

# Indoor and Outdoor Localization for Fast Mobile Robots

Kai Lingemann, Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg

Fraunhofer Institute for Autonomous Intelligent Systems (AIS)

Schloss Birlinghoven, 53754 Sankt Augustin, Germany

E-mail: {lingemann|surmann|nuechter|hertzberg}@ais.fraunhofer.de

**Abstract**—This paper describes a novel, laser-based approach for tracking the pose of a high-speed mobile robot. The algorithm is outstanding in terms of accuracy and computational time, being 33 times faster than real time. The efficiency is achieved by a closed form solution for the matching of two laser scans, the use of natural landmarks and fast linear filters. The implemented algorithm is evaluated with the high-speed robot Kurt3D (4 m/s), and compared to standard scan matching methods in indoor and outdoor environments.

## I. INTRODUCTION

In order to fulfill user specified tasks, autonomous mobile robots have to be able to determine their pose (position and orientation) while moving – a precise and stable self-localization is one of the most important requirements to act successfully in any environment. For an application on flat ground, the pose  $\mathbf{p}$  of a robot is denoted by a triple  $\mathbf{p} = (x, y, \theta)$ , with position  $(x, y)$  and orientation  $\theta$ . Standard robotic approaches often use dead reckoning, e.g. odometry (counting wheel rotations), to estimate their position. But wheels slip, especially when driving with high speed, and miscalculation occurs<sup>1</sup>. These errors accumulate over time, and the robot’s position estimate becomes increasingly inaccurate. One method to improve localization is to track the robot’s pose with on-board laser range finders. There are, in principle, two different approaches to determine the pose of a robot:

- **Relative:** Given an arbitrary initial pose, e.g.,  $\mathbf{p}_0 = (0, 0, 0)$ , the robot’s current pose relative to  $\mathbf{p}_0$  is incrementally updated when the robot is moving. For this purpose, consecutively acquired sensor data, e.g., laser scan data, are compared. Let  $\mathcal{R}$  be the *reference* scan and  $\mathcal{S}$  be the *current* scan. If both scans have been acquired from different robot poses, a transformation that maps  $\mathcal{S}$  onto  $\mathcal{R}$  is calculated. This transformation corresponds to the movement of the robot between the scans.
- **Absolute:** Instead of tracking the pose incrementally, the position and orientation of the robot within an a priori given, assumable exact map or a known area with distinguishable active or passive landmarks is calculated.

In the following, we consider relative localization techniques, i.e., so-called *pose tracking* algorithms, since they do not impose any limitation to already mapped environments. Furthermore, the aim is not directly to build an accurate 2D map, but rather to ensure a stable and fast localization, regardless of the robot’s speed and without



Fig. 1. The mobile robot Kurt3D, equipped with AIS 3D laser scanner.

any restrictions of the covered distance. This aim reflects the underlying idea that localization, the preliminary stage of SLAM (*Simultaneous Localization And Mapping*), is not an end in itself, but the fundament of further algorithms, e.g., the building of three dimensional maps with a 3D sensor and solving 6D SLAM [1], [2]. The localization algorithm has to be computationally efficient and should not restrict the operational area or the speed of the mobile robot. Hence, subsequent algorithms for improving the calculated map, like cycle detection and correction [3] or global relaxation [4], are not considered here.

The implemented algorithm is evaluated with the high-speed robot Kurt3D (Figure 1), equipped with the AIS 3D laser scanner that is adjusted horizontally and used for localization purpose while driving [1].

### A. State of the Art

Relative localization algorithms based on proximity sensors are distinguishable by the manner of computing the pose change between two sensor inputs acquired from two different poses. We concentrate on methods using a laser range finder, since it has significant, critical advantages in accuracy and processing speed compared to sonar and stereo camera sensors. State of the art are algorithms comparing two scans directly (point-wise scan matching) [5] or utilizing special representations of the scanned data, e.g., histograms [6], features like extracted lines [7] or landmarks [8]. The latter ones are further distinguished by the type of landmarks, i.e., natural or artificial. In contrast to natural ones, artificial landmarks are, by definition, placed in the environment for simplifying the location task, being either active or passive.

The main drawbacks of current approaches are precision

<sup>1</sup>A video is available at <http://www.ais.fhg.de/ARC/kurt3D/HAYAI/videos/localization.skidding.divx.avi>.

