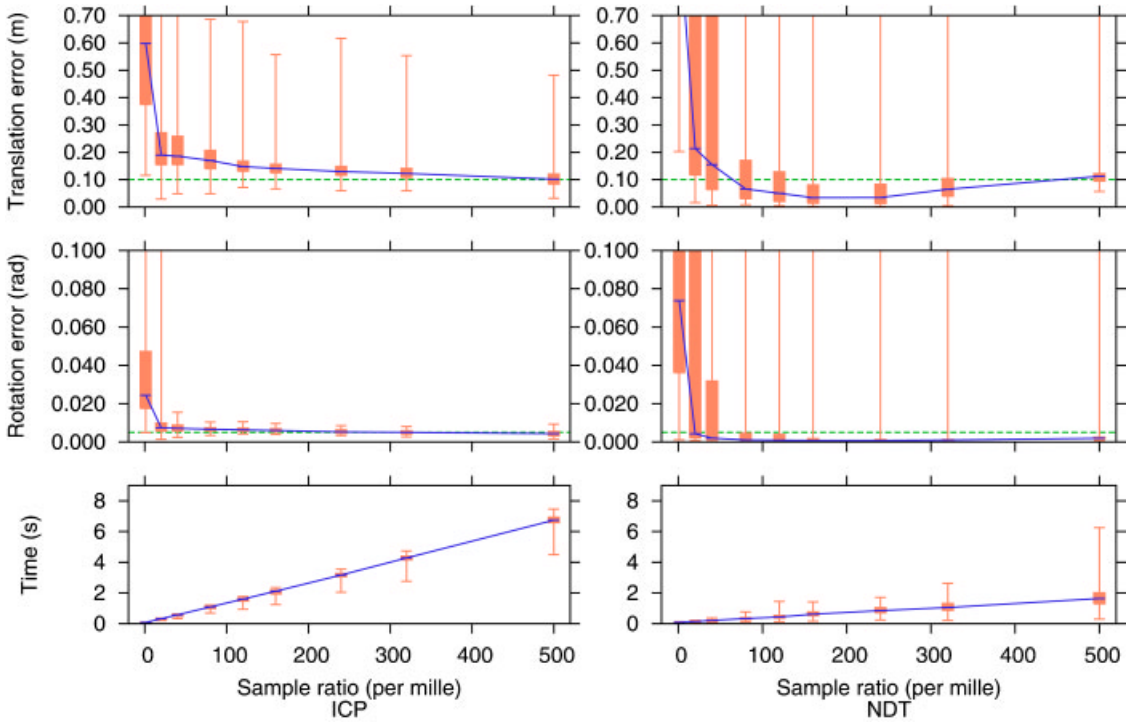

Figure 13. Sample ratio tests for the TUNNEL set, using spatially distributed sampling.

of test sequences were run with increasing numbers of samples. From 0.1% up to 50% of the points in the data scan were sampled and used for matching together with all of the points in the model. Figures 12 and 13 show the results of tests where all other parameters were set according to the baseline setup.

The conclusion is that ICP is less error-prone when using very low sample ratios (less than a few percent), and that the execution time is around three times longer than for 3D-NDT. Even though 3D-NDT succeeds at registering the two scans from most of the start poses tried, it fails for some poses when using a very low sample ratio. Around 10% of the total number of points is enough to give reliable results for the JUNCTION data set when the initial error is moderate. The median error is lower for 3D-NDT in all cases with larger sampling ratios, but there are some outlier cases where the error is much larger. There were failed registrations at up to 12% sample ratio. ICP gives acceptable results down to around 8% for the same data and initial error.

As can be seen from Figure 13, the TUNNEL data set is much more challenging than the JUNCTION set, both for ICP and 3D-NDT. The median error is still smaller for 3D-NDT than for ICP, but, with an initial translation error of 1 m and a rotation error of 0.1 rad, the algorithms fail to register the scans from a rather large number of the initial pose estimates.

Figure 12 shows that the rotation error actually increases for ICP as the sample ratio goes above 20% for the JUNCTION data set. The reason for this could be that more of the scan noise is used, in other words, overfitting. A similar effect can be seen for 3D-NDT on the TUNNEL data set in Figure 13. Because the two scans in this data set are only partially overlapping, ICP tends to move the source scan a bit too much towards the center of the target scan to maximize the amount of overlap. The pose that 3D-NDT converges to when using a high sample ratio is similar to the one that ICP converges to.

If both the data scan and the model are subsampled using the same ratio, and not just the data scan, the required sample ratio is much higher.

Sampling method: Spatially distributed sampling is generally more robust than uniformly random sampling. The results of using a uniform prob-

