

MARKER-FREE REGISTRATION OF TERRESTRIAL LASER SCANS USING THE NORMAL DISTRIBUTION TRANSFORM

Nora Ripperda and Claus Brenner

Institute of Cartography and Geoinformatics, University of Hannover, Germany
{Nora.Ripperda, Claus.Brenner}@ikg.uni-hannover.de

Working Group V/4

KEY WORDS: LIDAR, Laser scanning, Terrestrial, Registration, Matching

ABSTRACT

The registration of scan data often uses special markers which are placed in the scene. This leads to a reliable registration but the method is not very efficient. Therefore, we search for a registration method which works without markers. There are methods like the iterative closest point (ICP) algorithm which calculate the registration on the basis of the data itself. However, these algorithms have a small convergence radius and therefore a manual pre-alignment is necessary. In this paper, we explore a registration method called the normal distribution transform (NDT) which does not require markers, has a larger convergence radius than ICP and a medium alignment accuracy. The NDT was initially proposed in robotics for single-plane horizontal scans. We investigate three modifications to the original algorithm: a coarse-to-fine strategy, multiple slices, and iterative solution using the method of Levenberg-Marquardt. We apply the modified algorithm to real terrestrial laser scanner data and discuss the results.

1 INTRODUCTION

Nowadays, the standard technique to align terrestrial scans uses specialized markers, e.g. retro-reflective targets or spheres, which have to be placed in a scene prior to scanning. As there are typically only a few such targets in the scene, they have to be placed quite deliberately. Automatic extraction operators usually followed by some manual interaction are used to locate the markers in the scene, which are then scanned again in high resolution in order to obtain a high redundancy and thus, a high accuracy.

This technique works reliably, but it is not very efficient. According to our experience, 10–15 terrestrial scans with different standpoints are feasible per day. This means that scanning time and total time differ by a factor of about eight. The most time-consuming parts are (i) the distribution and collection of markers in the scene, (ii) checking and selection of automatically extracted marker positions in the scan image, and (iii) fine scan of the markers. Thus, it is desirable to replace the marker approach by another technique, which, however, must be able to align scans rapidly, so that the result is immediately available during the measurement campaign. Such an attempt would also be reasonable in a setting where the laser scanner is mounted on a vehicle.

There are well-known methods for aligning scans based on the scan data itself, most notably the iterative closest point (ICP) algorithm (Besl and McKay, 1992) and its variants. In fact, those methods are nowadays available in commercial software packages. However, usually the convergence radius is not very high, so that scans have to be pre-aligned interactively before starting automatic alignment. What we seek for is a large convergence radius and medium align-

ment accuracy, as we can still improve accuracy by post-iterating using the ICP algorithm once an initial alignment is found.

In this paper, we explore a method to align terrestrial scans without artificial markers. It is borrowed from robotics, where the task is to simultaneously navigate in an unknown environment while at the same time building a map of it (so-called simultaneous localization and mapping, SLAM). In order to determine the transformation between successive scans, the normal distribution transform described in (Biber, 2003) is used.

In the context of robotics, the NDT is used to align 2D scan data. Especially in indoor environments, scanning in one plane parallel to the ground leads to sufficiently characteristic scan data. However, in an outdoor setting, a certain fixed plane will often not contain enough information to align scans. Thus, we explore a method to obtain subsets of our original full 3D scan data which contain enough information to align properly.

The paper presents the original NDT and the modifications we have implemented in order to use it with 3D data. The methods are tested using real data acquired with our terrestrial scanner. Finally, the results are presented and discussed.

2 RELATED WORK

A well-known registration method is the iterative closest point algorithm (Besl and McKay, 1992, Chen and Medioni, 1992). ICP is based on a nearest point to point or nearest point to tangent plane search. For each point, the nearest object is calculated and from all the pairs the rigid

transformation which aligns both scans is estimated. This transformation is applied to one of the scans. It is iterated until a convergence criterion is reached. The convergence behaviour of ICP is monotone with respect to the mean square distance and very slow. In the average case there are about 30 to 50 iterations.

To accelerate convergence of the ICP, (Besl and McKay, 1992) propose to update the parameter vector by a linear or parabolic extrapolation. (Pottman et al., 2004) force the parameter vector to a helical movement. This leads to an quadratic convergence but there is the possibility of exceeding the solution or to impair the orthogonality of the rotation matrix.

(Fitzgibbon, 2001) use the Levenberg-Marquardt algorithm to minimize the error. This is a combination of Gauss-Newton and gradient descent and presents a closed update of the transformation. This enables the use of the Huber- or Lorentzian-kernel as error function which enlarge the convergence radius.

