Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

The Normal Distributions Transform: A New Approach to Laser Scan Matching

Peter Biber (biber@gris.uni-tuebingen.de)
Wolfgang Straßer
WSI/GRIS
University of Tübingen
Sand 14, 72070 Tübingen, Germany

***Abstract*— Matching 2D range scans is a basic component of many localization and mapping algorithms. Most scan match algorithms require finding correspondences between the used features, i.e. points or lines. We propose an alternative representation for a range scan, the *Normal Distributions Transform*. Similar to an occupancy grid, we subdivide the 2D plane into cells. To each cell, we assign a normal distribution, which locally models the probability of measuring a point. The result of the transform is a piecewise continuous and differentiable probability density, that can be used to match another scan using Newton's algorithm. Thereby, no explicit correspondences have to be established.**

We present the algorithm in detail and show the application to relative position tracking and simultaneous localization and map building (SLAM). First results on real data demonstrate, that the algorithm is capable to map unmodified indoor environments reliable and in real time, even without using odometry data (see video).

I. Introduction

Undoubtedly, simultaneous localization and mapping (SLAM) is a basic capability of a mobile robot system. Laser range scanners are popular sensors to get the needed input, mainly for their high reliability and their low noise in a broad class of situations. Many SLAM algorithms are based on the ability to match two range scans or to match a range scan to a map. Here we present a new approach to the low-level task of scan matching and show, how we use it to build maps.

The main purpose of this paper is the introduction of the *Normal Distributions Transform* (in the following called NDT) and its application to the matching of one scan to another scan or of one scan to several other scans. The NDT transforms the discrete set of 2D points reconstructed from a single scan into a piecewise continuous and differentiable probability density defined on the 2D plane. This probability density consists of a set of normal distributions, that can be easily calculated. Matching a second scan to the NDT is then defined as maximizing the sum, that the aligned points of the second scan score on this density.

We also present a simple algorithm for the SLAM problem, that fits nicely into our matching scheme. However, the proposed matching scheme does not depend on this algorithm. For this reason, we review only methods concerning the matching of two scans in the related work in section II and not methods for building whole maps like the approaches of Thrun [15] or Gutmann [6]. Nevertheless, one component of such approaches is a perceptual model, that is the likelihood of a single scan given a map and a pose estimate. Since our methods yields exactly a measure for such a model, we believe, that our scan matcher could also be integrated into more sophisticated SLAM algorithms.

The rest of this paper is organized as follows: Section III introduces the NDT, section IV gives an overview of the scan matching approach and defines the measure for comparing a scan to a NDT, which is optimized in section V using the Newton algorithm. The scan matcher is applied to position tracking in section VI and to a simple SLAM approach in section VII. We finally show some results on real data and conclude the paper with an outlook on future work.

II. Previous work

The goal of matching two range scans is to find the relative pose between the two positions, at which the scans were taken. The basis of most successful algorithms is the establishment of correspondences between primitives of the two scans. Out of this, an error measure can be derived and minimized. Cox used points as primitives and matched them to lines, which were given in an a priori model [3]. In the Amos Project [7], these lines were extracted from the scans. Gutmann matched lines extracted from a scan to lines in a model [8]. The most general approach, matching points to points, was introduced by Lu and Milios [9]. This is essentially a variant of the ICP (*Iterated Closest Point*) algorithm ([1],[2],[18]) applied to laser scan matching. We share with Lu and Milios our mapping strategy. As in [10], we do not build an explicit map, but use a collection of selected scans with their recovered poses as an implicit map.

hand-crafted features

ICP

NDT

In all of these approaches, explicit correspondences have to be established. Our approach differs in this point, as we never need to establish a correspondence between primitives. There are also other approaches, that avoid

0-7803-7860-1/03/$17.00 © 2003 IEEE

solving the correspondence problem. In [12], Mojaev combines the correlation of local polar occupancy grids with a probabilistic odometry model for pose determination (using laser scanner and sonar). Weiss and Puttkammer [17] used angular histograms to recover the rotation between two poses. Then x- and y-histograms, which were calculated after finding the most common direction were used to recover the translation. This approach can be extended by using a second main direction [7].