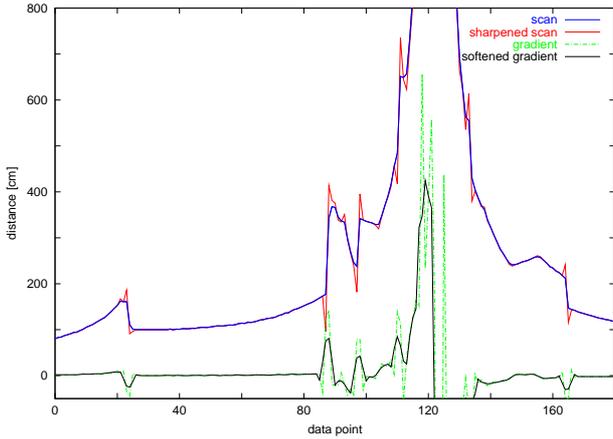


Fig. 2. Result of different filters used for feature extraction.

and computational cost, both in practical implementations as well as from a theoretical point of view, preventing a stable self localization of a robot driving with high-speed (e.g., 4 m/s), especially when operating in non-restricted domains<sup>2</sup>. Problems arise from the iterative way of calculating an optimal matching between two scans, preventing theoretically efficient computations, and from algorithms with complexities exponential in  $\mathcal{O}(n^k)$ ,  $k > 1$ , with  $n$  the number of distance values of a scan. To match two scans, usually an error function is defined, implementing a metric on scan matches as an evaluation of the matching. Minimizing the error function is a typical optimization problem with well-known problems. Generally, gradient descent methods are implemented, transforming the second scan such that the difference is reduced. Iteration leads to a local minimum of the error function. There are several algorithms that, in addition, compute the transformation within each epoch in an iterative way, employing a second level of iterations.

In this paper, we describe a method for computing the optimal transformation in a *closed form* manner, eliminating any iteration. For evaluating the proposed algorithm, it is compared to three main scan matching algorithms, in the implementation of S. Gutmann [9]:

- IDC (*Iterative Dual Correspondence*): Matching of raw data points of both scans. Instead of an elaborated feature detection, points are assigned according to their distance or angle [10].
- Cox: Data points from one scan are matched against lines, prior extracted from the other scan [11].
- Hist: Angle- and  $xy$ -histograms are calculated from both scans and compared based on a cross correlation function [12].

## II. SCAN MATCHING WITH HAYAI

This section describes the newly developed algorithm HAYAI (*Highspeed And Yet Accurate Indoor/outdoor-tracking*). The matching algorithm is based on the following scheme:

<sup>2</sup>A video is available at <http://www.ais.fhg.de/ARC/kurt3D/HAYAI/videos/kurt3d.corridor.divx.avi>.

- 1) Detect features within scan  $\mathcal{R}$ , yielding feature set  $M$  (*model set*). Likewise set  $D$  (*data set*) from scan  $\mathcal{S}$ .
- 2) Search for pairwise corresponding features from both sets, resulting in two subsets  $\check{M} \subseteq M$  and  $\check{D} \subseteq D$ .
- 3) Compute the pose shift  $\Delta \mathbf{p} = (\Delta x, \Delta y, \Delta \theta)^T$  as the optimal transformation for mapping  $\check{D}$  onto  $\check{M}$ .
- 4) Update the robot's pose  $\mathbf{p}_n \xrightarrow{\Delta \mathbf{p}} \mathbf{p}_{n+1}$  according to formula (1).
- 5) Save the current scan as new reference scan  $\mathcal{R} \leftarrow \mathcal{S}$ .

Given a pose  $\mathbf{p}_n = (x_n, y_n, \theta_n)$  and a transformation  $\Delta \mathbf{p} = (\Delta x, \Delta y, \Delta \theta)$ , the transition  $\mathbf{p}_n \xrightarrow{\Delta \mathbf{p}} \mathbf{p}_{n+1}$  is calculated as follows:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} + \begin{pmatrix} \cos \theta_n & \sin \theta_n & 0 \\ -\sin \theta_n & \cos \theta_n & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{pmatrix} \quad (1)$$

The following subsections describe the implemented data filtering techniques, extraction and matching of the used features as well as the mathematical background and practical matters of calculating the optimal transformation between two scans.

### A. Data Filtering

Scanning is noisy and small errors may occur, namely Gaussian noise and salt and pepper noise. The latter one arises for example at edges where the laser beam of the scanner hits two surfaces, resulting in a mean and erroneous data value. Furthermore reflections, e.g., at glass surfaces, lead to suspicious data. We propose two fast filtering methods to modify the data in order to enhance the quality of each scan, typically containing 181 data points. The data reduction, used for reducing Gaussian noise, works as follows: The scanner emits the laser beams in a spherical way, such that the data points close to the source are more dense. Multiple data points located close together are joined into one point. The number of these so-called *reduced points* is one order of magnitude smaller than the original one. For eliminating salt and pepper noise, a median filter removes the outliers by replacing a data point with the median value of the  $n$  surrounding points (here:  $n = 7$ ). The neighbor points are determined according to their index within the scan, since the laser scanner provides the data sorted in a counter-clockwise direction. The median value is calculated with regard to the Euclidian distance of the data points to the point of origin. In order to remove noisy data but leave the remaining scan points untouched, the filtering algorithm replaces a data point with the corresponding median value if and only if the Euclidian distance between both is larger than a fixed threshold (e.g., 200 cm).

