Grant agreement no: FP7-600877

SPENCER:

Social situation-aware perception and action for cognitive robots

Project start: April 1, 2013

Duration: 3 years

DELIVERABLE 2.7

SLAM and localization with socially annotated mapping

Due date: month 34 (January 2016)

Lead contractor organization: ORU

Dissemination Level: PUBLIC

Contents

Abstract

This deliverable describes the work on task T2.4 done after what was reported in deliverable D2.6. We have mainly pursued two directions of work during 2015.

The first direction of work was research on socially annotated mapping, with focus on human motion patterns. We have developed a method to model velocity patterns in the environment using Gaussian mixture models and worked on extending it with usage of Markov Chains. The Gaussian mixture model models the dominant motion patterns (direction and speed) while the associated Markov Chain represents the temporal relation between directions of motion ("exit velocity, conditioned on entry velocity").

We have also focused on improving localization robustness and optimizing mapping and localization modules. A major extension has been to employ a 3D sensor for mapping and localization purposes, in order to overcome problems with extreme occlusion of the 2D sensors at leg height during busy periods in the airport.

1 Introduction

Task T2.4 should provide the robot with precise localization and maps for navigation and planning (T5.2, T5.5), object tracking from a mobile observer (T2.1, T2.3), and social relation analysis (T4.3). Another activity within this task is socially annotated mapping. In addition to providing precise localization in a metric map, the technologies developed within this task also need to provide a map with social annotations. To fulfill the requirements, we need to provide the following modules.

Mapping This module consists of the *environment representations* that allow us to model the static and dynamic parts of the environment, as well as social activities therein.

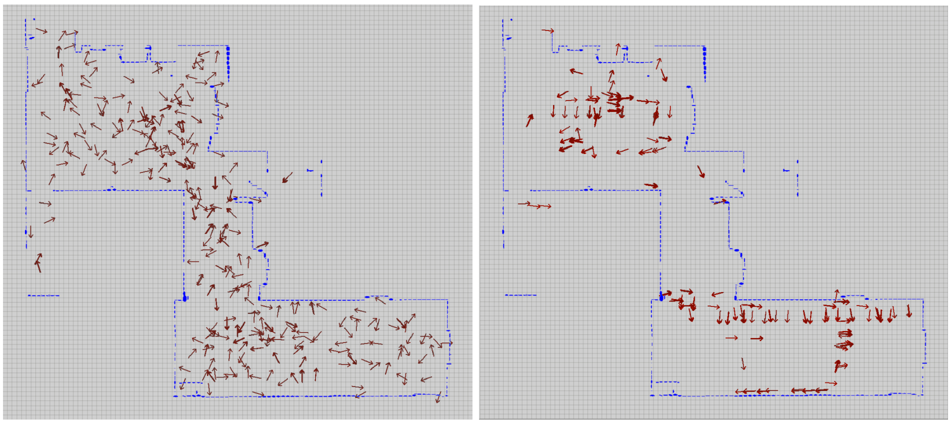
SLAM This module *builds* a geometrically consistent map using the sensor data from the robot platform. (The output map of the SLAM module is represented using the data structures of the mapping module.)

Localization This module provides accurate continuous localization in the presence of different levels of dynamic changes in the environment.

2 Localization

2.1 3D-NDT-MCL Tuning and Testing

Deliverable D2.6 described our prototype localization system using 2D-NDT-MCL for localizing the robot. While performing extensive localization tests both at Schiphol and ALU-FR, we concluded that it is not possible to provide accurate localization in all situations, using only the 2D laser sensors that were originally mounted on the robot platform. It was decided to mount an additional, 3D, laser scanner (Velodyne VLP-16) on the robot's shoulder. Within the one week of testing at Schiphol, during the integration week in December 2015, 2D localization only failed on one occasion. This can



(a) Uniform initialization

(b) GMM initialization

Figure 1: Comparing the initial particle distribution using a uniform prior belief (a) vs an informed prior based on an NDT map (b).

still be considered to be a good result, but as an effort to ensure zero localization failures, the main mode of operation has been to use 3D-NDT mapping and localization from then on.