Our work was also inspired by computer vision techniques. If the word probability density is replaced by image intensity, our approach shares a similar structure to feature tracking [13] or composing of panoramas [14]. These techniques use the image gradient at each relevant position to estimate the parameters. Here, derivatives of normal distributions are used. Opposed to image gradients, these can be calculated analytically. 概率密度：可微可导

III. The Normal Distributions Transform

This section describes the Normal Distributions Transform (NDT) of a single laser scan. This is meant to be the central contribution of the paper. The use of the NDT for position tracking and SLAM, described in the following sections, is then relatively straightforward.

单纯的分割似乎没有考虑到2D/3D图本身的拓扑结构

The NDT models the distribution of all reconstructed 2D-Points of one laser scan by a collection of local normal distributions. First, the 2D space around the robot is subdivided regularly into cells with constant size. Then for each cell, that contains at least three points, the following is done:

1) Collect all 2D-Points $\mathbf{x}_{i=1..n}$ contained in this box.
2) Calculate the mean $\mathbf{q} = \frac{1}{n}\sum_i \mathbf{x}_i$.
3) Calculate the covariance matrix $\Sigma = \frac{1}{n}\sum_i (\mathbf{x_i} - \mathbf{q})(\mathbf{x_i} - \mathbf{q})^t$.

The probability of measuring a sample at 2D-point $\mathbf{x}$ contained in this cell is now modeled by the normal distribution $N(q, \Sigma)$:

$$p(\mathbf{x}) \sim \exp\left(-\frac{(\mathbf{x}-\mathbf{q})^t \Sigma^{-1} (\mathbf{x}-\mathbf{q})}{2}\right). \quad (1)$$

Similar to an occupancy grid, the NDT establishes a regular subdivision of the plane. But where the occupancy grid represents the probability of a cell being occupied, the NDT represents the probability of measuring a sample for each position within the cell. We use a cell size of 100 cm by 100 cm.

What's the use for this representation? We now have a piecewise continuous and differentiable description of the 2D plane in the form of a probability density. Before we show an example, we have to note two implementation details. cell和gird的区别?

To minimize effects of discretization, we decided to use four overlapping grids. That is, one grid with side length l of a single cell is placed first, then a second one, shifted by $\frac{l}{2}$ horizontally, a third one, shifted by $\frac{l}{2}$ vertically and finally a fourth one, shifted by $\frac{l}{2}$ horizontally and vertically. Now each 2D point falls into four cells. This will not be taken into account for the rest of the paper explicitly and we will describe our algorithm, as if there were only one cell per point. So if the probability density of a point is calculated, it is done with the tacit understanding, that the densities of all four cells are evaluated and the result is summed up.

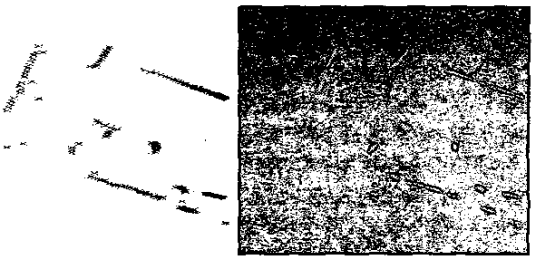

Fig. 1. An example of the NDT: The original laser scan and the resulting probability density.

