2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Normal Distributions Transform Monte-Carlo Localization (NDT-MCL)

Jari Saarinen, Henrik Andreasson, Todor Stoyanov and Achim J. Lilienthal

Abstract— **Industrial applications often impose hard requirements on the precision of autonomous vehicle systems. As a consequence industrial Automatically Guided Vehicle (AGV) systems still use high-cost infrastructure based positioning solutions. In this paper we propose a map based localization method that fulfills the requirements on precision and repeatability, typical for industrial application scenarios. The proposed method - Normal Distributions Transform Monte Carlo Localization (NDT-MCL) is based on a well established probabilistic framework. In a novel contribution, we formulate the MCL localization approach using the Normal Distributions Transform (NDT) as an underlying representation for both map and sensor data. By relaxing the hard discretization assumption imposed by grid-map models and utilizing the piecewise continuous NDT representation the proposed algorithm achieves substantially improved accuracy and repeatability. The proposed NDT-MCL algorithm is evaluated using offline data sets from both a laboratory and a real-world industrial environments. Additionally, we report a comparison of the proposed algorithm to grid-based MCL and to a commercial localization system when used in a closed-loop with the control system of an AGV platform. In all tests the proposed algorithm is demonstrated to provide performance superior to that of standard grid-based MCL and comparable to the performance of the commercial infrastructure based positioning system.**

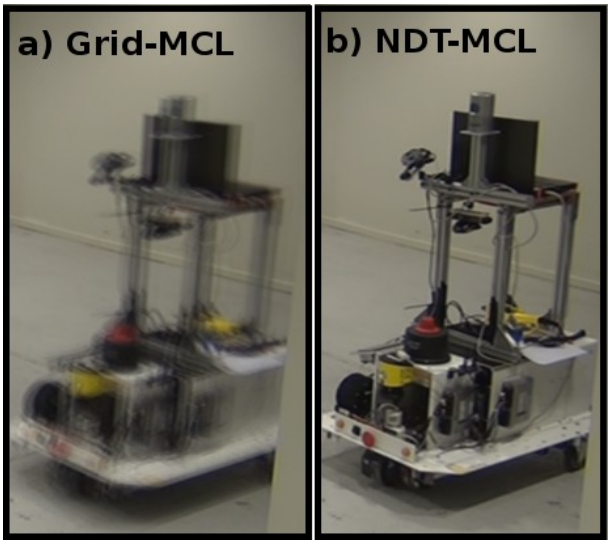

Fig. 1. Positioning accuracy visualized: Fifteen images recorded from a stationary camera overlaid in positioning accuracy test using a) grid-based MCL and b) NDT-based MCL.

I. Introduction

Localization systems are an essential enabling component of mobile robotic systems. The importance of accurate positioning is even higher in modern logistics application scenarios, which rely on precise and repeatable trajectory following using Automatically Guided Vehicles (AGV). An AGV typically has to achieve positioning accuracy of under 3cm, especially during loading operations. Moreover, as AGVs are typically non-holonomic platforms, sufficient accuracy has to be maintained throughout the trajectory in order to guarantee accuracy in orientation at loading points. In addition to being accurate, the pose estimate must be smooth and updated in real-time in order to avoid violent corrective control actions which can be catastrophic for heavily loaded vehicles. As a consequence of hard requirements, industry is still using positioning methods that rely on fixed infrastructure in the factories. Setting up a localization infrastructure is, however, a high-cost investment and thus there is a strong demand for more flexible solutions.

Monte Carlo Localization (MCL) [3] is among the most popular localization approaches in the robotics research community. MCL is a probabilistic map based localization approach that has been shown to be robust in real-world scenarios [3], [15], [9]. However, in prior contributions MCL has not been reported to reach the precision requested by industrial applications [3], [15], [9]. Röwekämper et al. in [9] analyzed the accuracy of an MCL localization system with KLD-sampling and reported mean MCL errors of approx. 6cm in a repeatability test. Their solution for improving the accuracy was to perform scan matching alignment to reach accurately the final pose. Unfortunately this approach only works for holonomic vehicles, while the majority of AGVs are non-holonomic and cannot reliably correct their position and orientation over a short trajectory.

The insufficient accuracy of the standard MCL algorithm is further confirmed by our experiments. Fig. 1a visualizes the positioning error of a standard MCL approach. The figure is composed of fifteen overlaid pictures recorded with a stationary camera during a positioning accuracy test. In this experiment the vehicle was instructed to repeatedly execute a pre-determined trajectory and stop at the same location at the end of each trial. The blurriness of Fig. 1a demonstrates that even in a simple stationary environment the standard MCL approach fails to meet the accuracy requirements needed in typical industrial applications. We argue that the main reason for this failure stems from the grid-based representation used by MCL. Occupancy grids [8], used for map representation in the standard MCL approach, are discretized by definition and thus cannot represent the environment accurately, especially in presence of sensor noise.

In this paper we propose to use Normal Distributions Transform (NDT) representation for an MCL implementa-

All autohors are with Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. Email: {first name}.{last name}@oru.se. This work has partly been supported by the European Commission under contract number FP7-ICT-600877 (SPENCER).

tion. The NDT is a piecewise continuous representation, which represents the space as a set of normal distributions [1], [6]. It is a compact representation allowing to model the environment with a significantly lower number of cells than occupancy grid maps. Moreover, NDT is by nature a likelihood model, which makes it an ideal candidate to be utilized in MCL. The accuracy of the proposed Normal Distributions Transform based Monte Carlo Localization (NDT-MCL) is demonstrated in Fig. 1b, which is obtained through the same procedure as for the grid-based MCL (Fig 1a).

