

Localization in highly dynamic environments using dual-timescale NDT-MCL

Rafael Valencia, Jari Saarinen, Henrik Andreasson, Joan Vallvé, Juan Andrade-Cetto and Achim J. Lilienthal

Abstract—Industrial environments are rarely static and often their configuration is continuously changing due to the material transfer flow. This is a major challenge for infrastructure free localization systems. In this paper we address this challenge by introducing a localization approach that uses a dual-timescale approach. The proposed approach - Dual-Timescale Normal Distributions Transform Monte Carlo Localization (DT-NDT-MCL) - is a particle filter based localization method, which simultaneously keeps track of the pose using an apriori known static map and a short-term map. The short-term map is continuously updated and uses Normal Distributions Transform Occupancy maps to maintain the current state of the environment. A key novelty of this approach is that it does not have to select an entire timescale map but rather use the best timescale locally. The approach has real-time performance and is evaluated using three datasets with increasing levels of dynamics. We compare our approach against previously proposed NDT-MCL and commonly used SLAM algorithms and show that DT-NDT-MCL outperforms competing algorithms with regards to accuracy in all three test cases.

I. INTRODUCTION

Localization is an essential enabling component of mobile robotic systems. The importance of accurate positioning is crucial 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 3 cm, especially during loading operations. Moreover, as AGVs are typically non-holonomic platforms where side position errors are difficult to correct at destination points, sufficient accuracy has to be maintained throughout the trajectory. These requirements are especially hard when the operating environment is dynamic. A typical logistics application is depicted in Fig. 1, where autonomous vehicles are used for material transport. Here the environment dynamics consists of other vehicles and people. In addition, the configuration is continuously changing due to semi-static objects, that is, objects that change their position slowly, e.g. trolleys that are temporarily placed into the storage area. For this type of environments localization cannot rely solely on a static map. Current SLAM solutions are not robust and accurate enough in long-term operation to be accepted by the industry. Consequently, industry is still

This work has been supported by the European Commission under contract number FP7-ICT-600877 (SPENCER) and by the Spanish Ministry of Economy and Competitiveness under Project DPI-2011-27510 and the EU Project CargoAnts FP7-605598.

R. Valencia, J. Saarinen H. Andreasson and A. J. Lilienthal are with Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. {first name}.{last name}@oru.se.

J. Vallvé and J. Andrade-Cetto are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens Artigas 4-6, 08028 Barcelona, Spain. {jvallve, cetto}@iri.upc.edu.



Fig. 1. An example of the real-world application challenges for localization. The dynamics of the environment includes people, moving vehicles, and semi-static layout.

using positioning methods that rely on fixed infrastructure. This has drawbacks in terms of increased installation costs and the limited reconfigurability of the system, thus there is a demand for more flexible solutions.

We build on our previous work in [1]: Normal Distributions Transform Monte-Carlo Localization (NDT-MCL). It employs the Normal Distributions Transforms (NDT) representation for an MCL implementation. The NDT is a piecewise continuous representation, which represents the space as a set of normal distributions [2, 3]. 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. In [1] it was shown that NDT-MCL provides superior performance over the occupancy grid-MCL, where occupancy grids cannot represent the environment accurately, especially in presence of sensor noise. Moreover, NDT-MCL showed potential to meet the industrial requirements for both accuracy and repeatability. However, it was also shown that the accuracy of the algorithm decreased in a dynamic environments with semi-static objects.

In this paper we formulate an extension of NDT-MCL that is designed to provide an accurate and reliable pose estimate in the presence of substantial structural changes caused by semi-static obstacles, as illustrated in Fig. 1. Our approach to the problem is a dual-timescale update

formulation for the NDT-MCL. The first timescale uses the static map of the environment and the second timescale uses a short-term NDT Occupancy Map (NDT-OM) [4]–[6] that is continuously updated based on the expected pose returned by the algorithm.

We evaluate our approach with several test scenarios with increasing difficulty: from a stationary environment to one that undergoes substantial structural changes. The results show that the dual-timescale NDT-MCL provides equal performance as the previously proposed NDT-MCL in stationary environments and superior performance in dynamic environments with semi-static objects. We also show that our approach outperforms commonly used SLAM algorithms and that these algorithms do not provide sufficient accuracy for industrial requirements.