A second issue is, that for a noise free measured world line, the covariance matrix will get singular and can not be inverted. In practice, the covariance matrix can sometimes get near singular. To prevent this effect, we check, whether the smaller eigenvalue of Σ is at least 0.001 times the larger eigenvalue. If not, it is set to this value. 特征值

Fig. 1 shows an example laser scan and a visualization of the resulting NDT. The visualization is created by evaluating the probability density at each point, bright areas indicate high probability densities. The next section shows, how this transformation is used to align two laser scans. 最终的结果是每一个网格内每个点的概率分布？？？

IV. Scan Alignment

The spatial mapping T between two robot coordinate frames is given by

$$T : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}, \quad (2)$$

where $(t_x, t_y)^t$ describes the translation and ϕ the rotation between the two frames. The goal of the scan alignment is to recover these parameters using the laser scans taken at two positions. The outline of the proposed approach, given two scans (the *first* one and the *second* one), is as follows:

1) Build the NDT of the first scan.
2) Initialize the estimate for the parameters (by zero or by using odometry data).
3) For each sample of the second scan: Map the reconstructed 2D point into the coordinate frame of the first scan according to the parameters.
4) Determine the corresponding normal distributions for each mapped point.

5) The score for the parameters is determined by evaluating the distribution for each mapped point and summing the result.
6) Calculate a new parameter estimate by trying to optimize the score. This is done by performing one step of Newton's Algorithm.
7) Goto 3 until a convergence criterion is met.

The first four steps are straightforward: Building the NDT was described in the last section. As noted above, odometry data could be used to initialize the estimate. Mapping the second scan is done using T and finding the corresponding normal distribution is a simple lookup in the grid of the NDT.

The rest is now described in detail using the following notation:

- $\mathbf{p} = (p_i)_{i=1..3}^t = (t_x, t_y, \phi)^t$: The vector of the parameters to estimate.
- $\mathbf{x_i}$: The reconstructed 2D point of laser scan sample i of the second scan in the coordinate frame of the second scan.
- $\mathbf{x_i'}$: The point $\mathbf{x_i}$ mapped into the coordinate frame of the first scan according to the parameters $\mathbf{p}$, that is $\mathbf{x_i'} = T(\mathbf{x_i}, \mathbf{p})$.
- $\mathbf{\Sigma_i}, \mathbf{q_i}$: The covariance matrix and the mean of the corresponding normal distribution to point $\mathbf{x_i'}$, looked up in the NDT of the first scan.

协方差矩阵和q应该对应于哪一个grid，概率密度函数来描述一个点落在哪里的"概率"

为什么要划分网格?

相当于每次只处理一小部分，用各部分的局部收敛形成全局的收敛?

The mapping according to $\mathbf{p}$ could be considered optimal, if the sum evaluating the normal distributions of all points $\mathbf{x_i'}$ with parameters $\mathbf{\Sigma_i}$ and $\mathbf{q_i}$ is a maximum. We call this sum the score of $\mathbf{p}$. It is defined as:

$$\texttt{score}(\mathbf{p}) = \sum_i \exp\left(\frac{-(\mathbf{x_i'} - \mathbf{q_i})^t \mathbf{\Sigma_i^{-1}} (\mathbf{x_i'} - \mathbf{q_i})}{2}\right). \quad (3)$$

This score is optimized in the next section.

V. Optimization using Newton's Algorithm

Since optimization problems normally are described as minimization problems, we will adopt our notation to this convention. Thus the function to be minimized in this section is $-$score. Newton's algorithm iteratively finds the parameters $\mathbf{p} = (p_i)^t$, that minimize a function f. Each iteration solves the following equation:

$$\mathbf{H}\Delta\mathbf{p} = -\mathbf{g} \quad (4)$$

Where $\mathbf{g}$ is the transposed gradient of f with entries

$$g_i = \frac{\partial f}{\partial p_i} \quad (5)$$

and $\mathbf{H}$ is the Hessian of f with entries

$$H_{ij} = \frac{\partial f}{\partial p_i \partial p_j}. \quad (6)$$

The solution of this linear system is an increment $\Delta\mathbf{p}$, which is added to the current estimate:

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \quad (7)$$

If $\mathbf{H}$ is positive definite, $f(\mathbf{p})$ will initially decrease in the direction of $\Delta\mathbf{p}$. If this is not the case, $\mathbf{H}$ is replaced by $\mathbf{H'} = \mathbf{H} + \lambda\mathbf{I}$, with λ chosen such, that $\mathbf{H'}$ is safely positive definite. Practical details on the minimization algorithm itself can be found for example in [4].