The most expensive part of the computation is the nearest point search. (Zhang, 1992) accelerate the the nearest point search by a coarse-to-fine strategy. The first iterations only use a subset of the points whereas for the fine registration all points are used.

Another way to reduce computation time is to use special search structures. (Besl and McKay, 1992) suggested a k-d tree search and (Greenspan and Yurick, 2003) developed an approximate k-d tree search. This method returns a point which is not necessarily the nearest neighbour but it guarantees that the distance of the solution and the original point is not bigger than $1 + \epsilon$ times the distance between the nearest neighbour and the original point. Different combinations of k-d tree search and approximate k-d tree search are used to get the benefits of shorter computation time of the approximate k-d tree in the first iterations and the full accuracy in later iterations.

(Greenspan and Godin, 2001) use a preprocessing step to compute correspondences. For each reference point, all points within a certain distance are calculated. These points are stored in increasing distance from the reference point. At runtime, the relations are kept up to date and the correspondence of the previous iteration is used as an estimate for the current correspondence.

3 TERRESTRIAL SCAN DATA

Large progress has been made in the area of terrestrial laser scanners during the last decade. A number of scanners is nowadays available, with measurement ranges from 25 to 1000 meters, accuracies from as few as 6 millimeters to centimeters, and scanning rates of more than 100,000 points per second. Major applications are the scanning of industrial facilities for as-built documentation, acquisition and monitoring of engineering structures, buildings and facades, and cultural heritage. Although it is usually the case that dense terrestrial data is acquired for selected objects

only, using a fixed number of viewpoints, it is worth to note that there are commercial systems which allow to acquire dense, large scale 3D point clouds from terrestrial laser scanning using moving platforms.

The test data sets we have used for the investigations in this paper were acquired using the scanner Riegl LMS-Z360i. A number of scans were taken along a street in Hannover, Germany, with buildings of different height. Color information has been acquired as well, however it has not been used for the registration. Each scan requires about 4 minutes and yields approximately 2,250,000 scanned points. We have chosen the street scene deliberately because, as noted earlier, scanning from a mobile platform would be one application that could benefit from a marker-free registration procedure. Figure 1 shows several scans of the street scene where each scan has a different color.

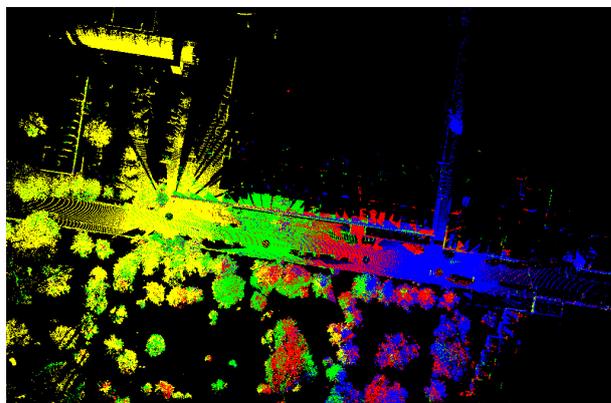


Figure 1: Four scans taken along a street in Hannover, shown in different colors.

4 SCAN ALIGNMENT USING NDT

This section describes the normal distribution transform (Biber, 2003) and how it is used for the registration of two scans. As in the original publication, the transform is first described in two dimensions, although it is straightforward to generalize it to higher dimensions.

The NDT converts the original point cloud of the first scan into another representation, which captures the distribution of the points, rather than individual points. In order to do this, the area covered by the first scan is subdivided into a regular grid of cells. It is assumed that the distribution of points inside each cell is characterized to a sufficient extent by a normal distribution. Thus, for each cell the mean q and the covariance matrix Σ are calculated, using

$$q = \frac{1}{n} \sum_i x_i, \quad \Sigma = \frac{1}{n-1} \sum_i (x_i - q)(x_i - q)^t.$$

The probability of measuring a point x in cell i is then modeled by the normal distribution $N(q_i, \Sigma_i)$,

$$p(x) = C \cdot \exp\left(-\frac{1}{2}(x - q_i)^t \Sigma_i^{-1} (x - q_i)\right).$$

For the purposes of the NDT, C is set to 1. A set of points $x_j, 1 \leq j \leq J$ is assigned the score

$$\text{score} = \sum_{j=1}^J \exp\left(-\frac{1}{2}(x_j - q_{i_j})^t \Sigma_{i_j}^{-1} (x_j - q_{i_j})\right) \quad (1)$$

where i_j is the index of the cell the point i falls into.