The main contribution in this paper lies in the proposed localization algorithm that tolerates substantial changes in the environment and yet provides superior position accuracy in real-time. Another key novelty is that our approach does not have to select an entire timescale map but rather use the best timescale locally.

The rest of this paper is organized as follows. Sec. 2 presents related work. Sec. 3 describes our extension to NDT-MCL. Sec. 4 presents the test setup and analyzes the obtained results. Lastly, Sec. 5 concludes the paper.

II. RELATED WORK

There are several contributions that deal with localization in dynamic environments. Some of them use static maps while considering dynamic objects as outliers, e.g. [7, 8]. Although it works well in some applications, it is not effective if the environment is changing.

An alternative to improve the localization is to use two maps to distinguish between static and dynamic objects in the environment. Wolf and Sukhatme [9] propose an approach that labels every cell into a dynamic map, which has been detected to change state. In our experiments, this approach would have rendered most of the space to be dynamic. Hähnel et. al [10] use an Expectation Maximization algorithm to classify the outliers from the complete dataset. The approach is not real-time and focuses on the extraction of dynamic objects.

Meyer-Delius et. al [11] proposed to use a combination of temporary maps and a static map for localization in a dynamic parking lot environment. Localization is done with a particle filter and the empty parking lot map is used as static map. Temporary maps are created using scan matching, which are then kept consistent by using a graph SLAM approach. The position of the robot is always computed against the static map, if the observation is explained by it well enough. If not, then the localization is done with a temporary map. This method is similar to ours, with an exception that our approach does not have to select an entire timescale map but rather use the best timescale locally. Moreover, we do not have a separate approach for keeping track of the short-term map, which can be a source of error as well.

Walcott-Bryant et. al [12] introduced a dynamic pose graph model, which extends the pose graph model by annotating the observations in the nodes. This representation allows to detect low-dynamic objects, such as a box that was moved, and to determine if certain nodes are not needed anymore because of the dynamics. While this approach is appealing in theory, the resulting accuracy reported in [12] is low: 3.4-13.3 cm in a fairly simple setup, with low dynamics. Moreover, the runtime of the algorithm depends on the number of nodes, which is always the case with pose-graph based approaches.

An alternative way to deal with the dynamics in the environment was introduced by Biber and Duckett [13]. They introduced a spatio-temporal map where the environment is represented at multiple timescales simultaneously. They showed that this approach can be used for long-term localization. The major disadvantage of this approach is the need to store and update large amount of data for the map. Our approach uses only two timescales, maintained as a compact NDT-OM representation [4, 5], which has lower memory requirements. Moreover, unlike [13], the approach introduced in this paper does not assume that a current observation belongs to one timescale or another. It always attempts to employ the static map when possible, but relies on the short-term map for the parts that are not explained by the static map.

A. Monte Carlo Localization

Monte Carlo Localization (MCL) [14] is among the most popular localization approaches in the robotics community. MCL is a probabilistic, map based localization approach that has been shown to be robust in real-world scenarios [14]–[16].

MCL estimates the posterior of the robot pose $p(x_t | u_t, z_t, m)$, given a map m , a sequence of controls $u_t = \{u^1, \dots, u^t\}$, and a sequence of observations $z_t = \{z^1, \dots, z^t\}$, i.e.,

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

with η a scaling factor, $p(z_t | x_t, m)$ the measurement model, $p(x_t | u_t, x_{t-1})$ the motion model, and $p(x_{t-1} | z_{t-1}, u_{t-1}, m)$ the posterior of the previous state.

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

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

where x_t^k is one pose sample and w_t^k is a weight associated to it. In this work we consider localization in the 2D-plane, so a pose $x_t^k \in \mathbb{R}^3$ is represented with (x, y) the position and the 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. We use the so called odometry model, i.e. 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 [17]. Since dx_{t-1}^t is affected by noise in the odometry measurement, a normally distributed noise vector $\sigma_t \in \mathbb{R}^3$, proportional to relative motion, is added for each particle. In this work we use 10% odometry error to simulate such a noise. The prediction is done by transforming each particle pose according to Eq. 3:

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

Next, in the update step, the measurement model is used to evaluate the likelihood of observations given a state. Finally, a resampling step is needed in order to avoid particle divergence [18, 19].