The main contribution of this paper is to formulate NDT-MCL and to evaluate it using both pre-recorded datasets and in a closed-loop fashion on board of an AGV test platform. The experiments performed confirm that NDT-MCL provides accuracy and repeatability superior to those of the standard grid-based MCL and comparable to the performance of a commercial infrastructure based system.

The rest of this paper is organized as follows. Sec. 2 first introduces and formalizes the standard MCL approach, before presenting the proposed NDT-MCL algorithm. Sec. 3 describes the test setup and analyzes the obtained results and finally Sec. 4 concludes the paper with discussion.

II. NDT-MCL

A. Grid-MCL

In this paper we follow closely the formulation for Monte Carlo Localization (MCL) introduced in [3]. MCL is most commonly used with 2D occupancy grid maps [8], and therefore we refer to it as grid-MCL, in order to distinguish it from the proposed NDT-based approach. MCL is a recursive map based localization algorithm, which estimates the posterior of the robot's pose $p(x_t|u_t, z_t, m)$, given a map m, a sequence of controls $u_t = \{u^1, ..., u^t\}$ and a sequence of observations $z_t = \{z^1, ..., z^t\}$, i.e,

$$\begin{array}{c} p(x_t|u_t, z_t, m) = \eta p(z_t|x_t, m)\cdot \\ \int_{x_{t-1}} p(x_t|u_t, x_{t-1}) p(x_{t-1}|z_{t-1}, u_{t-1}, m) dx_{t-1}, \end{array} , \quad (1)$$

where η is a scaling factor, $p(z_t|x_t, m)$ is a measurement model, $p(x_t|u_t, x_{t-1})$ is a motion model and $p(x_{t-1}|z_{t-1}, u_{t-1}, m)$ is the posterior of the previous state. The motion model describes the probability that a control u_t causes the state change from x_{t-1} to x_t. The measurement model provides the probability of observing z^t from a state x_t given a map m.

MCL uses a particle filter approach to solve Eq. 1. It estimates the posterior of pose with a set of N weighted particles:

$$p(x_t|u_t, z_t, m) \propto \left\{x_t^i, w_t^i\right\}_{i=1}^{N}, \quad (2)$$

where x_t^i is one pose sample and w_t^i is a weight associated to it. In this paper we are considering localization in 2D-plane, so a pose $x_t^i \in \mathbb{R}^3$ is represented with x,y position and a heading of the robot.

MCL estimation consists of three steps: 1) prediction, 2) update and 3) resampling. The prediction step implements the motion model by applying the control for each particle and generating noise according to the model. In this paper we use the so called odometry model. Given two odometry readings x_{t-1}^o and x_t^o we compute the differential motion dx_{t-1}^t with respect to x_{t-1}^o, so that $x_t^o = x_{t-1}^o \oplus dx_{t-1}^t$, where $\oplus$ is the 2D pose compounding operator as introduced in [11]. Since dx_{t-1}^t is affected by noise in the odometry measurement, a normally distributed noise vector $\sigma_i \in \mathbb{R}^3$, proportional to relative motion, is sampled for each particle. In this paper we use 10% odometry error in generation of the noise. The prediction is done by transforming each particle pose according to Eq. 3:

$$x_t^i = x_{t-1}^i \oplus (dx_{t-1}^t + \sigma_i). \quad (3)$$

Next, in the update step, the measurement model is used to evaluate the likelihood of observation given a state. A beam sensor model was used in the original formulation [3]. The beam model predicts the measurement using a raycasting approach and the map, and the predicted measurement is compared against the real measurement in order to compute the likelihood. This approach was found to result in degeneration of the particle distribution with accurate sensors, such as laser scanners [15]. An alternative, likelihood field, or end point model, was introduced in [14]. The likelihood field model is generated by computing for each cell in the map the likelihood for observing a point in the particular cell. The likelihood field model is computationally lighter and it has been shown to produce smoother posterior distributions than the beam model [14].

Finally, a resampling step is needed in order to avoid the divergence of the particles [7], [2]. The resampling step replaces the particles with low weight, with ones that are representing the posterior better. An efficient sampling strategy, called KLD-sampling, was introduced by Fox in [4]. The baseline implementation for grid-MCL uses this strategy, however, NDT-MCL implementation presented in this paper runs without any adaptation scheme.

B. NDT-MCL

In this paper we use NDT to represent both, the map and the measurements. The NDT was first introduced by Biber and Strasser [1] for 2D scan matching and it was later extended to 3D [6]. Here we assume that a measurement is represented in the sensor coordinate frame with a set of N_t points $z^t = \{p_i\}_{i=1}^{N_t}$, where $p_i \in \mathbb{R}^2$ or $p_i \in \mathbb{R}^3$. The measurement is transformed to an NDT representation by accumulating the sensor measurements to a regular grid with given resolution and then computing the mean and covariance estimates for each grid cell containing points. After this process, the measurement is represented with a set of N_{zt} normal distribution parameters $\bar{z}_t = \{\mu_i, \Sigma_i\}_{i=1}^{N_{zt}}$, where typically $N_{zt} \ll N_t$. The result describes the probability of a point being measured at a particular physical location. Thus, the NDT by nature is a likelihood model [1] and moreover, it is a piecewise continuous representation.