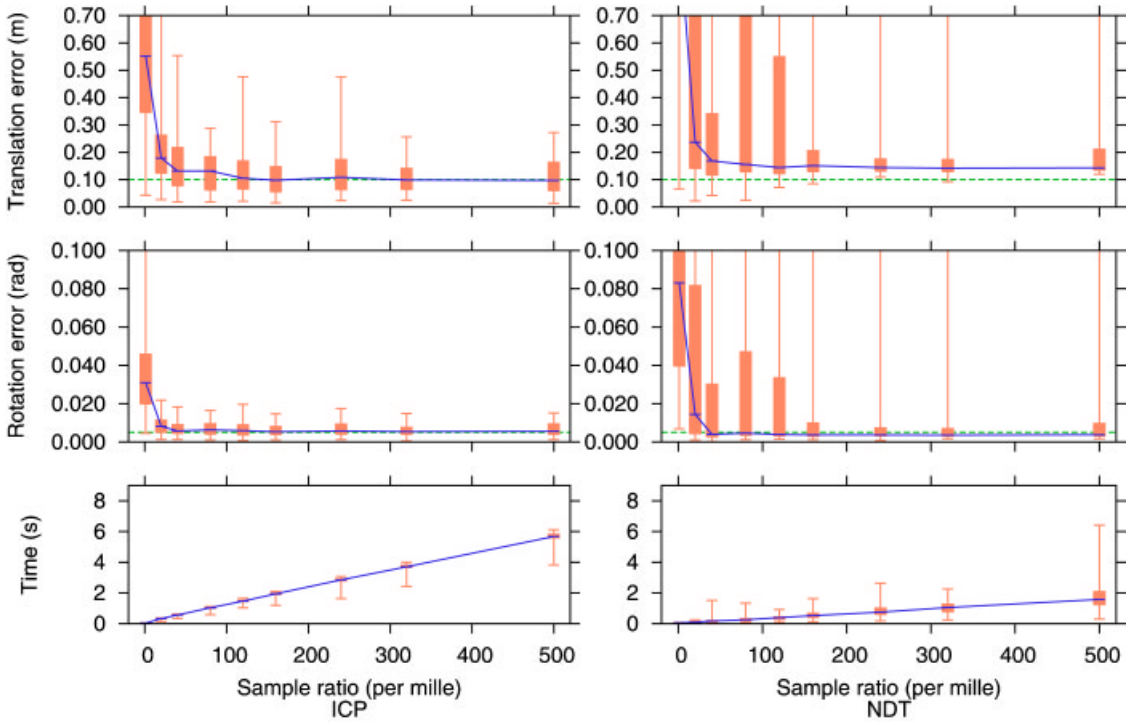

Figure 14. Sample ratio tests for the TUNNEL set with uniform random subsampling instead of spatially distributed selection. The other settings are the same as in Figure 13.

ability distribution when selecting the subset for matching is shown in Figure 14. As discussed earlier, using uniformly random sampling will preserve the general distribution of points in the scan, and that is not optimal for tunnel scans, where the concentration of points is much higher near the sensor location than further away.

Comparing Figures 13 and 14, it can be seen that the registration errors are larger when using uniform random sampling for 3D-NDT. Using spatially distributed sampling with ICP with this data meant that more non-overlapping points were selected. Therefore, it makes sense not to use this sampling method for ICP. For 3D-NDT, the interquartile range was rather large for both sampling methods, but the median translation and rotation errors were significantly lower when using spatially distributed sampling, because the more evenly distributed sampling gives a more representative view of the scan. For the JUNCTION set, uniformly random sampling works almost equally well compared to spatially distributed sampling, however, since the overlap of the data scan and model is 100%.

Table I. The parameters that were manipulated for ICP and NDT on the JUNCTION and TUNNEL and data sets.

Parameter	ICP	3D-NDT
Sample ratio	•	•
Sampling method	•	•
Initial translation error	•	•
Initial rotation error	•	•
Cell size	–	•
Discretization method	–	•

Cell size: To show the effect of different cell sizes for 3D-NDT, registration with sizes ranging from 0.5 m up to 3 m are shown in Figure 15. Each box plot shows the results of 50 test runs.

The running times are shorter when the cells are larger (and fewer). The translation error is at its smallest within a certain cell size range, and increases with both smaller and larger cells. For

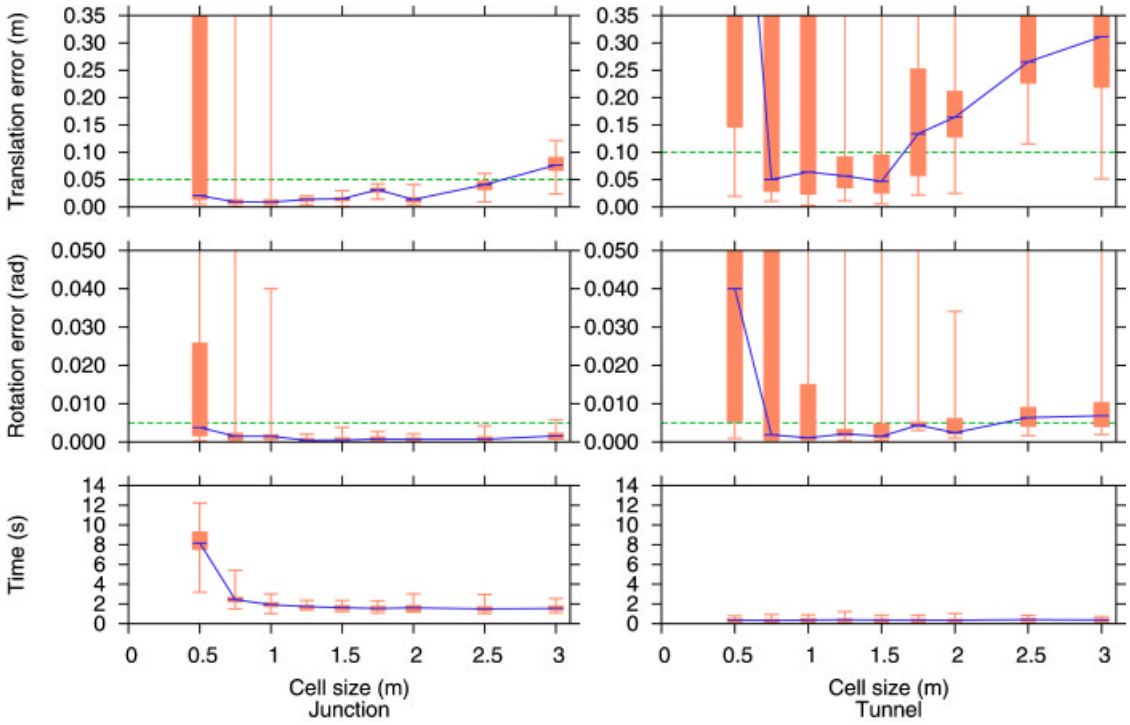

Figure 15. Comparing the effect of 3D-NDT registration with different cell sizes, using fixed cells. Each test sequence is 50 runs. The initial error is $|\mathbf{e}_t|=1$ and $|\mathbf{e}_r|=0.1$.

smaller cells, the algorithm fails to register the two scans from many start poses, because of small regions of influence. This can be seen in Figure 15, where the upper quartile of the tests with JUNCTION and 0.75 m cells is comfortably below the acceptable threshold, but the error of the worst few runs is much larger. This result is due to the fact that, depending on the direction of the initial pose error, for some test runs the small cells will not be able to "attract" enough points. For larger cells, the accuracy decreases because of loss of surface shape information. Based on these results, a cell size of around 1 to 2 m is most suitable for the given environment.