As an example, figure 2 shows original scan points in a) and the corresponding NDTs with cell size b) $30\text{m} \times 30\text{m}$, c) $10\text{m} \times 10\text{m}$ and d) $5\text{m} \times 5\text{m}$. White pixels represent a high probability and black ones a low probability.

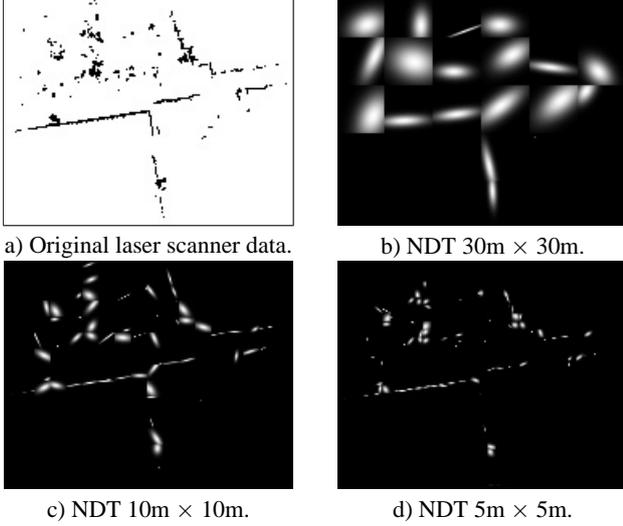


Figure 2: Laser scanner data and associated NDTs for different cell sizes.

Figure 3 illustrates how the score function varies when a point set is rotated. The original scan from figure 2 a) is converted into the NDT using a cell size of $10\text{m} \times 10\text{m}$. The second set is chosen identical to the original scan, except for a rotation which is carried out from $-\pi$ to π in steps of $\frac{\pi}{50}$. For each rotation angle, the score is evaluated. The plot shows the expected peak at zero.

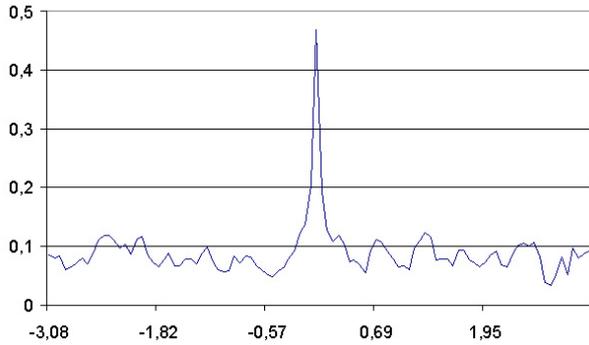


Figure 3: Behaviour of the score function when the point set is rotated.

Figure 4 shows the behaviour of the score function when the scan is translated. The translation is performed from

-100 to 100 in x direction and from -30 to 40 in y direction. This corresponds to translations where the two scans overlap by at least 50 percent in each direction.

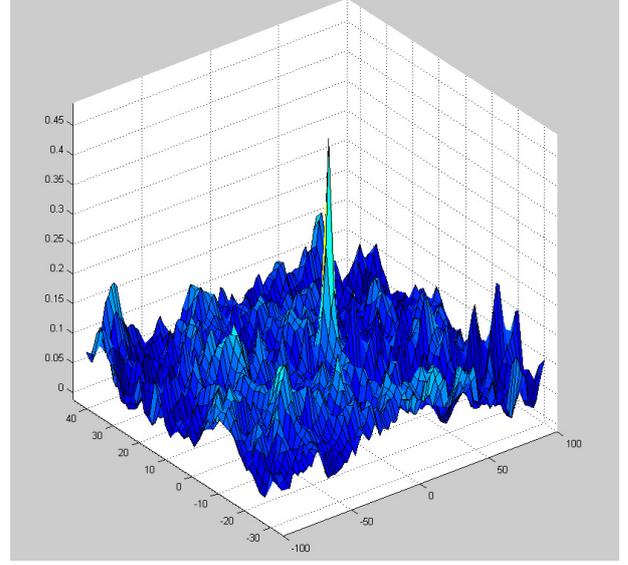


Figure 4: Behaviour of the score function when the point set is translated.

In order to register two scans, the first one is held fixed and its NDT is computed. The second one is transformed using the Euclidean transformation in the plane,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (2)$$

The three parameters of this transformation are $p = (t_x, t_y, \phi)$. They are to be estimated so as to maximize the score function (1), which is now dependent on the transformation parameters p . Since the optimization algorithms minimize functions, we treat the negation of the score function $f = -\text{score}(p)$. There are no start values for p assumed, so all parameters are initialized to zero. The following steps are iterated until a convergence criterion is reached.