However, in prior contributions, MCL has not been reported to reach the precision requested by industrial applications [14]–[16]. Röwekämper et al. in [16] analyzed the accuracy of an MCL localization system with KLD-sampling and reported mean MCL errors of approx. 6 cm 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. Moreover, it is unclear how well ICP type of scan matching performs if the environment changes substantially.

III. DUAL-TIMESCALE NDT-MCL

A. NDT-MCL

Normal Distribution Transform Monte-Carlo Localization (NDT-MCL) was introduced in [1]. It employs NDT [2, 3] to represent both, the map and the measurements. NDT assumes 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 $p_i \in \mathbb{R}^3$. In this paper we assume that a measurement corresponds to a 2D laser scan. Thus, such a measurement is transformed to an NDT representation by accumulating the range points in a regular grid and then computing the mean and covariance estimates for each grid cell containing points. As a result, 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 and thus can directly serve as a likelihood model.

An NDT map is a regular grid containing the estimated normal distribution parameters of the mapped environment. We used the recursive implementation described in [4] to build an NDT map. The map is maintained in a grid representation for an efficient access to the Gaussian components, however, for sake of notation we can assume that the map is a set of normal distribution parameters $m = \{\mu_j, \Sigma_j\}_{j=1}^{N_m}$.

We now consider a particle pose x_t^k , represented as a rotation matrix R_k and translation t_k with respect to the global frame of reference. Now, the likelihood of a measurement \bar{z}_t , given the map m and the state x_t^k , can be given as the L_2 -likelihood [20],

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

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 measurement \bar{z}_t 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:

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

We use a basic resampling approach [19], which is triggered when the variance of the weights grows over a given threshold. To obtain the final pose estimate we use the maximum a posteriori estimate over the particles.

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

B. Dual-Timescale NDT-MCL

The original formulation of NDT-MCL uses a static map for measurement updates. This means that when the environment changes, the correspondence between the map and the observation is reduced. In severe cases this results in wrong associations between map and observation, which causes the pose distribution to diverge. On the other hand, in [1] it was shown that the maximum a posteriori estimate of NDT-MCL is very accurate in static environments. The accuracy also decreased minimally in the presence of dynamic objects (e.g. moving persons). This gives us a reason to assume that if the environment, within a certain time interval, is not completely altered, we can use the pose estimate of the NDT-MCL to build a consistent map. This “short-term map” describes the current state of the environment and is used in this paper to improve localization.

The localization process is divided into two parts: 1) localization with a known map and 2) mapping with a known pose. We use both maps in the measurement update step, however, unlike in [13], we do not assume that our current observation belongs to one timescale or another. Instead, we always use the static map if possible and the short-term map only for the parts that are not sufficiently explained by the static map. This provides the advantage to ground the pose estimates against the static map reference frame whenever possible. This gives us a continuous “loop-closing”, which keeps the short-term and the static map aligned.

An overview of the Dual-Timescale NDT-MCL is given in Algorithm 1.

C. Short-Term Mapping

We use the Normal Distribution Transform Occupancy Map (NDT-OM) representation [4]–[6] for the short-term map. NDT-OM is an NDT map, which also models the occupancy of the space. It has been shown to be robust against

Algorithm 1: Dual-Timescale NDT-MCL pseudocode

```
Prediction (Eq. 3)
Build a local NDT map from the observation
forall Particles  $p_i$  do
   $L_2 \leftarrow 0$ 
  Transform local NDT map according to  $p_i$ 
  forall NDTs in Local map,  $i = 1 \dots N$  do
     $l2_i \leftarrow L_2(\text{STATIC MAP}, \text{NDT}_i)$ 
    if  $l2_i < \xi$  then
       $l2_i \leftarrow L_2(\text{SHORT-TERM MAP}, \text{NDT}_i)p(c_i = 1)$ 
     $L_2 = L_2 + l2_i$ 
  endforall
Normalize
Resample
Compute current pose
if  $\text{trace}(\Sigma_{\text{position}}) < \gamma$  then
   $\text{Update short-term map}$ 
endif
```

dynamic changes [4] and provides a good performance in long-term operations [5].

We use the maximum a posteriori pose estimate directly obtained from the particle filter for updating the NDT-OM. However, we only update the map when the pose estimate is reliable enough. One measure of pose uncertainty is the determinant of the pose covariance. However, as noted in [21], the covariance might easily become ill defined, with one or more eigenvalues near 0. Instead, we chose to use the trace of the covariance of the position estimate. Thus, confidence in the pose estimate in this paper is based on thresholding the trace with an experimentally selected value (see Sec. IV-B).