### B. Extraction and Matching of Features

As described above, the scan matching algorithm computes a transformation  $\Delta \mathbf{p}$  such that a *set of features*, extracted from the first scan, is mapped optimally to a feature set of the second scan. In order to be usable for a pose tracking algorithm, features have to fulfill two requirements:

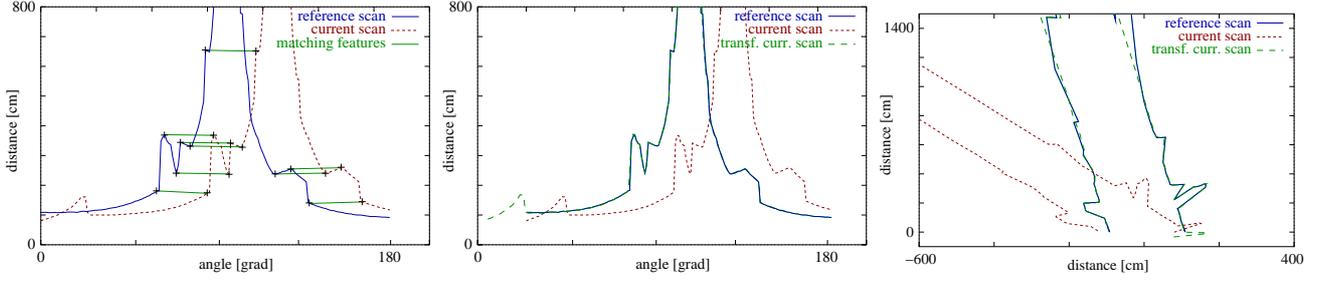


Fig. 3. Matching of features. Left: Detected features in both scans. Matched features are connected by a line. Middle: The second scan is transformed based on the calculated rotation & translation. Right: The transformed scan, in Cartesian coordinates.

- They have to be *invariant* with respect to rotation and translation.
- They have to be *efficiently* computable in order to satisfy real time constraints.

Using the inherent order of the scan data allows the application of linear filters for a fast and reliable feature detection. The described algorithm chooses *extrema* in the polar representation of a scan as natural landmarks. These extrema correlate to corners and jump edges in Cartesian space. The usage of polar coordinates implicates a reduction by one dimension, since all operations deployed for feature extraction are fast linear filters, operating on the sequence of range values  $(r_i)_{i \in \mathbb{N}}$  of a scan  $\mathcal{S} = ((\varphi_i, r_i))_{i=1, \dots, N}$ .

Given a one dimensional filter  $\Psi = [\psi_{-1}, \psi_0, \psi_{+1}]$ , the filtered value  $r_i^\Psi$  of a scan point  $r_i$  ( $i = 2, \dots, N-1$ ) is defined as  $r_i^\Psi = \sum_{k=-1}^1 \psi_k \cdot r_{i+k}$ . For feature detection, the scan signal is filtered as follows:

- 1) Sharpen the data in order to emphasize the significant parts of the scan, i.e., the extrema, without modifying the residual scan, by applying a sharpen filter of the form  $\Psi_1 = [-1, 4, -1]$ .
- 2) Compute of the derivation signal by using a gradient filter  $\Psi_2 = [-\frac{1}{2}, 0, \frac{1}{2}]$ .
- 3) Smooth the gradient signal to simplify the detection of zero crossings with a soften filter  $\Psi_3 = [1, 1, 1]$ .

Figure 2 illustrates the effects of the used filters.

After generating the sets of features  $M, D$  from both scans, a matching between both sets has to be calculated. Instead of solving the hard optimization problem of searching for an optimal match, we use a heuristic approach, utilizing inherent knowledge about the problem of matching features, e.g., the fact that the features' topology cannot change fundamentally from one scan to the following. The basic aim is to build a matrix of possible matching pairs, based on an error function defining the distance between two points  $\mathbf{m}_i, \mathbf{d}_j$ , with  $\mathbf{m}_i = (m_i^x, m_i^y)^T$  in Cartesian, or  $(m_i^\varphi, m_i^r)^T$  in polar coordinates, resp. ( $\mathbf{d}_j$  analogously):