1. Transform each point x_j of the second scan according to (2) using the current transformation parameters p .
2. Compute the cell index i_j in which each point x'_j lies and retrieve the parameters q_{i_j}, Σ_{i_j} of the normal distribution of that cell.
3. Calculate the score value, $\text{score}(p)$, according to (1).
4. Compute new parameters by optimizing the function $f = -\text{score}(p)$ using Newton's Algorithm.

$$H \Delta p = -g,$$

where H is the Hessian and g the transposed gradient of f .

Figure 5 shows several snapshots of the iterative alignment of two datasets.

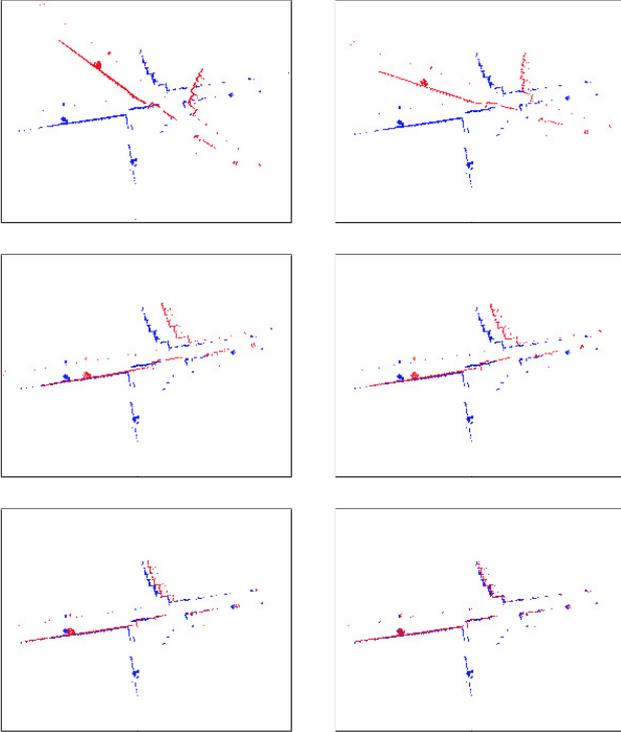


Figure 5: Snapshots of the iterative registration process for two scans.

5 MODIFICATIONS TO THE ORIGINAL NDT ALGORITHM

In this section we describe the different modifications we made on the NDT. Namely there are a coarse-to-fine strategy, the use of slices as subsets of the data and the choice of another optimization technique.

5.1 Addition of a coarse-to-fine strategy

The parameter estimation of the NDT is iterative. Just as with the ICP, bad initial values may lead to local minima. In fact, since the score function (1) relies on points hitting cells of the NDT, no estimation will take place at all if the second data set is not overlapping the cells of the NDT of the first data set. Also, we experienced problems when the two data sets to be aligned are mainly linear in structure and initially cross in a point, since in this case, the ability to turn the second data set depends entirely on the NDT cell where both data sets cross. If the normal distribution of this cell has a main axis far off the correct direction, the second data set will not be tied versus the first one. Especially if the cells are small, their normal distributions may not be meaningful.

In order to address this problem, the original algorithm of (Biber, 2003) was augmented by a coarse-to-fine strategy. The cell size of the NDT is chosen large when iteration starts. This helps to align the data sets according to their coarse structure. Then, the cell size is successively decreased, making it possible to take the fine structure into account.

In the experiments, a cell size of $200\text{m} \times 200\text{m}$ was found to be appropriate to initially align the data sets. Successive sizes of 200m, 100m, 50m, and 25m were used. This sequence may be too elaborate for some data sets, however we found that it leads to good results in the case of more ‘difficult’ data sets as well. The usage of another optimization method is leading to a smaller initial cell size of 100m (see sec. 5.3). For any cell size, the algorithm iterates until the score function does not improve for a fixed number of iterations. Then, it continues with the next smaller cell size.

5.2 Slicing 3D scans

The discussion so far has addressed 2D scan data, which is especially useful in robotics applications. Instead, in our case, full 3D scans have to be aligned for the purpose of extracting 3D scene descriptions. It is straightforward to extend the NDT to three dimensions: the grid cells become voxels which contain 3D normal distributions. However, this will lead to a large number of voxels, possibly negating the advantages of the NDT.