For every timestep t the short-term NDT-OM is represented with a set of N_{m_t} Gaussian components:

$$m_t = \{\mu_j, \Sigma_j, p_j\}_{j=1}^{N_{m_t}}, \quad (6)$$

where μ_j and Σ_j are the current mean and covariance estimates and p_j is the occupancy probability, which represents the confidence of existence attached to the normal distribution parameters.

In [6] and [5] two parameters were introduced that affect the behavior of NDT-OM in dynamic environments. The first one is an occupancy clamping limit, which in effect limits the maximum and minimum confidence that a cell can have. The second parameter is the so called covariance adaptation weight, which sets the rate for the covariance recency weighting. In our tests, we found that these parameters have a minor effect on localization accuracy as long as the occupancy limit is set high enough (>100) and the covariance adaptation weight is also set to a large enough value (> 250).

D. Measurement update using two timescales

The measurement update process for dual-timescale NDT-MCL follows closely that of NDT-MCL, with the exception of the timescale selection. That is, the update step computes the weights for each particle as

$$w_t^k \propto w_{t-1}^k L_2^k(\bar{z}_t | x_t^k, m_s, m_t), \quad (7)$$

where $L_2^k(\bar{z}_t | x_t^k, m_s, m_t)$ is the observation likelihood between \bar{z}_t given both the static map m_s and the short-term map m_t .

To compute the observation likelihood the measurement is projected according to the predicted pose of the corresponding particle. After the projection, the observation is a set of N_{zt} normal distribution parameters $\bar{z}_t = \{\mu_i, \Sigma_i\}_{i=1}^{N_{zt}}$ transformed with respect to the global map. Then, for each observed NDT c_{zi} , we search for the closest normal distribution c_{mi} , from the static map and evaluate the L2-likelihood between them,

$$L_2(c_{zi}, c_{mi}) = \exp\left(\frac{-(\mu_{zi} - \mu_{mi})^T (P_{zi} + P_{mi})^{-1} (\mu_{zi} - \mu_{mi})}{2}\right), \quad (8)$$

where μ_{zi} and P_{zi} are the mean and the covariance of c_{zi} and μ_{mi} and P_{mi} are the mean and covariance of c_{mi} . If the obtained likelihood is above a given value, we choose the component to belong to the static timescale, otherwise we repeat the procedure with the short-term map.

When an observation is selected to belong to a short-term map, we also observe the occupancy value, $p(c_i = 1)$, of the short-term map cell c_i . Thus, for this case, our observation model is defined as follows,

$$\begin{aligned} p(c_{zi}, c_i = 1 | x_t^k, m_t) &= p(c_{zi} | x_t^k, m_t) p(c_i = 1 | x_t^k, m_t) \\ &\propto L_2(c_{zi}, c_{m_{ti}}) p(c_i = 1), \end{aligned}$$

where $c_{m_{ti}}$ is the normal distribution from the short-term map closest to c_{zi} . The occupancy describes the confidence of existence of an NDT within that cell. Therefore, if an NDT in the short-term map has low confidence, it should have a small effect in the measurement update.

Thus, the observation likelihood in Eq. 7 becomes

$$L_2^k(\bar{z}_t | x_t^k, m_s, m_t) = \sum_{i=1}^{N_{zt}} \begin{cases} L_2(c_{zi}, c_{m_{si}}) & , \text{ if } L_2(c_{zi}, c_{mi}) > \xi \\ L_2(c_{zi}, c_{m_{ti}}) p(c_i = 1) & , \text{ otherwise} \end{cases}, \quad (9)$$

where $c_{m_{si}}$ is the normal distribution from the static map closest to c_{zi} .

The value for ξ can be chosen adaptively, e.g., based on the mean observation likelihood. However, in our tests we found that a fixed value (0.4) is sufficient to provide good performance in all tested conditions (See IV-B).