The NDT map is based on a regular grid containing estimated normal distribution parameters of the mapped environment. The map can be built during a short-term operation in a similar manner than described above, but accumulating points to the map over several observations. However, in this paper we used the recursive implementation described in [10] to build the static map. The map is maintained in grid representation for an efficient access to the Gaussian components, however, for a sake of notation, let us assume that the map is a set of normal distribution parameters $m = \{\mu_j, \Sigma_j\}_{j=1}^{N_m}$.

Next let us investigate one particle pose x_k. This pose can be represented as a rotation matrix R_k and translation t_k with respect to the global frame of reference. Now, the likelihood between $\bar{z}_t$, given map m and state x_i can be given as L_2-likelihood [12]:

$$L_2^k(\bar{z}_t|x_k, m) = \sum_{j=1}^{N_m}\sum_{i=1}^{N_z} d_1 exp(-\frac{d_2}{2}\mu_{ij}^T(R_k\Sigma_i R_k^T + \Sigma_j)^{-1}\mu_{ij}) \quad , \quad (4)$$

where $\mu_{ij} = R_k\mu_i + t_k - \mu_j$, and d_1 and d_2 are scaling parameters. Eq. 4 computes the likelihood over all observations and all map components. In practice we approximate Eq. 4 by finding the cell from the map that corresponds to the mean $\bar{\mu}_i = R_k\mu_i + t_k$ and search the local neighborhood in order to find the closest normal distribution to $\bar{\mu}_i$. The update step computes the weights for each particle as:

$$w_t^k = \frac{1}{\sum_{i=1}^{N} w_t^i} w_{t-1}^k L_2^k. \quad (5)$$

In this paper we use a basic resampling approach [2]. The resampling is triggered when the variance of the weights grows over a given threshold. This way resampling is done when necessary. For the final pose estimate we use the maximum aposteriori estimate over the particles.

In this paper we do not analyze the global initialization and thus the filter is initialized to a known initial location with given variance.

III. Tests and Results

A. Test setup

We evaluated NDT-MCL using a commercial Automatically Guided Vehicle (AGV) system from Kollmorgen. A Vehicle Master Controller (VMC 500) controls the vehicle along the predefined trajectories. The localization is based on a commercial infrastructure based positioning system, which tracks wall mounted reflectors using a rotating laser [5]. Since 1991 there have been approx. 15000 AGVs' deployed using the VMC system and approx. 10000 of those are using reflector based positioning. The reflector based localization requires pre-installation and calibration of the reflector network. Once properly set-up and calibrated it provides an accurate position information, which in this paper is used as a baseline for evaluation. The test vehicle is a training platform for AGV operators, which has the same kinematics and hardware as commercial AGV's but is smaller in size (see Fig. 2).

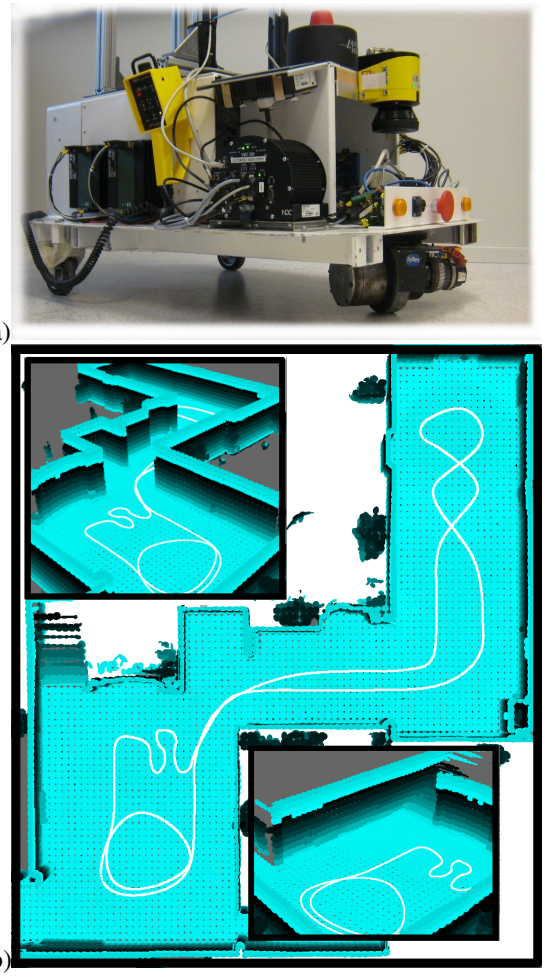

Fig. 2. Test vehicle (a) and (b) the map of the test environment with the trajectory that the vehicle was set to drive in most of the tests.

The primary comparison in this paper is done between grid-MCL and NDT-MCL. The implementation for grid-MCL was chosen to be AMCL[1], which is a mature and widely used localization node of ROS. In the experiments we used the likelihood field model, with a maximum of 100 beams, and adaptation range set from 100 to 1000 particles. These parameters were selected in order to have as much particles and measurements as possible, while still maintaining the real-time constraints. The odometry model was set to differential steered and the motion model parameters were tuned so that the filter provided consistently reliable estimates in real-time.