Considering the typical application, though, it is usually reasonable to assume that the laser scanner is set up approximately upright for each scan. Also, the approximate distance to the ground will be always the same or, if it varies, it will be technically simple to measure it. In order to apply the 2D NDT directly, a slice parallel to the ground can be cut out of the 3D data (see figure 6).

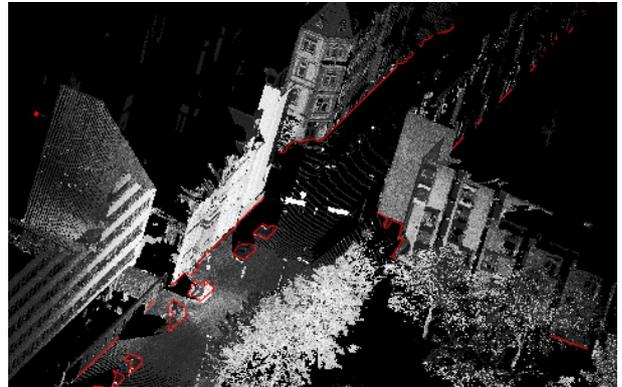


Figure 6: Cutting a 2D slice out of the 3D scan.

A single slice may, however, not contain enough data for an alignment. As compared to the case in robotics, where highly structured indoor environments prevail, natural scenes may lack the desired amount of information. For example, when driving along a corridor of buildings, a single slice may yield essentially points along two parallel lines (the projection of façades to the left and right of the street). Thus, transformation (2) will be under-determined, since a translation along the street axis does not change the score.

We also made the experience that taking a single slice increases the chance that different real-world structures that cause similar patterns in the scan data are erroneously assigned to each other. For example, figure 7 shows a case

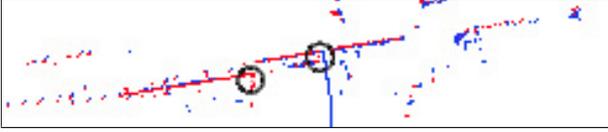


Figure 7: Erroneous registration of two data sets.

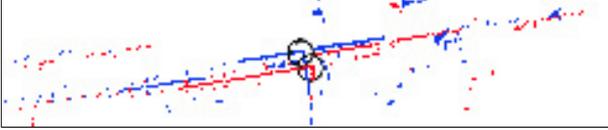


Figure 8: Correct registration of Fig. 7.

where a corner is present in each scanned data set. The single slice NDT associates both corners, although they belong to different real-world objects.

Single slices also lead more often to the situation where coarse alignment, using large cell sizes, yield good results, which however are later getting worse when cell sizes are reduced. For example, figure 9 shows the result of a good coarse alignment, and figure 10 shows how this gets much worse at a later stage with smaller cell sizes.

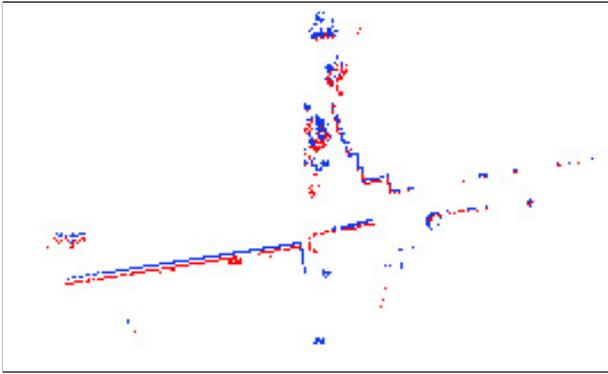


Figure 9: Coarse registration using a large cell size.

To solve these problems we took a set of n slices instead of a single slice. Each slice has a height of 40 cm, all points within a slice are projected to the ground plane. With our data sets, the lower slices typically contain 15,000 to 20,000 points. Since there is only one high building in the scene, we have appreciably smaller data sets at a height of 12 meters or higher, containing 1,000 to 5,000 points.

The slices lead to n independent NDTs, calculated with the same cell size. Since only one single transformation is sought for, the score function sums up the probabilities over all points and height levels

$$\text{score}(p) = \sum_{k=1}^n \text{score}_k(p), \quad (3)$$

where $\text{score}_k(p)$ is the score function (1), defined for slice k and parameters p . Equation (3) is optimized in the same way as the single score solution (1).