Because simple arrays were used for cell storage (storing both occupied and unoccupied cells), memory usage increased drastically for the tests with the smallest cells. This also led to slower performance because of memory swapping, particularly for the JUNCTION data set. The times reported here were measured with the ANSI C clock() function, which only measures CPU time. The actual time was larger for the tests with 0.5 m cell size. A straightforward way to fix this problem would be to store the cells in a data type more suitable for sparsely populated data (for example, run-length encoded lists). For all other tests, where the NDT cells were not pathologically small, memory allocation was not a problem and the reported time and wall clock time were the same.

Initial error: The sensitivity of the algorithms with respect to the amount of error in the initial pose estimate was also tested, both for the translational and rotational components. The results are shown in Figures 16 and 17. For the translation error tests, the initial rotation error was set to zero, and the translation error was set to zero when testing the sensitivity to the initial rotation error.

Again, 3D-NDT shows a smaller median error in most cases, although failed registrations start to occur at smaller values for the initial error than is the case for ICP. The reason that the median error is smaller for 3D-NDT is probably because the PDFs are a better surface description than point clouds, which have no information about the surface be-

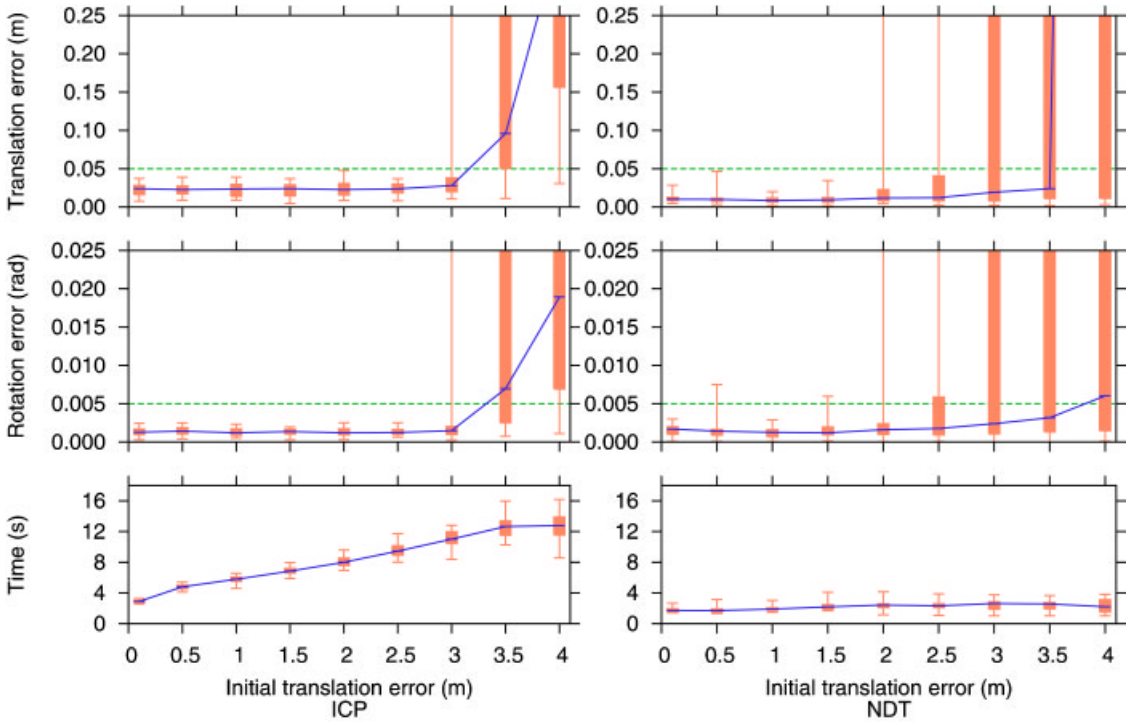

Figure 16. Comparing the sensitivity to the initial error in the translation estimate for the JUNCTION set. The initial rotation error was 0 for these tests.

tween points. Without infinite outer bounds, fewer points from the data scan are used when the initial pose error is large. With the baseline settings for ICP, an initial translation error of up to 2.5 m (when the error in rotation is zero) or a rotation error of up to 0.35 rad (when the translation error is zero) can be handled reliably for the JUNCTION data set. Using 3D-NDT, failed registrations start to occur at 2 m translation error or 0.3 rad rotation error. The results for the TUNNEL set show the same tendencies.

The time taken by ICP increases with the magnitude of the initial pose error, while 3D-NDT takes about the same amount of time for all of the runs.

Discretization methods: Test results for 3D-NDT with different discretization methods on the JUNCTION and TUNNEL data sets are shown in Figure 18. For these tests, cell sizes varying between 2 and 1 m were used. The results for fixed 2 m cells are shown for comparison. The fixed cell plots are labelled F (without infinite bounds) and FI (with infinite bounds). With the initial error set according to the baseline setup, all methods performed equally well on the JUNCTION set. To show the differences in the methods' efficiency, the initial pose error was increased a little for the tests on the JUNCTION set so that e_r=0.2. The different discretization methods are described in Section 4.3.