Our tests include two different scenarios 1) offline and 2) closed-loop. In the offline scenarios the vehicle was controlled using the reflector-based position, and the data from odometry, SICK S300 safety laser and the reflector-based position were logged. The evaluated localization methods used odometry and SICK S300 data as inputs. The evaluation was done by offline processing the data using the two approaches and the accuracy was compared against the pose given by the reflector-based localization system.

[1] http://www.ros.org/wiki/amcl

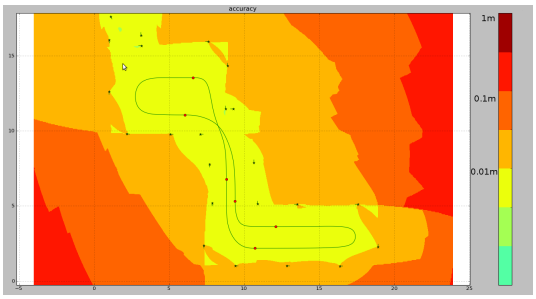

Fig. 3. Expected reflector based positioning accuracy in basement setup. The accuracy of the reflector based positioning system should be 1cm or less through out the test area.

In closed-loop testing the vehicle was controlled using the pose output from the two MCL approaches. In both cases the map for the algorithms was precomputed using the reflector-based pose for mapping and the measurements from the SICK S300. The mapping was carefully done in a static environment and the same data was used for creating both an occupancy map and an NDT map.

The test environment is a basement in Örebro University (referred as "basement" from now on) approximately 25m x 25m area, with 22 reflectors installed and calibrated for providing an accurate ground truth estimate. Fig. 3 illustrates the best case expected position accuracy of the setup. The figure is generated by a tool provided by the manufacturer and it shows that the positioning accuracy in our setup should be below 1cm throughout the trajectory. The test area and the trajectory used for the vehicle in trajectory following tests are illustrated in Fig. 2b. The speed of the vehicle while it followed the trajectory was 1m/s for the most parts.

B. *Offline accuracy test*

These tests are based on recorded datasets and the results are generated offline. The evaluation was done against the pose given by the vehicle's navigation system. As a measure of accuracy we use the Absolute Trajectory Error (ATE) [13][2]. Three different tests were conducted: 1) static-, 2) dynamic- and 3) sliptest. The static test was performed in an empty basement without disturbances. The dynamic test had additional obstacles inserted into the basement, and eight people that both disturbed the view of the vehicle and moved the obstacles in the basement during the experiment (see Fig. 4c). The obstacles were initially placed into the basement and were also present in the maps (see Fig. 4b). In sliptest the vehicle was driven in a static environment and small disturbances to the odometry were occasionally caused while driving by giving a "gentle push" to the vehicle. The trajectory lengths in the three tests were: 120m in static test, 180m in dynamic test and 300m in the slip test.

Fig. 5 shows the ATE plots for both grid-based MCL and NDT-MCL with different resolutions in the static case. The best case value for grid-MCL was 5.4cm while for NDT-MCL it was 1.4cm. In fact, the worst case value (for a

[2]ATE implementation from the Rawseed Project (http://www.rawseeds.org) was used

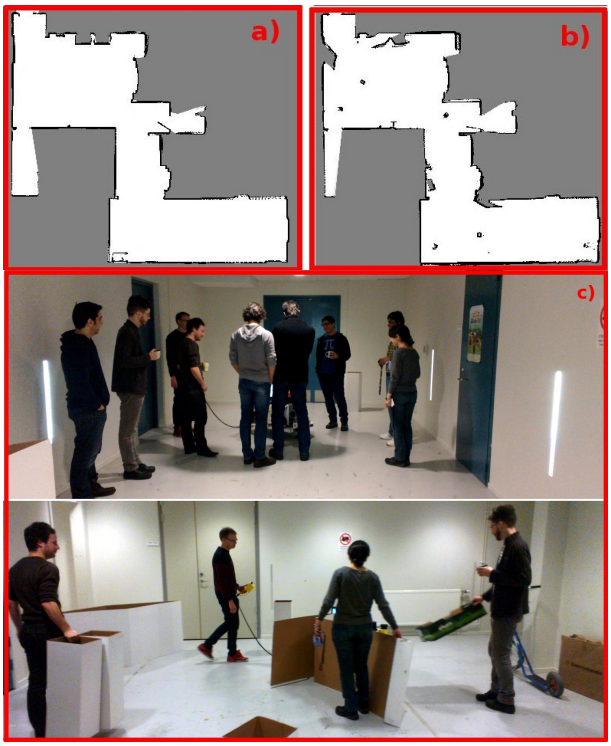

Fig. 4. Experimental setup. Occupancy maps of a) static trial and b) dynamic trial. c) shows snapshots from dynamic trial.

cell resolution up to 1.8m) for NDT-MCL is 2.8cm. The grid-MCL follows closely a rule of thumb that the accuracy corresponds to the resolution of the map, however, not better than the 5.4cm mentioned above even with small cell sizes. These findings are consistent with those reported in previous works on MCL [3], [9]. NDT-MCL, on the other hand, performs slightly worse with very high resolutions. This is due to two reasons: 1) with high resolution, due to small cell size, there is not sufficient statistics to represent the surfaces with normal distributions correctly and 2) our approximation of Eq. 4 accounts only for neighboring cells and thus when the cell size is small there is a risk of erroneous associations.