确保Hessian矩阵是正定的

This algorithm is now applied to the function $-$score. The gradient and the Hessian are built by collecting the partial derivatives of all summands of equation 3. For a shorter notation and to avoid confusing the parameter number i and the index of the laser scan sample i, the index i for the sample number is dropped. Additionally, we write

$$\mathbf{q} = \mathbf{x_i'} - \mathbf{q_i} \quad (8)$$

As can be verified easily, the partial derivatives of $\mathbf{q}$ with respect to $\mathbf{p}$ equal the partial derivatives of $\mathbf{x_i'}$. One summand s of $-$score is then given by

$$s = -\exp\frac{-\mathbf{q}^t\mathbf{\Sigma^{-1}}\mathbf{q}}{2}. \quad (9)$$

For on such summand, the entries of the gradient are then (using the chain rule):

向量的导数?

$$\begin{aligned} \tilde{g}_i &= -\frac{\partial s}{\partial p_i} = -\frac{\partial s}{\partial q}\frac{\partial q}{\partial p_i} \\ &= \mathbf{q}^t\mathbf{\Sigma^{-1}}\frac{\partial q}{\partial p_i}\exp\frac{-\mathbf{q}^t\mathbf{\Sigma^{-1}}\mathbf{q}}{2}. \end{aligned} \quad (10)$$

The partial derivatives of $\mathbf{q}$ with respect to p_i are given by the Jacobi matrix $\mathbf{J_T}$ of T (see equation 2):

$$\mathbf{J_T} = \begin{pmatrix} 1 & 0 & -x\sin\phi - y\cos\phi \\ 0 & 1 & x\cos\phi - y\sin\phi \end{pmatrix}. \quad (11)$$

A summand's entries in the Hessian $\mathbf{H}$ are given by:

$$\begin{aligned} \tilde{H}_{ij} = -\frac{\partial s}{\partial p_i \partial p_j} &= -\exp\frac{-\mathbf{q}^t\mathbf{\Sigma^{-1}}\mathbf{q}}{2} \\ &((-\mathbf{q}^t\mathbf{\Sigma^{-1}}\frac{\partial \mathbf{q}}{\partial p_i})(-\mathbf{q}^t\mathbf{\Sigma^{-1}}\frac{\partial \mathbf{q}}{\partial p_j}) + \\ &(-\mathbf{q}^t\mathbf{\Sigma^{-1}}\frac{\partial^2 \mathbf{q}}{\partial p_i \partial p_j}) + (-\frac{\partial \mathbf{q}^t}{\partial p_j}\mathbf{\Sigma^{-1}}\frac{\partial \mathbf{q}}{\partial p_i})) \end{aligned} \quad (12)$$

The second derivatives of $\mathbf{q}$ are (see eq. 11):

$$\frac{\partial^2 \mathbf{q}}{\partial p_i \partial p_j} = \begin{cases} \begin{pmatrix} -x\cos\phi + y\sin\phi \\ -x\sin\phi - y\cos\phi \end{pmatrix} & i = j = 3 \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{otherwise} \end{cases} \quad (13)$$

As can be seen from these equations, the computational costs to build the gradient and the Hessian are low. There

is only one call to the exponential function per point and a small number of multiplications. The trigonometric functions only depend on ϕ (the current estimate for the angle) and must therefore be called only once per iteration. The next two sections will now use this algorithm for position tracking and for SLAM.

VI. Position tracking

This section describes, how the scan match algorithm can be applied to tracking the current position from a given time $t = t_{start}$. The next section then extends this approach to SLAM.