$$\begin{aligned} \text{dist}(\mathbf{m}_i, \mathbf{d}_j) &= \sqrt{(\omega_1 \cdot (m_i^\varphi - d_j^\varphi))^2 + \omega_2 (m_i^r - d_j^r)^2} \\ &\quad + \omega_3 \cdot \sqrt{(m_i^x - d_j^x)^2 + (m_i^y - d_j^y)^2} \\ &\quad + \Theta(\mathbf{m}_i, \mathbf{d}_j) \end{aligned}$$

with constants  $(\omega_k)_{k \in \{1, 2, 3\}}$ , implementing a weighting between the polar and Cartesian distances. The function  $\Theta$

inhibits matchings between two features of different types:

$$\Theta(\mathbf{m}_i, \mathbf{d}_j) = \begin{cases} 0 & \Gamma(\mathbf{m}_i) = \Gamma(\mathbf{d}_j) \\ \infty & \text{else} \end{cases}$$

with a classification function  $\Gamma: (M \cup D) \mapsto \{\text{maximum, minimum, inflection point}\}$ . The resulting matrix denoting feature correspondences is simplified until the match is non-ambiguous. See figure 3 for a matching of two scans.

### C. Pose Calculation

Given two sets of features  $M = \{\mathbf{m}_i \mid \mathbf{m}_i \in \mathbb{R}^2, i = 1, \dots, N_m\}$  and  $D = \{\mathbf{d}_i \mid \mathbf{d}_i \in \mathbb{R}^2, i = 1, \dots, N_d\}$ , the calculation of the optimal transformation for mapping  $D$  onto  $M$  is an optimization problem. Intuitively, we are searching for a rotation  $\mathbf{R}_{\Delta\theta}$  by the angle  $\Delta\theta$  and a translation  $\Delta\mathbf{t} = (\Delta x, \Delta y)^T$  such that the differences between points from the reference and the corresponding points from the current scan, transformed with  $\Delta\mathbf{p} = (\Delta x, \Delta y, \Delta\theta)$ , are minimized. This results in a minimization of the error function

$$E = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \|\mathbf{m}_i - (\mathbf{R}_{\Delta\theta} \mathbf{d}_j + \Delta\mathbf{t})\|^2 \quad (2)$$

with weights  $w_{i,j} = 1$  iff feature  $\mathbf{m}_i$  corresponds to  $\mathbf{d}_j$ , and 0 otherwise.

Let  $\check{M} = \{\check{\mathbf{m}}_i\}_{i=1, \dots, N} \subseteq M$  and  $\check{D} = \{\check{\mathbf{d}}_i\}_{i=1, \dots, N} \subseteq D$  be sets of *matched* features, with  $\check{\mathbf{m}}_i \sim \check{\mathbf{d}}_i \forall i \in 1, \dots, N$ . Correspondence between two points,  $\check{\mathbf{m}}_i \sim \check{\mathbf{d}}_i$ , means that both data points denote the same point in the real scene. Therewith, the error function  $E$  simplifies to

$$E^{(1)} = \sum_{i=1}^N \|\check{\mathbf{m}}_i - (\mathbf{R}_{\Delta\theta} \check{\mathbf{d}}_i + \Delta\mathbf{t})\|^2. \quad (3)$$

Let  $\widehat{M}, \widehat{D}$  be variants of  $\check{M}, \check{D}$  so that each point is shifted by the centroid of the respective set:

$$\begin{aligned} \widehat{M} &= \{\widehat{\mathbf{m}}_i \mid \widehat{\mathbf{m}}_i = \check{\mathbf{m}}_i - \check{\mathbf{c}}_m\}_{i=1, \dots, N} \\ \widehat{D} &= \{\widehat{\mathbf{d}}_i \mid \widehat{\mathbf{d}}_i = \check{\mathbf{d}}_i - \check{\mathbf{c}}_d\}_{i=1, \dots, N} \end{aligned}$$

with the centroids defined as

$$\check{\mathbf{c}}_m = \frac{1}{N} \sum_{i=1}^N \check{\mathbf{m}}_i \quad \check{\mathbf{c}}_d = \frac{1}{N} \sum_{i=1}^N \check{\mathbf{d}}_i.$$