Fig. 6 illustrates the results of all three offline tests. The Grid-MCL and NDT-MCL results are illustrated in separate subplots with different scale for better visualization. The first remarkable observation is that NDT-MCL is hardly affected by the resolution in the static cases and provides almost constant performance over the full range of resolutions from 0.1m up to 1.8m.

Both static- and sliptest produce very similar performance. In the dynamic test there is a slight loss of performance for NDT-MCL. The best mean accuracy in this case for NDT-MCL was 2.4cm and for grid-MCL it was 5.3cm. The worst case accuracy for NDT-MCL in this test was 6.6cm, which was obtained at 0.15cm resolution in the dynamic test data set. The resolutions between 0.3m to 0.9m give less than 3cm mean ATE.

Finally, Fig. 7 compares the mean runtimes of the two approaches. The large variance in grid-MCL is an effect that the particular implementation does not update the observa-

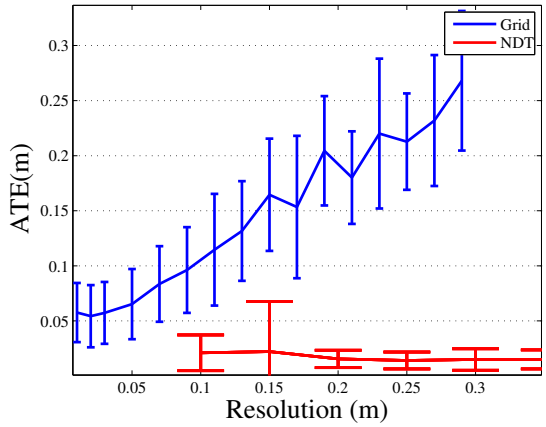

Fig. 5. Absolute Trajectory Error (ATE) comparison between grid-MCL and NDT-MCL in static environment. The curve represents the mean value and the bars the standard deviation.

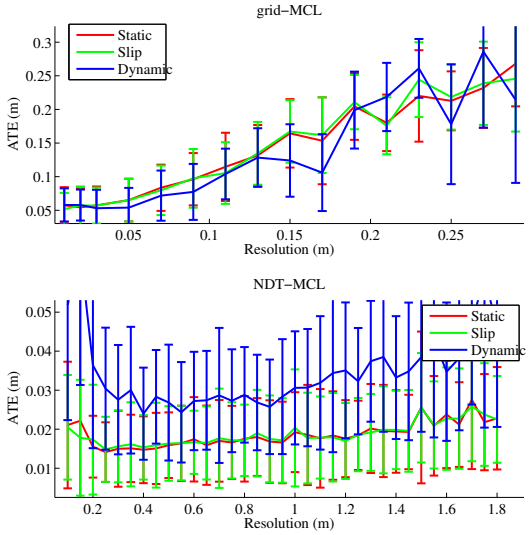

Fig. 6. Accuracy in three different scenarios using different resolutions. The static test was performed in a completely static environment, the Slip test includes disturbances to odometry while driving, and the dynamic test contained eight people around the robot causing changes to the environment by moving obstacles.

tion in every iteration and therefore resulting low runtimes in log files. Both approaches can run in real-time in our use-case. The NDT-MCL is clearly faster at lower resolutions. The resolution does not have (much) effecton the grid-MCL, since the likelihood field model is precomputed and the laser measurements are directly used for updates. Thus, the update of grid-MCL is always $O(NM)$, where N is the number of particles and M is the number of measurements in a scan, regardless of the resolution of the map.

C. Closed-loop tests

When evaluating a localization approach, one should not limit the analysis only to offline processing. A short term localization error or unsmoothness of the pose estimate

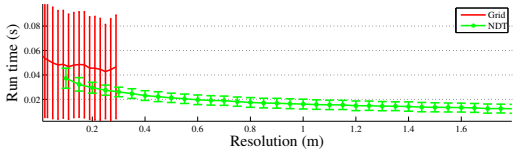

Fig. 7. Run time comparison between grid-based MCL and NDT-MCL.

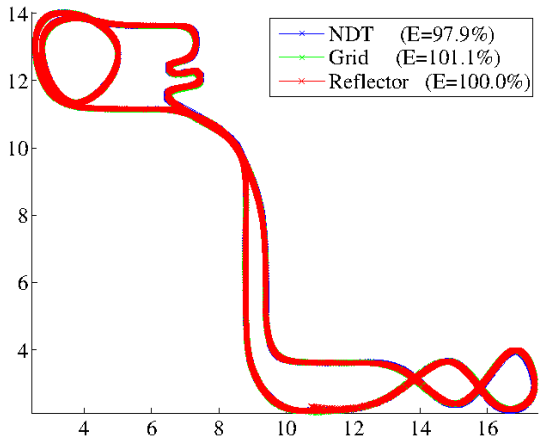

Fig. 8. Trajectories in online smoothness test. The value in the legend is the relative smoothness value (see Eq. 6) with respect to the reflector based approach.

may very well remain hidden in the mean value estimates, while in a real-world implementation the effects could be catastrophic. Moreover, the smoothness of the pose estimate will have an effect on the controller performance and thus on the overall performance of the system. In this section we analyze the localization approaches when used in a closed-loop with the vehicle control. The pose estimates of different approaches are fed to the controller, which in turn tries to follow a predefined trajectory. The controller, as mentioned in Sec. III-A, is a commercial AGV controller used by thousands of AGV's in production. The selected resolution for NDT-MCL was 0.5m and for grid-MCL 0.03m.