Figure 11 shows how the score function changes with an increasing number of slices, for the example of rotating one data set shown earlier in figure 3. As can be seen,

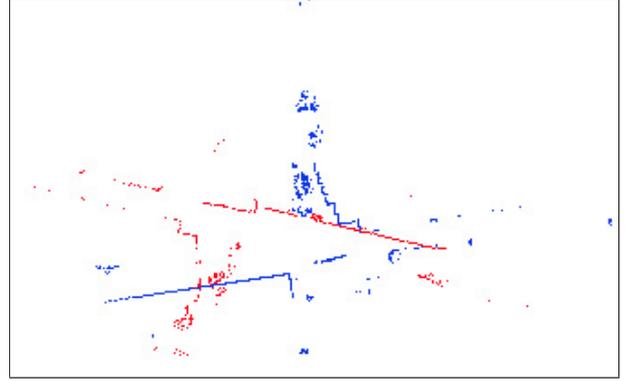


Figure 10: When iteration of Fig. 9 is continued with a smaller cell size, alignment is lost.

the score function becomes more and more smooth outside the peak at zero degrees. Figure 12 shows the same for translating one data set like in figure 4.

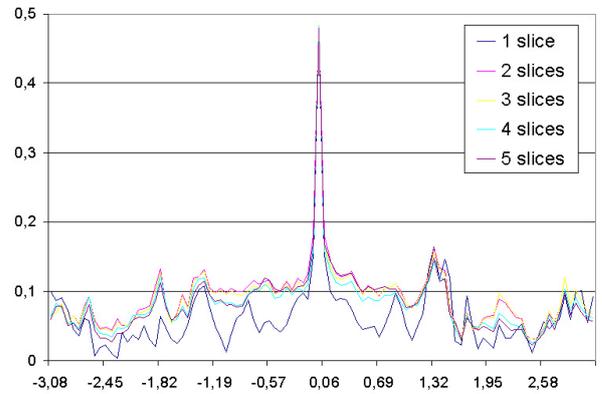


Figure 11: Behaviour of the score function from fig. 3 when different numbers of slices are used.

Figure 13 shows the result of using a two-slice NDT. Individually, the two slices, shown in figs. 7 and 9, cause the two problems discussed earlier. Combined in a single NDT, however, a correct registration is obtained.

5.3 Optimization using Levenberg-Marquardt

In (Fitzgibbon, 2001) the optimization uses the method of Levenberg-Marquardt (Press et al., 1992). We applied this method to the NDT algorithm as well. This section describes the method and the results obtained.

The Levenberg-Marquardt method combines Gauss-Newton optimization with gradient descent. To remain in our notation, we define the score s_j of a single point x_j ,

$$s_j = \exp \left(-\frac{1}{2} (x'_j - q_{i_j})^T \Sigma_{i_j}^{-1} (x'_j - q_{i_j}) \right).$$

$s(p) = (s_j(p))_{j=1, \dots, N_d}$ is the vector of all score functions where N_d is the number of points. Let J be the Jacobian $N_d \times 3$ -matrix with the entries $J_{i,j} = \frac{\partial s_i}{\partial p_j}$.

Gauss-Newton is similar to Newton optimization but the

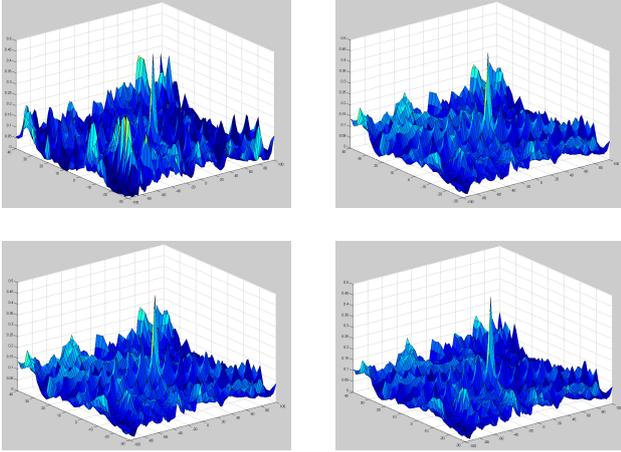


Figure 12: Behaviour of the score function from fig. 4 when 1 to 4 slices are used.

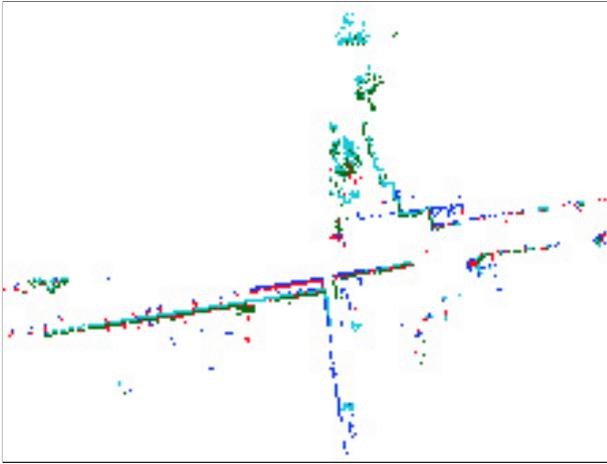


Figure 13: Combining two slices leads to a correct registration.