IV. TESTS AND RESULTS

A. Test Setup

All evaluation data were collected using a commercial Automatically Guided Vehicle (AGV) system from Kollmorgen¹. This allows us, apart from collecting odometry and laser scans (SICK S300), to obtain pose estimates with a reflector based localization system [22], which are used as ground truth. The platform has the same kinematics as normal forklift based AGV's but is smaller in size, see

¹<http://www.kollmorgen.com>

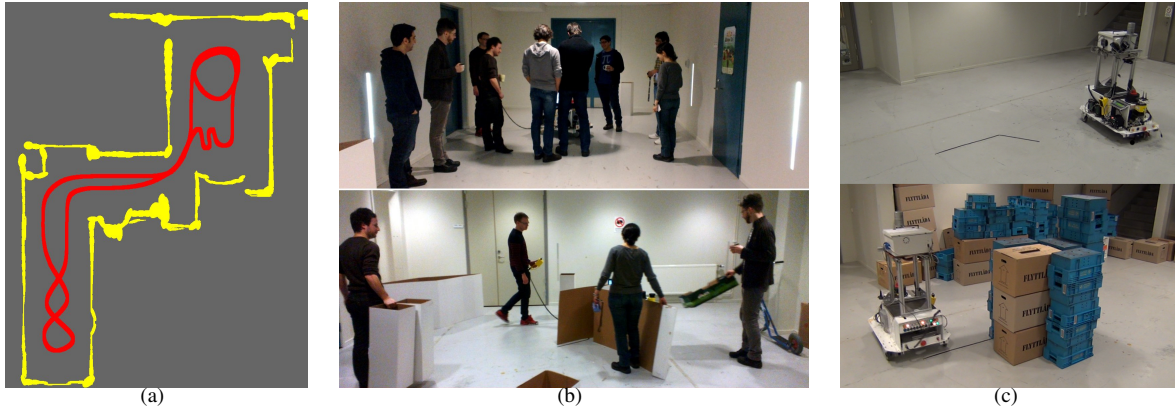


Fig. 2. (a) The trajectory used in all test scenarios in the evaluation (red). The ellipses (yellow) encodes the NDT representation for occupied cells, which in this figure was obtained using the static dataset. (b) Two snapshots taken during the dynamic scenario. (c) Two snapshots taken during the boxes scenario, in the beginning (top) and in the middle (bottom).³

Fig. 2c. All different test scenarios used the same predefined trajectory, which was executed by the AGV system at 1m/s for most parts, see Fig. 2a.

The output of each localization method was compared to the ground truth reflector based pose estimate. The metric used is the Absolute Trajectory Error (ATE) [23]². All datasets were recorded in a basement at Örebro University (referred to as “basement” hereafter) of a size approx. 25×25 meters. In the area we mounted 22 pre-mapped reflectors in order to assure accurate ground truth pose information with position error < 1 cm and orientation errors < 0.005 rad.

Three test scenarios were selected for the evaluation: 1) static, 2) dynamic and 3) boxes. The static test was performed without any additional obstacles or moving people. The dynamic test contains additional obstacles and people that moved around in the vicinity of the vehicle. The obstacles were moved around throughout the experiments (see Figure 2b). During the boxes scenario, four persons spread boxes around the basement making the layout to gradually change substantially (see Figure 2c). At the end of the experiment the boxes were gradually removed, such that the final state of the basement was identical to the initial configuration³. The trajectory lengths in the three scenarios were: 298.8 m (static), 298.9 m (dynamic) and 1135.9 m (boxes).

In all experiments we used 150 particles for the filter. A single core implementation of DT-NDT-MCL was found to run at a rate of 30-40HZ with a 0.4m map resolution when processed with an Intel Core i7 CPU at 1.60GHz.

B. Parameter Analysis and Selection

DT-NDT-MCL introduces two parameters: The value for the $L2$ threshold, ξ , in Eq. 9, and the value for the pose covariance trace, γ , in Algorithm 1. The parameter ξ determines how much we trust the static map. Setting ξ high means that the observation and the static map components must be a

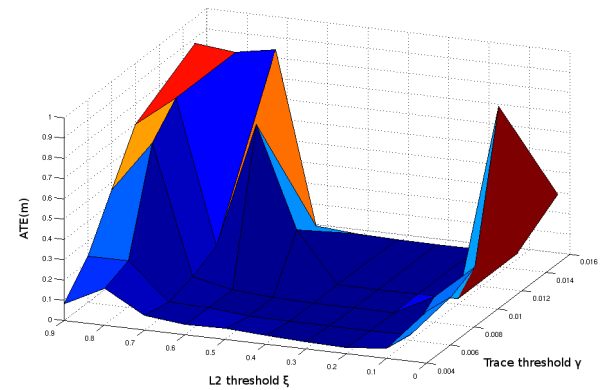


Fig. 3. The effect of DT-NDT-MCL parameters to localization accuracy. The x-axis is the $L2$ threshold ξ in Eq. 9, the y-axis is the position accuracy threshold γ in Algorithm 1, and the z-axis is the ATE. The large flat area indicates that the accuracy stays low for a large range of parameters values.