- Octree subdivision (O, OI) did not lead to a noticeable improvement for the JUNCTION data set. A probable reason for this is that the added detail was not needed for this data set, as it has clear and large features. Octree subdivision did improve the result for the TUNNEL data set, approximately halving the median error compared to using fixed cells.
- Additive octree subdivision (A, AI)—computing the score for each point by summing all leaves in the octree where it belongs instead of using a single leaf—improved the result of the TUNNEL set slightly, at the cost of a minor increase in execution time, because more cells were investigated for each point. However, for an unknown reason, 3D-NDT

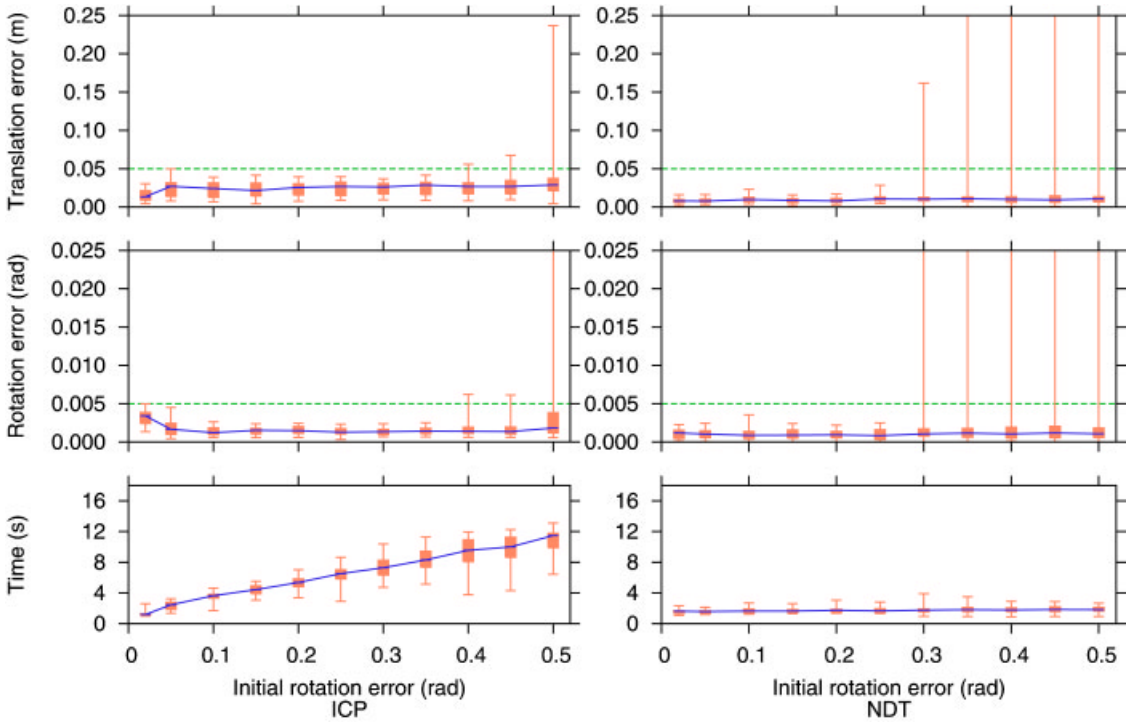

Figure 17. Comparing the sensitivity to the initial error in the rotation estimate for the JUNCTION set. The initial translation error was 0 for these tests.

with additive octree subdivision failed for two of the initial pose estimates when running on the JUNCTION data set. The results for the other 98 poses were still satisfactory.

- Iterative subdivision with varying cell size (I, II)—the more "brute-force" method—removed all of the failed registrations for the JUNCTION data set, at the cost of longer execution times. Iterative subdivision and additive subdivision with infinite outer bounds were the only methods that succeeded in accurately registering the TUNNEL data set from at least 75% of the inital poses. For the tests shown here, the first iteration used 2 m cells. For each subsequent iteration, the cell size was multiplied by 0.75, and the registration was stopped when the size was smaller than 1 m. In other words, the cell sizes used were 2, 1.5, and 1.125 m, respectively.
- Using linked cells led to a slight improvement for the TUNNEL data set, especially for the rotation component of the pose. Interestingly, it did not lead to an improvement for the JUNCTION data set. The likely reasons for this are that, firstly, the error in the initial pose estimate was not large enough for the outer cells to have any significant effect, and, secondly, that the scans overlap completely.

Based on these results, the best performance was obtained using iterative subdivision with infinite outer bounds, at a slightly higher computational cost than the noniterative variants of 3D-NDT, though it was still faster than ICP.

5.2.2. Results with Mobile Robot Data

The KVARNTORP-LOOP data set contains scans collected by a mobile robot, together with pose estimates for each scan, derived from the robot's two-dimensional odometry. This is more like the actual situation that can be expected in the mine mapping application than the artificial (but more complete) experimental setup used for the other two data sets.