The global reference coordinate frame is defined by the local robot coordinate frame at this time. The respective laser scan is called the *keyframe* in the following. Tracking is performed with respect to this keyframe. At time t_k, the algorithm performs the following steps:

1) Let δ be the estimate for the movement between time t_{k-1} and t_k (for example from the odometry).
2) Map the position estimate of time t_{k-1} according to δ.
3) Perform the optimization algorithm using the current scan, the NDT of the keyframe and the new position estimate.
4) Check, whether the the keyframe is "near" enough to the current scan. If yes, iterate. Otherwise take the last successfully matched scan as new keyframe.

The decision, whether a scan is still near enough is based on a simple empiric criterion involving the translational and the angular distance between the keyframe and the current frame and the resulting score. To be useful for position tracking, the algorithm has to be performed in real-time: Building the NDT of a scan needs around 10 ms on a 1.4 GHz machine. For small movements between scans the optimization algorithm typically needs around 1-5 iterations (and rarely more then ten). One iteration needs around 2 ms, so real-time is no problem.

VII. Application to SLAM

We define a map as a collection of keyframes together with their global poses. This section describes how to localize with respect to this map and how to extend and optimize this map, when the robot reaches unknown territory.

A. Localizing with respect to multiple scans

To each scan i in the map, an angle ϕ_i (or a rotation matrix $\mathbf{R}_i$) and a translation vector $(t_x, t_y)_i^t = \mathbf{T}_i$ is associated. These describe the pose of scan i in the global coordinate frame. The current robot pose is denoted by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{T}$. The mapping T' from the robot coordinate frame to the coordinate frame of scan i is then given by:

$$T' : \begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{R}_i^t(\mathbf{R}\begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{T} - \mathbf{T_i}) \tag{14}$$

Only small changes are required to adapt the algorithm of section V to this situation. The mapping of a 2D-point of scan i is now calculated by applying T'. Further, the Jacobian and the second partial derivatives of T' get now slightly more complicated. The Jacobian of the mapping is now given by:

$$\mathbf{J}_{\mathbf{T}'} = \mathbf{R_i}^t\mathbf{J}_{\mathbf{T}} \tag{15}$$

and second partial derivatives of T' are now given by:

$$\frac{\partial^2 \mathbf{x}'}{\partial p_i \partial p_j} = \begin{cases} \mathbf{R_i}^t \begin{pmatrix} -x\cos\phi + y\sin\phi \\ -x\sin\phi - y\cos\phi \end{pmatrix} & i = j = 3 \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{otherwise} \end{cases} \tag{16}$$

The gradient and the Hessian for the optimization algorithm could be built by summing over all overlapping scans. But we found an alternative, that is faster and yields equally good results: For each sample of the scan taken at the robot position, determine the scan, where the result of evaluating the probability density is maximal. Only this scan is used for this sample and for the current iteration. This way, the operations needed to build the gradient and the Hessian for the optimization algorithm are independent of the number of overlapping keyframes, except for finding the mentioned maximum.

B. Adding a new keyframe and optimizing the map

At each time step, the map consists of a set of keyframes with their poses in the global coordinate frame. If the overlap of the current scan with the map is too small, the map is extended by the last successfully matched scan. Then every overlapping scan is matched separately to the new keyframe, yielding the relative pose between the two scans. A graph is maintained, that holds the information of the pairwise matching result.

In this graph, every keyframe is represented by a node. A node holds the estimate for the pose of the keyframe in the global coordinate frame. An edge between two nodes indicates that the corresponding scans have been pairwise matched and holds the relative pose between the two scans.