The ground-truth estimate was not available in the closed-loop tests. This is due to the vehicle controller design; once an external position source is fed into the system the reflector position gets disconnected. Therefore other metrics are used in this evaluation.

The first test measures the smoothness of the control. The vehicle was driven in a static environment during five laps on the trajectory illustrated in Fig. 8. The same test was repeated using grid-MCL, NDT-MCL and reflector based localization. The smoothness was measured as:

$$E = \sum_{t=1}^{n} \triangle\omega(t), \tag{6}$$

where $\triangle\omega(t)$ is change of turning angle between time steps $t-1$ and t. This is a measure of the amount of control actions that the controller does in order to execute the given trajectory. The results in Fig. 8 are given relative to the

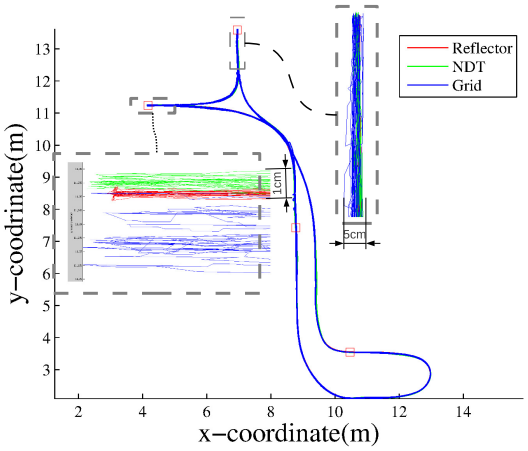

Fig. 9. Trajectories in repetition tests as returned by the algorithms. The stop points are illustrated as red squares. The areas with dashed rectangles are close-ups from the two different stop points and show that the grid-MCL reports the largest deviation in position in the test.

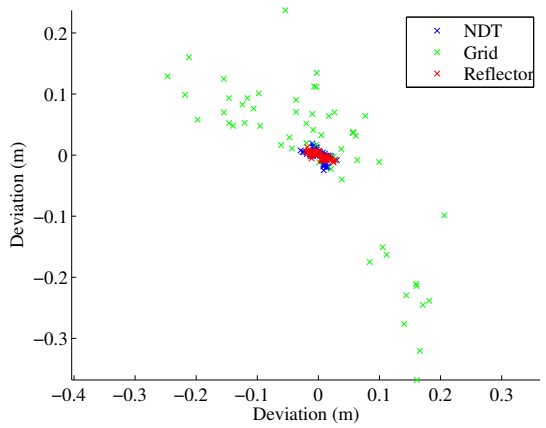

Fig. 10. Positioning accuracy in repeatability test.

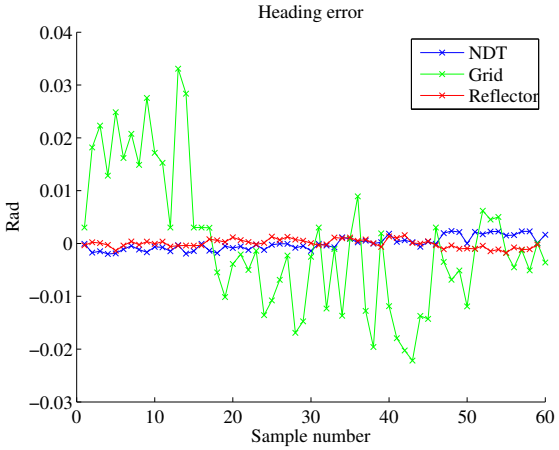

Fig. 11. Heading accuracy in repeatability test.

reflector based approach. For both grid-MCL and NDT-MCL the control visually looked smooth. According to Eq. 6 1.1% more control actions were needed for grid-MCL and 2.1% less for NDT-MCL, compared against reflector based positioning.

Repeatability, especially in loading operations, is crucial for an AGV system. An AGV has to be able to position itself within 3cm accuracy and with little angular error in order pick up pallets, for example. Loading points in industrial systems are at the end of a straight line, because typical AGV kinematics are non-holonomic and the loading point needs to be approached with less than 1deg error in heading. In order to simulate this, we use a trajectory illustrated in Fig. 9. The red rectangles in Fig. 9 are places where the vehicle stops. Also, the speed of the trajectory profile was set so that the vehicle approaches slowly the target points. The trajectory was a closed loop and we collected data from 15 loops, i.e, in total from 60 stop points.

In order to evaluate the accuracy of the AGV positioning at the loading locations, we use scan matching [12]. During the first loop, a reference scan and an associated pose were recorded for each stop point. In the following rounds the pose difference was computed by matching the newly acquired scan at each loading location to the recorded reference scan. Fig. 10 illustrates translational errors and Fig. 11 the heading errors. The results show that NDT-MCL performs nearly equally well with 1.4cm accuracy, compared to the reflector based positioning with 1.2cm accuracy. Grid-MCL, on the other hand performs substantially worse, giving a deviation of 13.6cm (see Table I) and almost one degree in heading. The most likely reason for the bad performance of Grid-MCL in this test is that it does not continuously provide a smooth pose estimate, as is visible in the close-up in Fig. 9. The test vehicle is non-holonomic and thus when a sudden jump in the pose estimate occurs while the vehicle approaches a stop point, the controller simply cannot make the necessary control actions to reach the correct pose and thus accumulates error. As can be seen in Fig. 1, it is quite clear that grid-MCL results in substantially worse repetition accuracy than NDT-MCL.