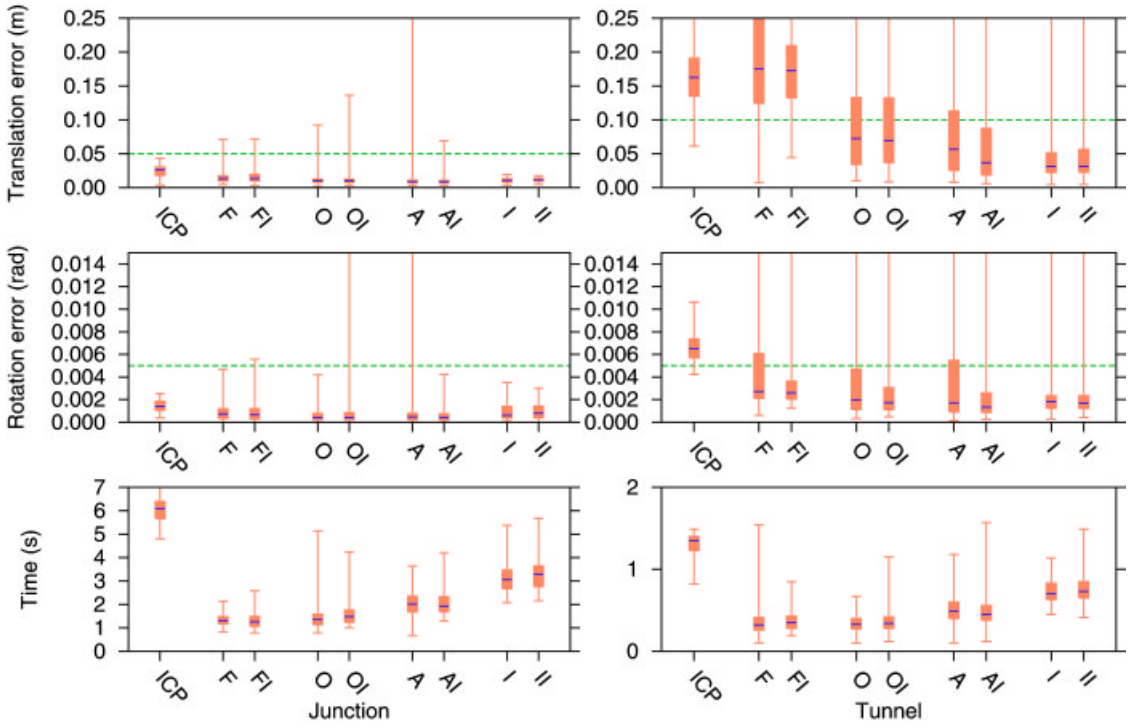

Figure 18. Comparing different discretization methods for 3D-NDT on the JUNCTION and TUNNEL data sets. For the JUNCTION tests, e_t=1 m and e_r=0.2 rad. For TUNNEL, e_t=1 m and e_r=0.1 rad. Baseline ICP is on the left. The next two plots (F and FI) show 3D-NDT with fixed cells, O and OI show octree subdivision, A and AI show additive subdivision, and I and II show iterative subdivision, The rightmost plot in each NDT plot pair (◦I) uses infinite outer bounds but not linked cells (not applicable for ICP).

The more artificial setup can be considered more complete because, for those experiments, the algorithms were tested from a larger set of possible starting poses, and the properties of the algorithms were investigated more thoroughly.

For the results presented here, 8000 random sample points (around 8%) from the data scan and all points from the model were used. Infinite outer bounds were used for 3D-NDT, but not linked cells. The following text covers the effects of using different cell sizes and discretization methods.

Because of some practical problems during the data collection in the Kvarntorp mine, the odometry had to be reset at three points (after scans number 11, 16, and 66). These results are for the longest consecutive scan sequence (scans 17–66).

The pose error from odometry was up to around 1.5 m and 0.2 rad from one scan to the next. Given that the size of each scan is around 10 by 30 m, a rotation error of 0.2 rad is quite large. An example of how bad the odometry can be when driving on gravel with a small mobile robot is shown in Figure 19. Scan 49 is severely rotated with respect to the previous scan, which was taken just 5 m earlier. Measuring the turn angle from odometry is always problematic, and especially so when driving over a surface with loose rocks.

The results are presented as histograms in Figures 20–23. Two limits were chosen for each component of the error of the pose estimate after registration. Because of the difficulty of finding a real ground truth pose, all registrations that came within a certain limit of the manually determined true pose were considered successful. The ground truth poses were determined by running and inspecting a number of registration attempts, and an average of the best matches was used as the ground truth pose for each scan pair. A second limit was also picked. Registrations that came inside this limit are not exact matches, but "acceptably" close for the application.

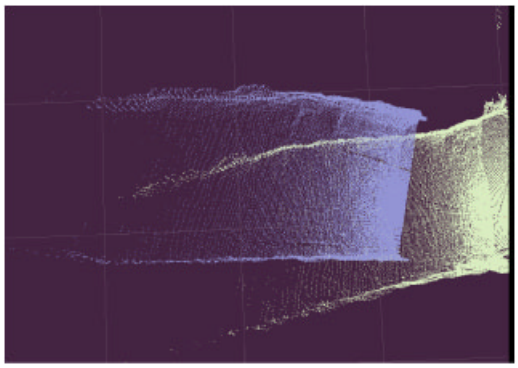

Figure 19. Scans 48 (light, yellow) and 49 (dark, blue) before registration, seen from above.

The limits for this data set were chosen to be 0.10 m and 0.005 rad for "good" matches, and 0.20 m and 0.010 rad for "acceptable" matches. Registrations where any of the pose components are outside of this limit were regarded as failures. The most important feature of the plots to judge the quality of each registration algorithm is the height of the leftmost histogram box, showing the number of successful registrations. The histogram boxes that only have one entry are labelled with the corresponding scan number, to make it clearer which scans fail to be registered. Also included in the plots are box plots showing the distribution of the results.

The results from using 3D-NDT with fixed cells with different sizes are shown in Figure 20. When the cells are too small (0.5 m), scans where the odometry pose is too far from the actual pose fail. When the cells are too large, features that are needed for accurate registration are smoothed out, also making registration fail in more cases. Looking at Figure 20, a cell size of around 2 m seems to be the preferable choice for this data set.

The orientation was generally easier to get right than the position, because the large-scale features of the tunnel scans were sufficient to get the correct rotation angle.

Figure 21 shows the results of different adaptive subdivision methods; starting with 2 m cells, and using cells with 1 and 0.5 m side length as needed. Octree subdivision improves the registration of a number of the scans, compared to using fixed 2 m cells, resulting in 40 successful registrations. Using additive subdivision instead of standard octree subdivision did not lead to an additional improvement for the KVARNTORP-LOOP data set. Iterative subdivision, however, registered 45 of the 50 scan pairs with very high accuracy, and only failed with two scans—the difficult scans number 49 and 41. It is interesting to note that only the rotation component of scan 41's pose and only the translation component of scan 49's pose were wrong. The time needed for 3D-NDT with iterative subdivision was longer than for the other subdivision methods, because two extra runs of the algorithm were performed for each scan. However, the increase is not linearly proportional to the number of iterations. Iterative 3D-NDT took about twice as long as a single iteration of the other versions of the algorithm, even though three passes were performed for each scan. The reason for this is that in most cases, the scans are already in registration at the last pass, so that the last iteration is very fast.