Hessian H is approximated by $J^T J$ where J is the Jacobian matrix. The parameter update is calculated as

$$\Delta p = -(J^T J)^{-1} J^T s.$$

If the underlying error function is close to being quadratic in the parameters p , this will converge fast to the minimum value. On the other hand, since J is a linearization, the update is not guaranteed to reduce the nonlinear error function.

In an accelerated gradient descent approach the parameters are updated using

$$\Delta p = -\lambda^{-1} J^T s,$$

where λ controls the magnitude of the steps in gradient direction. A small λ leads to large steps and vice versa. If λ is large enough, this approach guarantees to decrease the error term but convergence may be very slow.

To achieve good results in any situation the Levenberg-Marquardt algorithm combines the two procedures. The parameters are updated using

$$\Delta p = -(J^T J + \lambda I)^{-1} J^T s. \quad (4)$$

λ is set to a start value and updated in each iteration. The value of Δp is determined. If the parameter update reduces the error, Δp is accepted and λ is divided by a factor. In the other case λ is multiplied by the same factor and Δp is calculated again until it satisfies the condition for the first case. A typical start value for λ is 10^{-3} times the average of the diagonal elements of $N = J^T J$. The factor for division and multiplication is typically 10.

We applied this optimization algorithm with a small modification to our multi slice NDT. The modification is necessary because the score function involves only those points which fall into cells with at least three points. This possibly leads to a reduction of the score in some iterations. For this reason it is not possible to prohibit parameter updates which decrease the score function. So we use the Levenberg-Marquardt equation (4) with a fixed λ and accept all parameter updates.

We compared the case of the original Newton optimization with Levenberg-Marquardt. Figure 14 shows the score values of the first iterations with a cell size of $200\text{m} \times 200\text{m}$ for the Newton optimization and $100\text{m} \times 100\text{m}$ for Levenberg-Marquardt. It turns out that Newton optimization generates strong oscillations in the beginning. This is because of the fact that the score function does not take all points into account. This affects only the Newton optimization because Levenberg-Marquardt slows the update down and large parameter changes are suppressed.

Different cell sizes are chosen since the Levenberg-Marquardt method leads to good results with this starting cell size while the Newton optimization oscillates heavier so that no correct alignment can be found. The different cell sizes explain the smaller score value of Levenberg-Marquardt. A smaller cell size leads to less overlapping and several points are not included in the score.

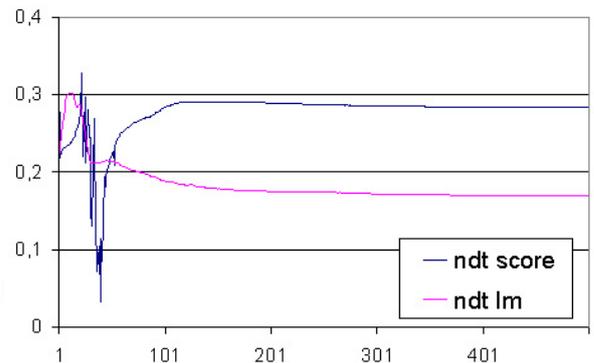


Figure 14: Progression of score values for Newton's method (ndt score) and Levenberg-Marquardt (ndt lm).

As can be seen in Figs. 15 and 16, the parameters t_x , t_y and ϕ converge to the final parameters much faster with Levenberg-Marquardt optimization than with Newton's method. The fastest convergence is reached with small values for λ .

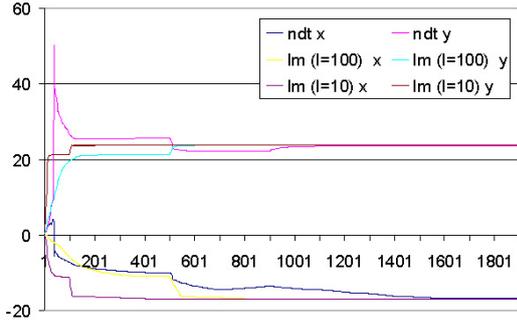


Figure 15: Convergence of t_x and t_y when using Newton's method and Levenberg-Marquardt.