D. Large-scale test in an industrial environment

The final test demonstrates the scalability of NDT-MCL for larger, more realistic, environments. The test is based on data logged using an AGV in production use (see Fig. 12). The duration of the dataset was 550 minutes and the total length of the trajectory driven by the AGV was 7.24km. The dataset contains Velodyne HDL-32 sensor measurements, odometry and reflector based pose reported by the navigation system, which we use for comparison. The AGV's trajectories are visible in Fig. 15 and the visible area (for laser) was

TABLE I
MEAN HEADING ERRORS AND ABSOLUTE DEVIATION OF POSITION FOR REPETITION TESTS.

Method	Heading deviation (deg)	Translation deviation (m)
NDT	0.074	0.014
Grid	0.743	0.136
Reflector	0.045	0.012

Fig. 12. AGV in operation.

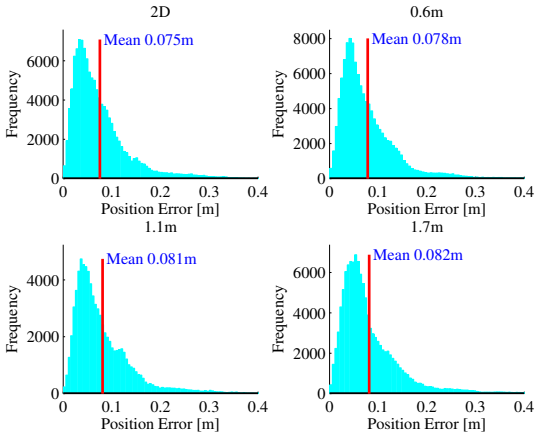

Fig. 13. Absolute trajectory errors for 10h localization set in industrial environment using different amounts of data from Velodyne HDL-32 sensor. The meter value indicates the amount of data included from horizontal level downwards.

approximately 150m x 150m. One 3D map was built for 3D tests and one 2D map for the 2D test. The maps were built using datasets collected on a different production day. The maps were created using the NDT-OM approach proposed in [10] with the pose information from the infrastructure based positioning system. Four tests runs were done with different height cut-off values h. The 2D tests use only one horizontal line from the scanner. The 3D tests used a segment of the scan such that the accepted z-values were within $z_s - h < z < z_s$, where z_s is the height of the sensor.

We noticed that the vehicle had a large bias in odometry while turning sharply. The amount of bias was such that it violated the motion model significantly and it took a long time for the filter to converge to a correct solution. Because of this, we added a map matching step to help the convergence. The map matching uses the recent measurement (with height cut-off) with an initial pose estimate from the NDT-MCL. The measurement is then registered against the map using the registration approach proposed in [12]. The result of the map matching is then used to replace one random particle from the distribution before calculating the observation likelihood. This was found to produce smooth trajectories, while maintaining the accuracy. The total number of particles used for the filter was 150 and the resolution of the map was 0.5m.

Fig. 13 summarizes the absolute trajectory errors obtained in the tests. The subplot titles in Fig.13 indicate the height cut-off value. Interestingly using 2D scans achieves the best accuracy. A likely reason for the larger deviation is that the dynamics of the environment increase close to the floor level. The 2D scans contain dynamics only from the other vehicles, while the 1.7m cutoff case captures also the continuously changing warehouse layout.

These differences in accuracy are not significant between the trials, however, the errors are larger than in our basement setup. One obvious reason for this is that the accuracy of the reflector based position in the factory was not as high as in our basement setup. In the basement we had 22 reflectors installed into a relatively small space. In a large scale facility, installation of reflectors is expensive and time consuming. Thus, when installing the system, the positioning accuracy is guaranteed only in the areas, where it is needed (i.e loading positions). This is also visible from the expected accuracy plot of the factory setup (Fig. 14). In regions other than loading areas the accuracy is in the range of 3-10cm. This clearly affects both the quality of the map used for localization, as well as the overall evaluation. The mean runtimes for both the map matching step and the NDT-MCL updates are shown in Table II.

TABLE II

MEAN RUNTIMES IN INDUSTRIAL SETUP.

	NDT-MCL	Map matching
2D	28ms	37ms
0.6m	48ms	56ms
1.1m	53ms	120ms
1.7m	87ms	200ms

Fig. 15 shows the trajectories of the AGV and close-ups of two often visited loading points. At these points there a slight offset from the ground truth trajectories is visible, however, the deviation of trajectories of individual methods is within 2-3cm, which is within the requirements for the use-case.

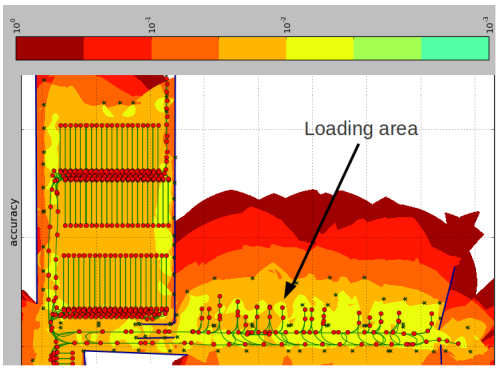

Fig. 14. Expected reflector-based position accuracy in factory setup.