nearly perfect match. Setting ξ low allows components to be matched even if they are far from each other. The parameter γ is used to prevent map updates when the pose is uncertain. Too low values of γ require the pose to be very accurate and may prevent the short-term map to be updated. Too high values may cause the map to be updated with uncertain pose information and cause inconsistency in the short-term map.

Fig. 3 illustrates the effect of these parameters in a localization test. The test was conducted by running the boxes dataset with several parameter combinations. Fig. 3 shows that the algorithm is not very sensitive to γ . Any value between 0.005 and 0.015 provides similar performance. For parameter ξ , values between 0.2 to 0.4 gave the best performance according to Fig. 3. For all tests we selected $\xi = 0.4$ and $\gamma = 0.01$.

C. Accuracy Test for Different Map Resolutions

This test is conducted by running each dataset using a wide variety of resolutions, i.e. varying the size of the NDT-OM cells. This evaluation serves two purposes: 1) to verify that the previous approach NDT-MCL is insufficient in the

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

³A video of the experiment is available at: <http://www.youtube.com/watch?v=O7q1v960ZgE>

TABLE I
MINIMUM MEAN ERRORS IN ACCURACY TESTS.

Dataset	NDT-MCL	DT-NDT-MCL
Static	1.6cm @ resolution: 0.2m	1.5cm @ resolution: 0.3m
Dynamic	1.7cm @ resolution: 0.4m	1.8cm @ resolution: 0.6m
Boxes	11.2cm @ resolution: 1.6m	2.2cm @ resolution: 0.5m

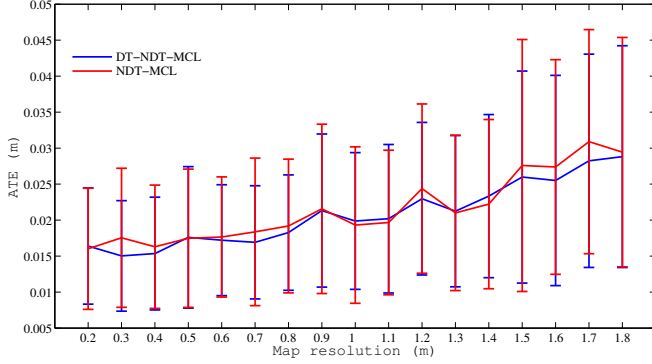


Fig. 4. Resolution tests with the static dataset

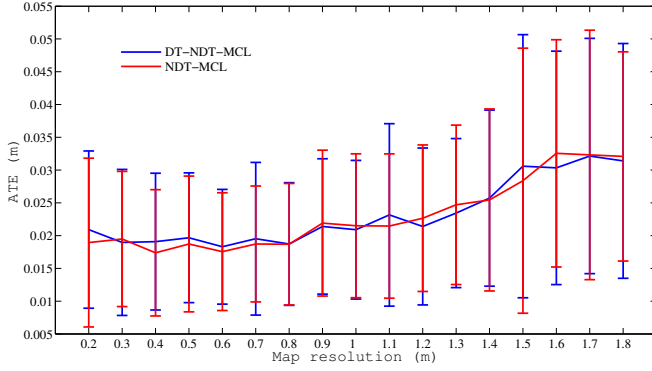


Fig. 5. Resolution tests with the dynamic dataset

case where we have semi-static objects, and 2) to check the sensitivity of the method with respect to map resolution.

Both, Fig. 4 and Fig. 5 show that the performance of DT-NDT-MCL and NDT-MCL is comparable in a static environment and in a crowded environment over a wide range of resolutions.

Fig. 6 shows the results of the boxes dataset. With this dataset it becomes clear that using only a static map is not sufficient, when the layout changes substantially. DT-NDT-MCL on the other hand maintains its performance even in these conditions. This is further confirmed in Table I, which summarizes the best mean error values in the experiments.