After a new keyframe is added, the map is refined by optimizing an error function defined over the parameters of all keyframes. The results of the pairwise registration are used to define a quadratic error model for each matched pair as follows: The global parameters of two scans also define the relative pose between two scans. Let Δp be the difference between the relative pose defined by the global parameters and the relative pose defined by the result of the pairwise matching. Then we model the score

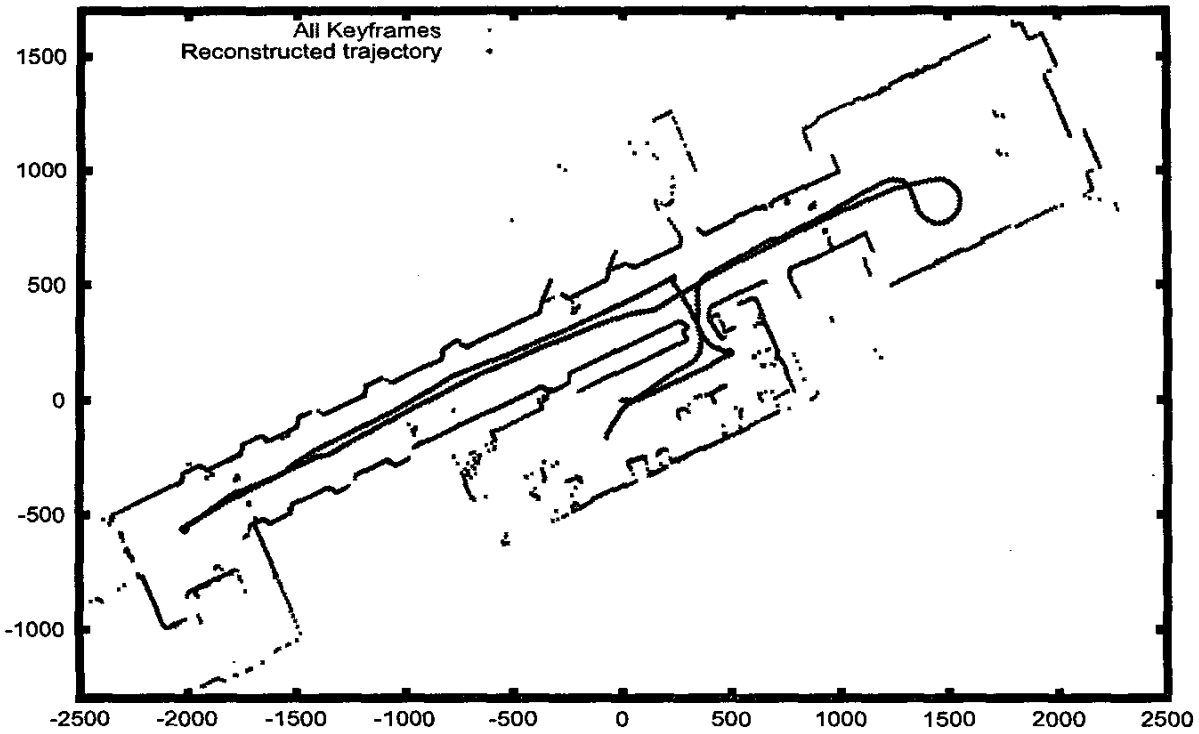

Fig. 2: A map, that was built using our scan matcher. Lengths are given in cm. Shown are the set of keyframes and the estimated trajectory (see video). More videos can be found on the authors' homepage [11].

of this two scans as function of Δp, using the quadratic model

$$\texttt{score}'(\Delta p) = \texttt{score} + \frac{1}{2}(\Delta p)^t \mathbf{H}(\Delta p). \quad (17)$$

Thereby score is the final score, when the pairwise matching had converged and $\mathbf{H}$ is the thereby obtained Hessian. This model is derived by a Taylor expansion of score around $\mathbf{\Delta p} = \mathbf{0}$ up to the quadratic term. Notice, that the linear term is missing, because we expanded about an extreme point. This score is now summed over all edges and optimized.

If the number of keyframes gets large, this minimization can no longer be performed under realtime conditions (the number of free parameters is $3N - 3$, where N is the number of keyframes). We therefore optimize only on a subgraph of the map. This subgraph is built by collecting all keyframes, which can be reached from the node of the new keyframe by traversing no more than three edges. We optimize the error function above now only with respect to the parameters, which belong to the keyframe contained in this subgraph. Of course, if a cycle had to be closed, we would have to optimize over all keyframes.