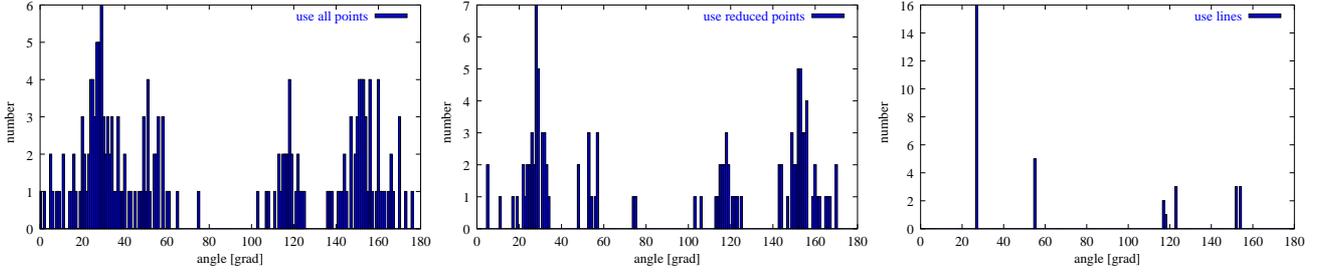


Fig. 4. Angle histograms. Left: Due to data noise, the histogram degenerates. Middle: Application of a reduction filter leads to clearer histograms. Right: Using a line extraction filter results in clear, singular peaks.

Substitution into formula (3) leads to

$$E^{(1)} = \sum_{i=1}^N \left\| \widehat{\mathbf{m}}_i - \mathbf{R}_{\Delta\theta} \widehat{\mathbf{d}}_i - \underbrace{(\Delta\mathbf{t} - \check{\mathbf{c}}_m + \mathbf{R}_{\Delta\theta} \check{\mathbf{c}}_d)}_{=\Delta\tilde{\mathbf{t}}} \right\|^2$$

$$= \sum_{i=1}^N \left\| \widehat{\mathbf{m}}_i - \mathbf{R}_{\Delta\theta} \widehat{\mathbf{d}}_i \right\|^2 \quad (4a)$$

$$- 2\Delta\tilde{\mathbf{t}} \cdot \sum_{i=1}^N (\widehat{\mathbf{m}}_i - \mathbf{R}_{\Delta\theta} \widehat{\mathbf{d}}_i) \quad (4b)$$

$$+ \sum_{i=1}^N \|\Delta\tilde{\mathbf{t}}\|^2. \quad (4c)$$

The second term (4b) is zero since the measurements are referred to the centroids. The third term (4c) is minimal at  $\Delta\tilde{\mathbf{t}} = 0$ , i.e.,  $\Delta\tilde{\mathbf{t}} = \check{\mathbf{c}}_d - \mathbf{R}_{\Delta\theta} \check{\mathbf{c}}_d$ . Therefore,  $E^{(1)}$  is minimal iff  $E^{(2)}$  is, with  $E^{(2)}$  depending only on the rotation [13]:

$$E^{(2)} = \sum_{i=1}^N \left\| \widehat{\mathbf{m}}_i - (\mathbf{R}_{\Delta\theta} \widehat{\mathbf{d}}_i) \right\|^2.$$

Since rotating does not change the length of a vector, in particular  $\|\mathbf{R}_{\Delta\theta} \widehat{\mathbf{d}}_j\|^2 \equiv \|\widehat{\mathbf{d}}_j\|^2$ , it is sufficient to calculate the maximum of

$$E^{(3)}(\mathbf{R}_{\Delta\theta}) = \sum_{i=1}^N \widehat{\mathbf{m}}_i \cdot \mathbf{R}_{\Delta\theta} \widehat{\mathbf{d}}_i. \quad (5)$$

By solving the equation  $\frac{\partial}{\partial \Delta\theta} E^{(3)} = 0$ , the rotation is calculated as

$$\Delta\theta = \arctan \left( \frac{\sum_{i=1}^N (\widehat{m}_i^x \cdot \widehat{d}_i^x + \widehat{m}_i^y \cdot \widehat{d}_i^y)}{\sum_{i=1}^N (\widehat{m}_i^y \cdot \widehat{d}_i^x - \widehat{m}_i^x \cdot \widehat{d}_i^y)} \right). \quad (6)$$

With given rotation, the translation is calculated as follows:

$$\underbrace{\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}}_{=\Delta\mathbf{t}} = \begin{pmatrix} \check{c}_m^x \\ \check{c}_m^y \end{pmatrix} - \underbrace{\begin{pmatrix} \cos \Delta\theta & \sin \Delta\theta \\ -\sin \Delta\theta & \cos \Delta\theta \end{pmatrix}}_{=\mathbf{R}_{\Delta\theta}} \cdot \begin{pmatrix} \check{c}_d^x \\ \check{c}_d^y \end{pmatrix}. \quad (7)$$

For calculating a transformation  $\Delta\mathbf{p}$  between two scans as described, at least *two* features  $\in \mathbb{R}^2$  are needed, since the four coordinates are sufficient to determine the three degrees of freedom (translation in  $x$ - and  $y$ -direction,

rotation). The following situations where it is impossible to find two corresponding features may occur:

- 1) *The two scans are too diverse to be matched.* Naturally, every relative localization algorithm is depending on an adequate overlap of two successive sensor inputs. In practice, however, this theoretical problem proved to be nonexistent for the proposed algorithm. Since the mobile robot platform Kurt3D that was used to evaluate HAYAI drives with a maximum speed of 4 m/s, the scanner's frequency is 75 Hz, and the implemented algorithm reaches an average frequency of 2300 Hz on the robot's on-board computer (Pentium III-800 MHz), it is guaranteed that *every* acquired scan is processed. Hence, the maximal difference between two scans is  $\Delta t = 6.43$  cm (forward translation) respectively  $\Delta\theta = 25.58^\circ$  (rotation on the spot with maximal angular velocity). Various experiments in miscellaneous environments showed that the algorithm is capable of matching reliably much larger differences.
- 2) *It is not possible to detect a sufficient number of matchable features in the scans.* This general problem of every feature-based scan matching technique may occur either if there are too few objects within the scanner's range, or the scenery is unsuitable to detect features. The first case is very unlikely, since the range of laser scanners is generally sufficiently high (in case of the model deployed here, 32 m), and the features used in this algorithm do exist frequently in a real world application. An example of the second case is an office environment, when the robot is turning at the end of a corridor and is directly facing a corner, detecting this corner as a sole feature. To cope with this exceptional case, we calculate angle-histograms to estimate the rotation. This estimation is used to support the correct mapping of the corresponding features, and furthermore enables the transformation calculation based on only one feature by applying equation (7).

#### D. Additive: Rotation Estimation with Angle Histograms

Regarding the single data points of a scan as vectors makes it possible to associate each data point with an angle, i.e.,  $\alpha(i)$  of point  $i = (x_i, y_i)$  is defined as the angle between  $x$ -axis and the vector from  $i$  to  $i+1$ , with

$$\alpha(i) = \arctan \left( \frac{x_{i+1} - x_i}{y_{i+1} - y_i} \right).$$

Discretizing and plotting the angles against their frequency leads to a histogram as a characteristic of the scan that is invariant with regards to rotation, that is, a movement of the robot results in a translational displacement of the circular angle histogram (Figure 4, right). Likewise, the shift between two histograms extracted from two different scans are used to calculate the rotation  $\Delta\theta$  of the robot between taking these scans. More precisely, given two histograms  $\mathcal{H}, \mathcal{G}$ , the angle  $\Delta\theta$  is computed by  $\Delta\theta = \arg \max_i K_i(\mathcal{H}, \mathcal{G})$ , where the error function  $K_i$  measures the similarity of the two scans. Weiß et al. suggest the use of a cross correlation function [12],

$$K_i(\mathcal{H}, \mathcal{G}) = \sum_{j=1}^n \mathcal{H}(j) \cdot \mathcal{G}((j+i) \bmod (n+1)).$$

In practice, Gaussian noise of the sensor leads to small variances of the scanned data points that in turn imposes significant differences of the calculated angles. Therefore, we apply the reduction filter as described in section II-A first, resulting in a much clearer histogram. Figure 4 shows the results of both methods, as well as a histogram calculation based on lines extracted from the scan. For a description of the real-time capable line filter, see [14].

### III. RESULTS

The algorithm has been tested in various indoor and outdoor environments. This paragraph demonstrates results of localization in different static and dynamic scenarios.

#### A. Indoor Tracking

Figure 5 shows the result of a 160 m cyclic drive in the office corridor of the Fraunhofer Institute AIS. The map was generated from 12727 scans, gaged during the drive, and is the direct result of the scan matching algorithm, *without* fusion with odometry, and without additional corrective algorithms like global relaxation or explicit correction of cycles, in order to illustrate the quality of the localization. In Figure 5 (bottom) the calculated trajectory, titled with HAYAI, as well as the results of other standard scan matching algorithms and the robot's odometry are presented. Apparently, these reference algorithms are incapable of processing each incoming scans, since small errors in the calculations accumulate in such a way that the resulting trajectories are useless. Artificial off-line reduction of the scanner's frequency to 10 Hz, i.e., taking about every 7th scan only, improves the result of the comparative algorithms, but is no general solution, since this procedure fails at higher speed of the robot: With increasing velocity, the distance between two acquired scans increases as well, and soon becomes too large for a reliable matching.