D. Comparisons with SLAM algorithms

DT-NDT-MCL utilizes two maps, a static and a short-term map. Another approach would be to dispose off the static map and completely work with a continuously updated short-term map, i.e. use SLAM. This section evaluates the performance of the three test scenarios for different SLAM

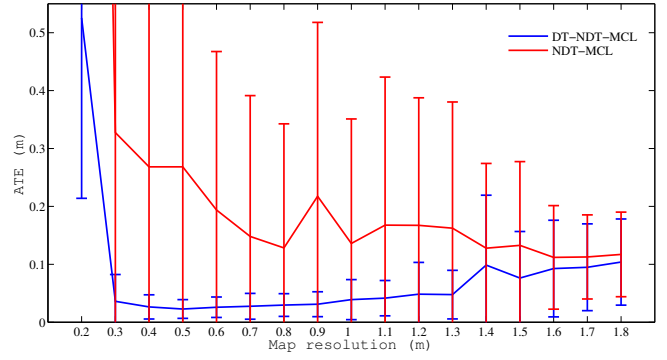


Fig. 6. Resolution tests with the boxes dataset

methods: gmapping [24]⁴, mapping and tracking based on the NDT-2-NDT registration [20] and a ROS implementation of Pose SLAM [25]⁵.

The first approach, gmapping [24] is a particle filter based SLAM approach that is widely used and known to be robust. The second approach, the NDT-2-NDT mapping and tracking approach uses a maximum likelihood principle to build a map incrementally by matching the current observation to the map [6, 26]. In addition we implemented a false matching detection based on the odometry noise model to prevent false matches, due to the heavily crowded environments, to be added to the map. The third approach, Pose SLAM [25] is a variant of SLAM where only the robot path is estimated and where landmarks are only used to produce relative constraints between robot poses. In Pose SLAM a probabilistic estimate of the robot pose history is maintained as a sparse graph using an information filter. Graph links indicate relative geometric constraints between robot poses, and the density of the graph is rigorously controlled using information measures. To enforce sparseness in Pose SLAM, only the non redundant poses and the highly informative links are added to the graph. A new pose is considered redundant when it is too close to another pose already in the trajectory and not much information is gained by linking this new pose to the map. However, if the new pose allows to establish an informative link, both the link and the pose are added to the map. The Pose SLAM implementation used for this paper uses the odometric data from the platform and the Iterative Closest Point (ICP) [27] registration method in the front-end.

The comparison is depicted in Figure 7 and summarized in Table II. The boxes scenario clearly showed to be the most difficult one. Gmapping handled the boxes dataset comparably well, which could be explained due to the multiple hypotheses entailed in the filter. The NDT-2-NDT mapping and tracking works well for the static and crowded environments, but clearly fails in the boxes experiment. Pose SLAM behaved comparably well during the three scenarios and for the boxes dataset it notably obtained the best performance compared to the rest of the mapping approaches,

⁴We used the `ros-fuerte-slam-gmapping` package.

⁵wiki.ros.org/iri/poseslam

TABLE II
MEAN ERROR VALUES IN SLAM COMPARISON.

Method	Static	Dynamic	Boxes
Gmapping	4.88cm	8.36cm	11.7cm
NDT2NDT	1.66cm	3.27cm	66.1cm
Pose SLAM	6.59cm	8.30cm	9.59cm
DT-NDT-MCL	1.56cm	1.58cm	2.35cm

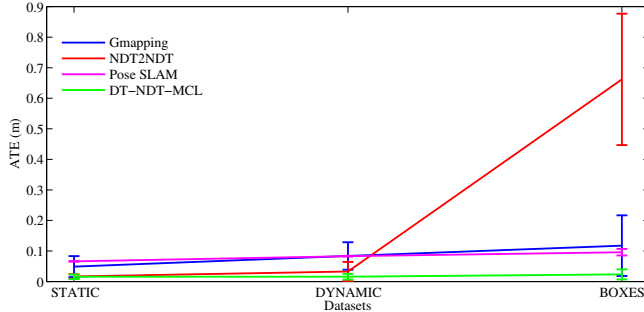


Fig. 7. Comparisons with SLAM algorithms

being able to sustain such a cluttered environment given its large path length. In all tests, DT-NDT-MCL provides the best performance, being constant despite the large change of the conditions. For all other methods the error grows substantially when compared to the static test. DT-NDT-MCL, however, provides an excellent 2.35cm mean ATE even with the boxes dataset.