VIII. Results

The results we present now (and also the example in section 6) were performed *without using odometry*. This should demonstrate the robustness of the approach. Of course, as Thrun already noted in [15], this is only possible as long as any 2D structure is present in the world.

The built map presented in fig. 2 was acquired by driving the robot out of its lab, up the corridor, down the corridor and then back into the lab.

So the situation requires both the extension of the map and the localization with respect to the map. The robot collected 28430 laser scans during a travel of 20 minutes, where it traversed approximately 83 meters. The scans were taken with a SICK laser scanner covering 180 degree with an angular resolution of one degree. To simulate a higher speed, only every fifth scan was used. The simulated speed is then around 35 cm/s and the number of scans per second is around 23. The map was built using a combined strategy. The position tracker of section 4 was applied to every scan, whereas we initialize the parameters by extrapolating the result of the last time step linearly. Every tenth scan, the procedure of section VII was applied.

Fig. 2 shows the resulting map. Shown are the 33 keyframes, that the final map consisted of. A closer look

reveals also, that our scan match algorithm is tolerant against small changes in the environment like opened or closed doors. Processing all frames offline needs 58 seconds on a 1.4 GHz machine, that's 97 scans per second. More speed could perhaps be gained by porting our current implementation from Java to a faster language.

IX. Conclusion and Future Work

We have presented a new representation of range scans, the Normal Distributions Transform (NDT). This transform can be used to derive analytic expressions for matching another scan. We also showed, how our scan matcher could be incorporated for the problem of position tracking and for the SLAM problem. The major advantages of our method are:

- No explicit correspondences between points or features have to established. Since this is the most error-prone part in most approaches, we are more robust without correspondences.
- All derivatives can be calculated analytically. This is both fast and correct. continuous, differentiable

The question is of course: Can everything be modeled well enough by local normal distributions?

Up to now, our tests were performed in indoor environments, where this was never a problem. Further tests in less structured environments, preferably outdoors, are planned. We also intend to systematically compare the radius of convergence of our method with the method of Lu and Milios.

X. Acknowledgments

Peter Biber is funded by a grant of the Land Baden-Wuerttemberg, which is gratefully acknowledged. We would also like to thank Prof. Andreas Zell and Achim Lilienthal for the possibility to use the robot and the data acquisition software. We would also like to thank Sven Fleck for many fruitful discussions and help with the video.

XI. References

[1] P.J. Besl and N.D. McKay. A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

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

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

[4] J.E. Dennis and R. B. Schnabel. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM Classics in Applied Mathematics, 1996.

[5] J. Gutmann, W. Burghard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proc. International Conference on Intelligent Robots and System*, 1998.

[6] J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2000.

[7] J. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.

[8] J. Gutmann, T. Weigel, and B. Nebel. Fast, accurate, and robust self-localization in polygonal environments. In *Robocup workshop at IJCAI-99*, 1999.

[9] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *CVPR94*, pages 935–938, 1994.

[10] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[11] Peter Biber's homepage. *www.gris.uni − tuebingen.de/ ∼ biber*.

[12] Alexander Mojaev. *Umgebungswahrnehmung, Selbstlokalisation und Navigation mit einem mobilen Roboter*. PhD thesis, University of Tübingen, 2001.

[13] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle, June 1994.

[14] R. Szeliski and H.Y. Shum. Creating full view panoramic image mosaics and environment map. In *Computer Graphics (SIGGRAPH 97)*, pages 251–258, 1997.

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

[16] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2001.

[17] G. Weiss and E. von Puttkamer. A map based on laserscans without geometric interpretation. In *Proceedings of Intelligent Autonomous Systems 4 (IAS-4)*, pages 403–407, 1995.

[18] Z. Zhang. Iterative point matching for registration of free-from curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.