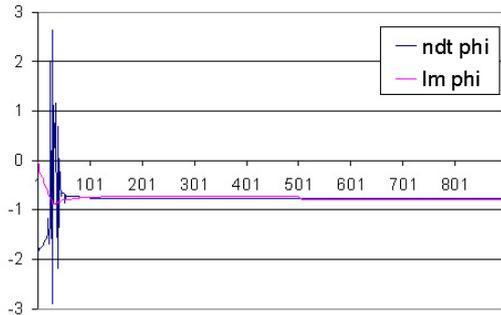


Figure 16: Convergence of ϕ when using Newton's method and Levenberg-Marquardt.

6 CONCLUSIONS

We tested the multi slice NDT with the Levenberg-Marquardt and Newton optimization with three slices. For the coarse-to-fine strategy we used different cell sizes for each method because we had no reliable results with small sizes using Newton's method.

For Levenberg-Marquardt we used cell sizes of 100m, 50m, 25m, 10m, 5m and 3m and iterated 100 times each. To assess our results we compared them to a marker-based registration as provided by the software RiSCAN PRO. Since the markers are well-defined retro-reflective targets which were measured by high density fine scans, this registration was taken as a reference. As a result, we found an average point distance of 8.8mm. The cell sizes for the Newton method were chosen to be 200m, 150m, 100m and 75m. The first size was iterated 500 times and the other 200 times. This led to an average point distance of 0.42m, mainly due to the large cell size.

We made additional convergence tests for the Levenberg-Marquardt method. For this, one of the scans was rotated in multiples of $\frac{\pi}{24}$. For the registration, we used cell sizes of 100m, 50m, 25m, 10m, 5m and 3m. Each size was iterated 100 times. A registration with our method was possible with a rotation angle from -1.83 to 2.09. The tests with starting angle -1.31, -1.047 and -0.785 only converged with an initial cell size of 200m. The difference of the starting angle and the registration angle is shown in figure 17.

We have explored how the normal distribution transform

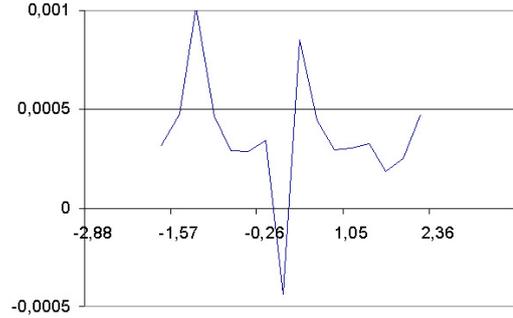


Figure 17: Difference between starting rotation angle and resulting registration angle.

can be used to align 3D scans. We added three enhancements to the original algorithm: a coarse to fine strategy, multiple slices, and iteration using Levenberg-Marquardt. From the datasets we have shown in this paper, we conclude that the algorithm yields a large convergence radius and can be used to provide initial values for a subsequent application of the ICP algorithm.

REFERENCES

- Besl, P. J. and McKay, N. D., 1992. A method for registration of 3-d shapes. In: IEEE, Trans. PAMI, Vol. 14.
- Biber, P., 2003. The normal distributions transform: A new approach to laser scan matching. Technical Report 3, Wilhelm Schickard Institute for Computer Science, Graphical-Interactive Systems (WSI/GRIS), University of Tübingen.
- Chen, Y. and Medioni, G., 1992. Object modelling by registration of multiple range images. *Image Vision Comput.* 10(3), pp. 145–155.
- Fitzgibbon, A., 2001. Robust registration of 2d and 3d point sets. In: British Machine Vision Conference.
- Greenspan, M. and Godin, G., 2001. A nearest neighbour method for efficient icp. In: 3-D Digital Imaging and Modeling, Quebec, pp. 161–168.
- Greenspan, M. and Yurick, M., 2003. Approximate k-d tree search for efficient icp. In: 3-D Digital Imaging and Modeling, Banff, pp. 442–228.
- Pottman, H., Leopoldseeder, S. and Hofer, M., 2004. Registration without icp. In: CVIU, pp. 54–71.
- Press, W., Teukolsky, S., Vetterling, W. and Flannery, B., 1992. *Numerical Recipes in C. 2 edn*, Cambridge University Press.
- Zhang, Z., 1992. Iterative point matching for registration of free-form curves. Technical Report 1658, INRIA.

ACKNOWLEDGEMENT

This work was done within in the scope of the junior research group "Automatic methods for the fusion, reduction and consistent combination of complex, heterogeneous geoinformation", funded by the VolkswagenStiftung, Germany.