Besides accuracy, another important criterion to evaluate the quality of a localization method is computation time. A slow algorithm imposes a hard restriction on the robot's velocity, or leads to periodical loss of acquired scans if they cannot be processed in real time. Skipping scans means to deal with larger pose changes between successive scans – if the robot drives fast enough, the scan matching will become unreliable due to a too small overlap, the pose

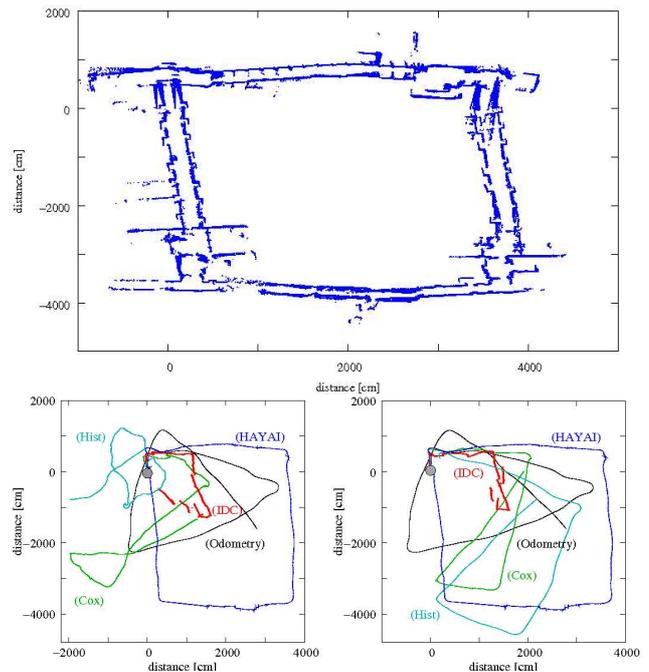


Fig. 5. Result of a 160 m cyclic drive, inside an office building. Top: The resulting map, based on the scan matching algorithm only, *without* fusion with odometry or application of additional corrective algorithms. Bottom left: Trajectories, calculated by different algorithms, based on all 12727 scans (75 scans per second). Bottom right: Artificial off-line reduction to 10 Hz enables the reference algorithms to improve their calculations.

TABLE I  
SPEED COMPARISON OF SCAN MATCHING ALGORITHMS: TIME [SEC] NEEDED TO CALCULATE THE TRAJECTORIES FROM FIGURE 5 (LEFT).

HAYAI	Cox	Hist	IDC
5.41	30.86	213.26	47.09

will be highly incorrect or lost completely. Furthermore, even slower moving robots profit from a fast, efficient algorithm, since it enables the on-board computer to carry out other, computationally expensive tasks, besides the robot localization. Table I compares the times needed to calculate the trajectories from the indoor experiment described above, measured on a Pentium III-800 MHz.

#### B. Outdoor Tracking

Outdoor localization is particularly demanding for an algorithm, since it has to deal with highly irregular and non-static environments, where even small changes in the pose lead to significant changes in the corresponding scans. For example, scans of plants result in varying output due to many jump edges and reflections on moving leaves and branches. Other outdoor capable localization methods typically depend on some kind of absolute information, like an a priori map of the environment, active beacons or GPS data. Contrary to the reference scan matching algorithms, HAYAI proved to be able to localize the robot in a clearly unstructured environment (Figure 7), as long as the ground itself is roughly even so that there are no significant changes of the robot's pitch and roll angle.

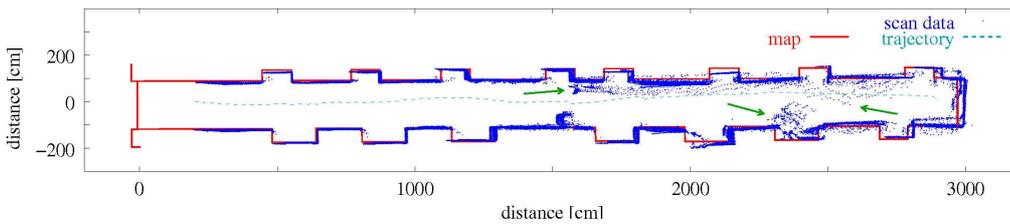


Fig. 6. Left: Pose tracking in dynamic environments. The red plotted map serves as ground truth, the calculated trajectory is based on scan matching only, without fusion with odometry. Green arrows mark scan data of dynamic objects (people walking around). Right: Photo of the corridor.