Figure 22 shows the results of registering the same data set with iterative 3D-NDT, with and without infinite outer bounds. Using linked cells did not give an improvement for these scans. When not using infinite outer bounds, the translation component of scan 41 and the rotation component of scan 48's final pose estimate were worse than when using infinite bounds. This shows that using infinite bounds for the outer cells helps in some cases. Apart from that, the results were very similar to when using infinite bounds.

The KVARNTORP-LOOP data set was also registered with ICP. For this experiment, a decreasing distance threshold was used, starting at 2 m and decreasing to zero, instead of the fixed 1 m threshold from the baseline setup. The results are shown in Figure 23. The number of successful registrations was comparable to that of 3D-NDT, though ICP had a few more failures. The main difference lies in the running time of the two algorithms. 3D-NDT was typically almost three times faster than ICP when using the same sampling ratio.

6. SUMMARY AND CONCLUSIONS

A new method for registration of 3D range scans, 3D-NDT, has been presented. A detailed analysis of the algorithm with respect to different methods and parameters based on real-world experiments in a mine

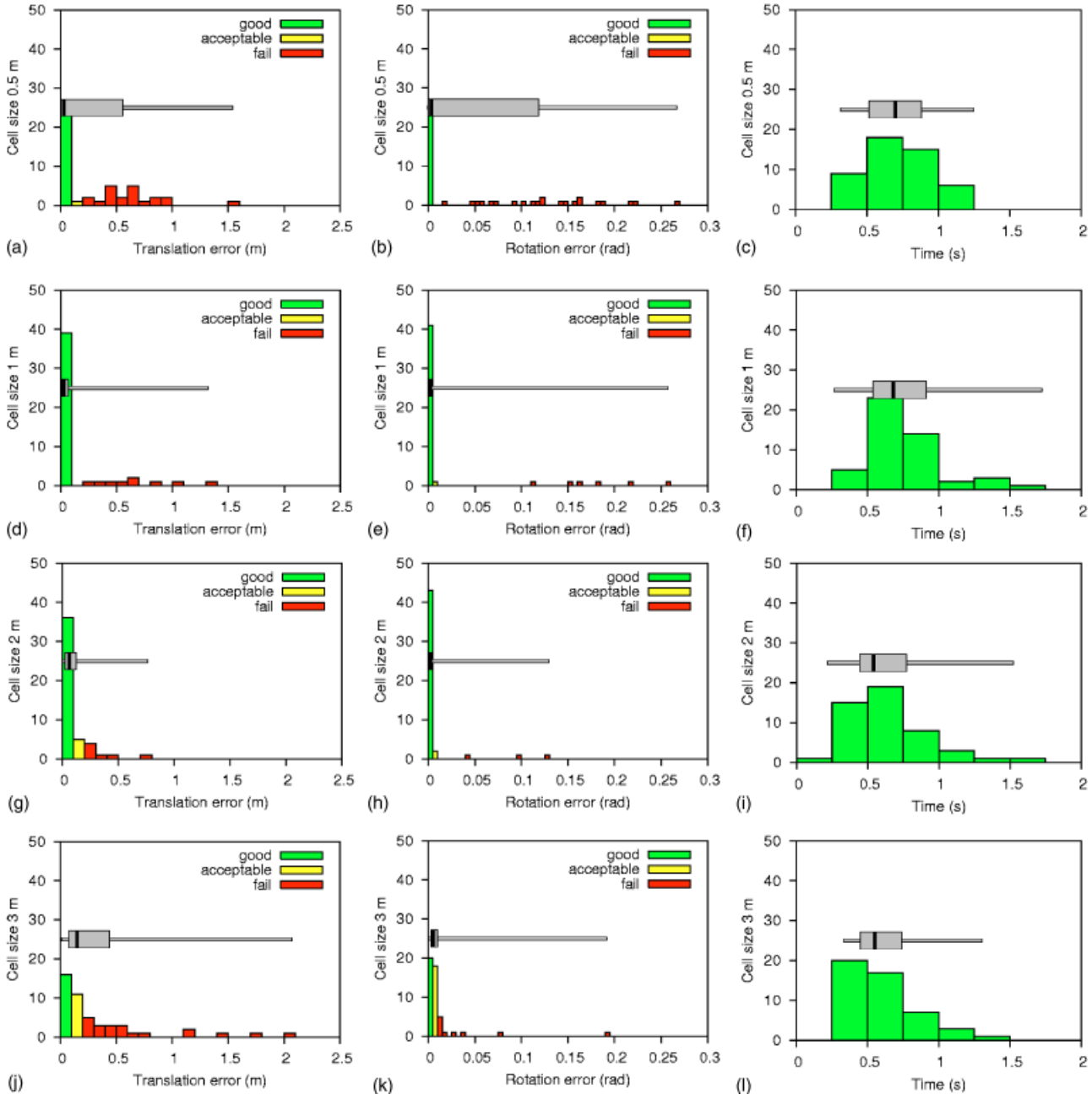

Figure 20. NDT with fixed cells, ranging from 0.5 m (top) to 3 m (bottom). In order to make the plots easier to read, the scan labels are not shown in these plots.

has also been presented, along with a comparison to ICP, the most common registration algorithm used today. The main reason why 3D-NDT is faster is because it avoids the computationally challenging nearest-neighbor search, which is central to the ICP algorithm. Using iterative subdivision for building NDT's model surface description overcomes the problems associated with discretizing the scan volume into fixed grid cells. It has been shown that 3D-NDT with iterative subdivision and infinite outer