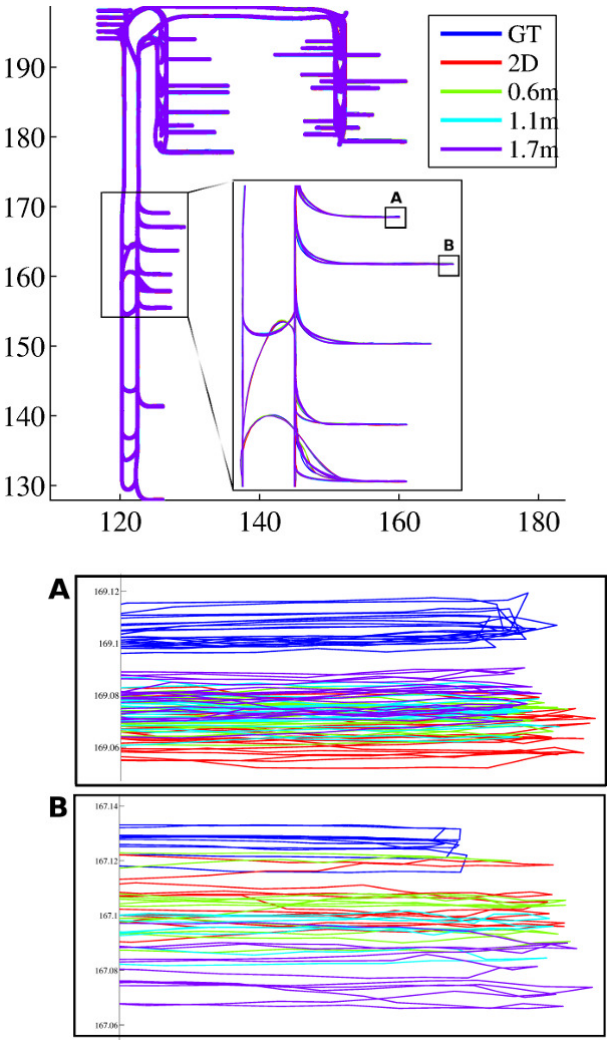

Fig. 15. Trajectories from the 10h localization test in an industrial environment. The top figure shows the overall trajectory and the other sub figures show close-ups from three selected loading points.

IV. Summary and Conclusion

In this paper we propose a Monte Carlo Localization approach, based on the Normal Distributions Transform representation (NDT-MCL). NDT-MCL was compared against a grid-MCL approach in several tests, both using pre-recorded datasets as well as in closed-loop control tests. In all tests NDT-MCL was found to provide superior performance over the grid-MCL and comparable to a commercial infrastructure based system. In addition the approach was successfully evaluated in a long-term test in a real world industrial sscenario.

Overall, the NDT representation was found to be very suitable for MCL. It is inherently a likelihood model and piecewise continuous. This results in an accurate and smooth posterior distribution that is clearly a desired property in localization.

Although the results obtained in this paper showed that NDT-MCL has potential to meet the industrial accuracy requirements, the tests in highly dynamic environments showed some loss of performance. The NDT-MCL implementation presented in this paper had no rejection policy for measurements originating from dynamic obstacles. Also, the present implementation of NDT-MCL did not consider global initialization and for this purpose an efficient sampling method has to be used [4].

References

[1] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2003)*, pages 2743–2748, 2003.

[2] James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. In *Radar, Sonar and Navigation, IEE Proceedings-*, volume 146, pages 2–7. IET, 1999.

[3] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.

[4] Dieter Fox. Adapting the sample size in particle filters through kld-sampling. *The international Journal of robotics research*, 22(12):985–1003, 2003.

[5] Kalevi Hyyppa. Method of navigating an automated guided vehicle, March 7 1989. US Patent 4,811,228.

[6] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007.

[7] Simon Maskell and Neil Gordon. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. In *Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE*, pages 2–1. IET, 2001.

[8] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 116 – 121, mar 1985.

[9] Jörg Röwekämper, Christoph Sprunk, Gian Diego Tipaldi, Cyrill Stachniss, Patrick Pfaff, and Wolfram Burgard. On the position accuracy of mobile robot localization based on particle filters combined with scan matching. In *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems October 7-12, 2012, Vilamoura, Algarve, Portugal*, 2012.

[10] Jari Saarinen, Henrik Andreasson, Todor Stoyanov, and Achim J. Lilienthal. Normal Distribution Transforms Occupancy Maps: Application to Large-Scale Online 3D Mapping. In *In Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2013)*, 2013. Available at `http://asrobo.hut.fi/~jari/papers/Saarinen_etal_NDT-OM_ICRA2013.pdf`.

[11] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer, 1990.

[12] Todor Stoyanov, Martin Magnusson, Henrik Andreasson, and Achim J Lilienthal. Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations. *The International Journal of Robotics Research*, 31(12):1377–1393, 2012.

[13] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[14] S. Thrun, W. Burgard, and D. Fox. Probabilistic robotics, 2006.

[15] Sebastian Thrun, Dieter Fox, Wolfram Burgard, et al. Monte carlo localization with mixture proposal distribution. In *Proceedings of the National Conference on Artificial Intelligence*, pages 859–865. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000.