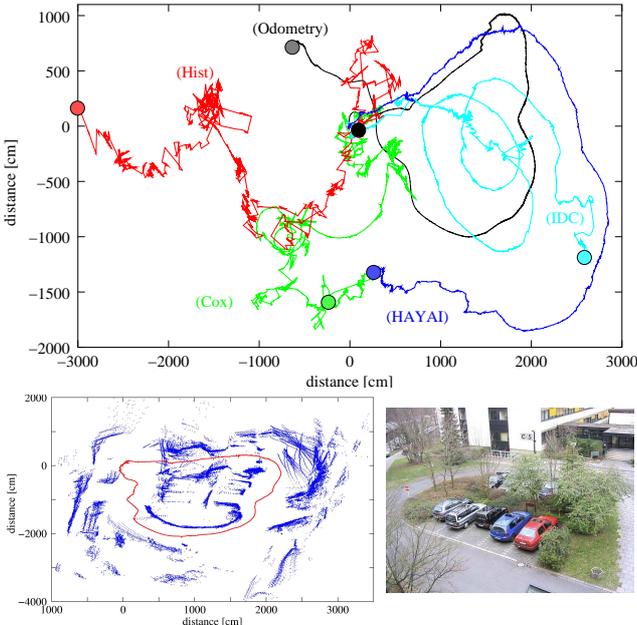


Fig. 7. Localization during a 84 m cyclic outdoor drive. Top: Calculated trajectories from HAYAI, other scan matching algorithms, and odometry. Bottom left: Map generated from 6903 scans, with calculated, exact, final trajectory (Kalman-filtered with odometry). Bottom right: Picture of the scene (rotated 90° counterclockwise).

### C. Dynamic Environments

In practice, the described algorithm showed to be robust even in dynamic environments, like scenarios with people walking. Due to the scanner's high clock rate of 75 Hz, the environment does not change significantly between two scans. In addition, the use of abstracted features instead of whole scans and an accurate calculation of corresponding features yields an inherent robustness, since data points that appear suddenly and do not match correctly with the previous scan features are ignored. Figure 6 shows a typical example of such an experiment. Comparison between ground truth map and plotted scans shows that the accuracy of the localization is still very high.

## IV. CONCLUSION

Mobile robots depend on a stable and reliable self localization. The described, newly developed localization algorithm, based on pairwise scan matching, is capable of tracking a robot's trajectory in indoor as well as outdoor environments, driving with a speed of up to 4 m/s. Comparison with three state of the art scan matching algorithms showed the outstanding performance of HAYAI

in terms of precision and computational speed. It is able to process scans with up to 2300 Hz on a moderate laptop (Pentium III-800 MHz) – if only the scanner supplied the data so fast. In reality, the scanner provides the data with 75 Hz, that is one scan every 5.3 cm at a speed of 4 m/s. The scan matching algorithm HAYAI is based on matching features that are naturally given in any standard operational environment and are extracted efficiently in the order of  $\mathcal{O}(n)$ . The method is not restricted to polygonal environments, independent from structural alteration of the surrounding like beacons or artificial landmarks, and proved to be stable even in dynamic environments.

## REFERENCES

- [1] H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. 6D SLAM – Preliminary Report on closing the loop in Six Dimensions. In *Proc. of the IFAC IAV*, page (accepted), June 2004.
- [2] H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45:181 – 198, 12 2003.
- [3] S. Thrun, D. Fox, and W. Burgard. A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping. In *Proc. of the IEEE ICRA*, April 2000.
- [4] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. In *Autonomous Robots*, volume 4, pages 333–349, 1997.
- [5] S. Pfister, K. Kriechbaum, S. Roumeliotis, and J. Burdick. Weighted Range Sensor Matching Algorithms for Mobile Robot Displacement Estimation. In *Proc. of the IEEE ICRA*, May 2002.
- [6] J.-H. Moon, B.-J. You, H. Kim, and S.-R. Oh. Fast Markov Localization using Angle-Histogram. In *Proc. of the IEEE ICAR*, pages 411–416, June 2003.
- [7] K. J. Kyriakopoulos and N. Skounakis. Moving Obstacle Detection for a Skid-Steered Vehicle Endowed with a Single 2-D Laser Scanner. In *Proc. of the IEEE ICRA*, pages 7–12, September 2003.
- [8] E. Santiso, M. Mazo, J. Ureña, J. A. Jiménez, J. J. García, and M. C. Serra. Mobile Robot Positioning with Natural Landmark. In *Proc. of the IEEE ICAR*, pages 47–52, June 2003.
- [9] J. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proc. of the IEEE Eurobot*, 1996.
- [10] F. Lu and E. Milios. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In *Proc. of the IEEE CVPR*, pages 935–938, 1994.
- [11] I. J. Cox. Blanche – An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle. *Journal of the IEEE TRA*, 7(2):193–204, July 1991.
- [12] G. Weiß, C. Wetzler, and E. von Puttkamer. Keeping Track of Position and Orientation of Moving Indoor Systems by Correlation of Range-Finder Scans. In *Proc. of the IEEE/RSJ IROS*, pages 595–601, 1994.
- [13] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, April 1987.
- [14] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg. Fast acquiring and analysis of three dimensional laser range data. In *Proc. of the Int. Fall Workshop VMV*, pages 59–66, November 2001.