应该是上面的一个变种

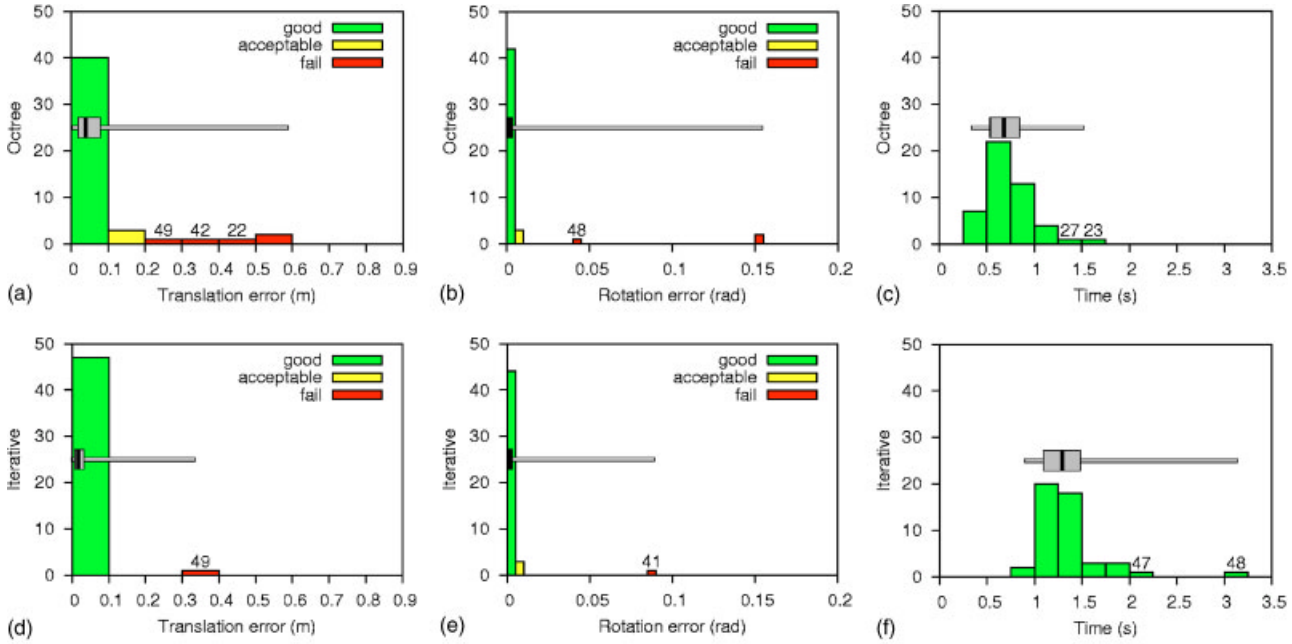

Figure 21. 3D-NDT with different discretization methods. Octree split top, iterative split at bottom. Cells with sizes 2; 1; and 0.5 m were used. Iterative subdivision is clearly the best choice here, as it has only two failed registrations (the position of scan 49 and the orientation of scan 41) and three "acceptable" matches. The time taken is about twice that of octree subdivision. Additive octrees and standard octrees had very similar performance for this data set.

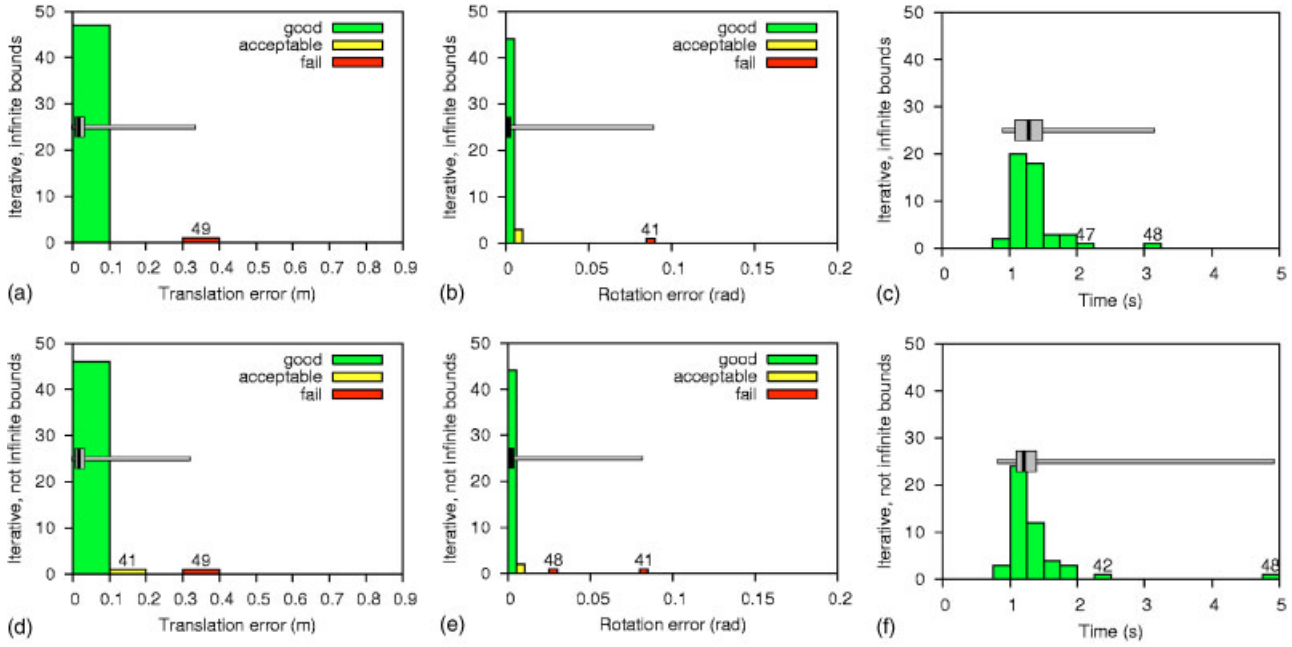

Figure 22. NDT with (top row) and without (bottom row) infinite outer bounds. Using linked cells did not give a noticeable improvement for this data set, but increased the time substantially.