V. CONCLUSIONS

In this paper we introduced a localization algorithm that tolerates substantial changes in the environment. The proposed DT-NDT-MCL algorithm uses two maps to achieve this; a static map and a short-term map, which is continuously updated. We showed that the algorithm maintains accuracy in extremely dynamic environments where current state-of-the-art methods fail. We also analyzed the parameters used by the algorithm and showed that the algorithm is not particularly sensitive to parameter selection.

In future work the algorithm will be tested in scenarios where also long-term effects play a role (such as when a part belonging to a static map is removed completely). Such events also require adaptation of the static map, and possibly introduce a need for more than two times cales.

REFERENCES

- [1] J. Saarinen, H. Andreasson, T. Stoyanov, and A. Lilienthal, "Normal Distribution Transform Monte-Carlo Localization (NDT-MCL)," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2013, pp. 382–389.
- [2] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2003, pp. 2743–2748.
- [3] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.
- [4] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal, "Normal Distribution Transforms Occupancy Maps: Application to Large-Scale Online 3D Mapping," in *Proc. of IEEE Int. Conf. Robotics and Autom.*, 2013, pp. 2233–2238.

- [5] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, "3D normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1627–1644, 2013.
- [6] J. Saarinen, T. Stoyanov, H. Andreasson, and A. J. Lilienthal, "Fast 3D Mapping in Highly Dynamic Environments using Normal Distributions Transform Occupancy Maps," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2013, pp. 4694–4701.
- [7] O. Aycard, P. Laroche, and F. Charpillet, "Mobile robot localization in dynamic environments using places recognition," in *Proc. of IEEE Int. Conf. Robotics and Autom.*, 1998, pp. 3135–3140.
- [8] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.
- [9] D. Wolf and G. Sukhatme, "Mobile robot simultaneous localization and mapping in dynamic environments," *Autonomous Robots*, vol. 19, no. 1, pp. 53–65, 2005.
- [10] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Proc. of IEEE Int. Conf. Robotics and Autom.*, 2003, pp. 1557–1563.
- [11] D. Meyer-Delius, J. Hess, G. Grisetti, and W. Burgard, "Temporary maps for robust localization in semi-static environments," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2010, pp. 5750 – 5755.
- [12] A. Walcott, "Long-term robot mapping in dynamic environments," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.
- [13] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *Proc. of Robotics: Science and Systems*, 2005, pp. 17–24.
- [14] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Montecarlo localization for mobile robots," in *Proc. of IEEE Int. Conf. Robotics and Autom.*, 1999, pp. 1322–1328.
- [15] S. Thrun, D. Fox, W. Burgard *et al.*, "Montecarlo localization with mixture proposal distribution," in *Proc. National Conf. on Artificial Intelligence*, 2000, pp. 859–865.
- [16] J. Röwekämper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard, "On the Position Accuracy of Mobile Robot Localization based on Particle Filters Combined with Scan Matching," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2012, pp. 3158–3164.
- [17] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous robot vehicles*. Springer, 1990, pp. 167–193.
- [18] S. Maskell and N. Gordon, "A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking," in *Target Tracking: Algorithms and Applications*, IEE. IET, 2001, pp. 1–2.
- [19] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," in *Radar, Sonar and Navigation, IEE Proceedings-*, vol. 146, no. 1. IET, 1999, pp. 2–7.
- [20] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1377–1393, 2012.
- [21] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *Proc. of IEEE Int. Conf. Robotics and Autom.*, 2005, pp. 661–666.
- [22] K. Hyypä, "Method of navigating an automated guided vehicle," Mar. 7 1989, uS Patent 4,811,228.
- [23] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2012, pp. 573–580.
- [24] G. Grisetti, G. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, "Fast and accurate SLAM with Rao-Blackwellized particle filters," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 30–38, 2007.
- [25] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact Pose SLAM," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 78–93, 2010.
- [26] T. Stoyanov, J. Saarinen, H. Andreasson, and A. J. Lilienthal, "Normal Distributions Transform Occupancy Map Fusion: Simultaneous Mapping and Tracking in Large Scale Dynamic Environments," in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2013, pp. 4702–4708.
- [27] P. Besl and N. McKay, "A method for registration of 3D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.