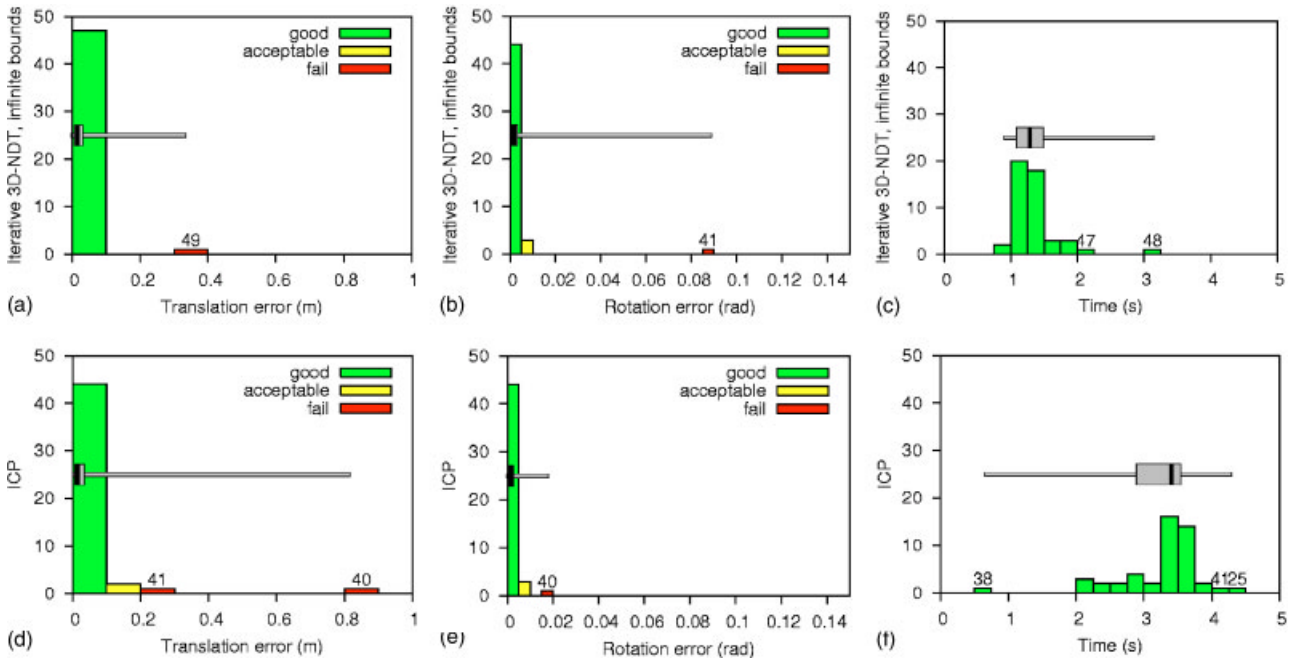

Figure 23. Results of using ICP on the KVARNTORP-LOOP data set, using a decreasing distance threshold of 2 m to 0 m. The results using iterative 3D-NDT are shown above for comparison.

bounds consistently leads to accurate registration of difficult scan data, requiring less time than ICP.

With the experimental setup used in the work presented here, the alignment speed is not critical. Since the vehicle is stopped and moved between scans, the few seconds saved from using a faster registration algorithm are not very important. In the real application, however, the plan is to collect 3D scan data while the vehicle is moving, possibly using a 3D lidar camera that collects a full-frame range image at vide frame rates. Also, the computer hardware on the intended platform (Atlas Copco's drill rigs) is slower than the system used for these tests. In such a scenario, the need for fast scan alignment will be much higher.

One of the other main advantages of 3D-NDT is that the scanned surface can be stored much more efficiently using the combined normal distributions than if the point clouds themselves, or even sparse subsamples of the point clouds, are stored. This is important for any large 3D map. In a scenario where a dynamic map is maintained over a long time, the storage requirements for 3D point cloud data would also soon grow uncomfortably large. Storing the NDT representations of the scans requires only a small fraction of the space required by ICP, and the NDT representation is still powerful enough for registering new scans to the collected data, as has been shown in this paper.

两个优点：

1. 计算速度快，不需要像ICP那样进行点对点的搜索和计算；

2. 使用的存储资源少，不需要存储整个点云数据

ACKNOWLEDGMENTS

Thanks to Atlas Copco Rock Drills AB for financial support and access to the mine and equipment, and to Optab Optronikinnovation AB for providing the laser scanner used for some of the experiments.

REFERENCES

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 239–256.

Biber, P., & Strasser, W. (2003). The normal distributions transform: A new approach to laser scan matching. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Vol. 3, pp. 2743–2748.

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. Image and Vision Computing, 10(3), 145–155.

Gelfand, N., Ikemoto, L., Rusinkiewicz, S., & Levoy, M. (2003). Geometrically stable sampling for the ICP algorithm. In Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling, pp. 260–267, Banff, October 2003.

Greenspan, M., & Yurick, M. (2003). Approximate k-d tree search for efficient ICP. Paper presented at Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM '03).

Mitra, N. J., Gelfand, N., Pottmann, H., & Guibas, L. (2004). Registration of point cloud data from a geometric optimization perspective. In Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 23–31, ACM Press, New York, NY.

Ripperda, N., & Brenner, C. (2005). Marker-free registration of terrestrial laser scans using the normal distribution transform. In Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH 2005, Mestre-Venice, Italy, 22-24 August, 2005.

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

Takeuchi, E., & Tsubouchi, T. (2006). A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), October 2006, pp. 3068–3073, Beijing, China.

Wulf, O., & Wagner, B. (2003). Fast 3D-scanning methods for laser measurement systems. Paper presented at the International Conference on Control Systems and Computer Science (CSCS14).