To lower the memory consumption of the localization module we have integrated a sub-map approach [4]. At all times we are using only a section of the whole map for localization. We achieve this by splitting the map into a set of tiles that are dynamically switched while the robot is moving. In this way we have decreased the memory consumption by one order of magnitude, compared to loading the map of the entire work area.

To reduce computation time have combined two methods. First we discard all observations below a certain height. In this way, we omit those parts of the scan data that are most likely to contain people and other dynamic objects (in an airport, we can safely assume planar motion). In this way we not only decrease the computation load but also remove the most noisy parts of the scan. In addition we also subsample the set of distributions in the NDT representation of the scan data, thus matching a sparse representation of the current scan to the map.

2.2 Bootstrapping MCL

MCL is one of the most popular map-based localization methods for mobile robots, and it has been shown to be robust in real-world scenarios. However, when starting the robot system, or when recovering from a localization failure, MCL needs to be initialized with a prior belief distribution of the robot's pose. This is typically done either by manually providing an initial estimate of the robot's pose, or by initializing the localization system with a uniform (uninformed) distribution over the whole map. In order to achieve accurate localization from the very start of deployment, it is necessary that the localization algorithm converges very quickly to the true pose, which is not the case when using a uniform prior. To over come this problem, we have developed an algorithm to bootstrap MCL with a map-based informed prior.

We have published a method for constructing an informed prior for MCL in the 2015 ECMR conference [1]. The main contribution of this paper is a novel algorithm for constructing a prior for MCL from the robot's current sensor readings, exploiting the NDT map representation. We address the problem of initialization and re-initialization of NDT-MCL for cases where no external knowledge about the robot pose is available. However, the proposed initialization method can be used together with any implementation of MCL. We describe a method of quickly building a Gaussian mixture model (GMM) representing the prior belief distribution of possible robot poses, given a range scan and a map. The initial set of particles is then sampled from this GMM. The published paper is included in the appendix of this deliverable.

Figure 1 illustrates a prior particle distribution using our proposed method, compared to a uniform prior with the same number of particles. After conducting a series of tests in both static and heavily dynamic environments, we conclude that our informed prior enables much faster and reliable global localization, compared to an uninformed prior. In the best case, with 1000 particles, the uniform prior converged to the correct pose in 56% of the cases in the static environment, and 45% in the dynamic environment. In comparison, the informed (GMM) prior could converge correctly in over 80% in the static environment even when using only 100 particles.

2.3 Registration benchmark

In addition to the work in priors for MCL described in the previous section, we also include the final version of the paper describing our registration benchmark results [3] (that was also mentioned in D2.6) in the appendix of this deliverable.

3 SLAM

The core of our work on constructing geometric maps for navigation has already been described in deliverable D2.6. In the period after D2.6, work in the geometric mapping module has consisted of experimentation with the new 3D setup in the target environment (Schiphol) and similar environments at ALU-FR. We have also worked on optimization of memory use and computation time of the 3D-NDT-MCL module.

4 Mapping

Our final module for representing geometric maps for navigation uses NDT-OM, as has been described in deliverable D2.6. However, as discussed above, mapping and localization is internally performed using 3D data. The map and positioning data that is communicated to the rest of the SPENCER system is converted to 2D planar motion.

Our final socially annotated mapping module extends our previous work on learning and representing motion patterns (CTMap [2] and T-CTMap, as described in D2.6).

We have since then developed a novel method for learning motion patterns in dynamic environments. The major challenge was to build a probabilistic model that will be able to grasp *continuous* heterogeneous quantities. The data that we have to model consists of two components: direction (a *circular* quantity) and speed (a *linear* quantity). To model them jointly we have employed Circular-Linear Gaussian Mixture Model (CL-GMM). By Circular-Linear we mean that support one of the marginal distributions (the linear one) is $\mathbb{R}+$ while the other one is $[0, 2\pi)$.

The model is built independently for each location in a grid. To each grid cell we associate a CL-GMM which represents, in a compact way, the dominant directions and velocities of objects in the neighborhood of the considered location. In this way we build a probabilistic model representing the motion model within the environment.

The modes of CL-GMM can be perceived as states representing the direction of motion in a time interval (assuming that we have observations at discrete time intervals). In order to exploit this observation, we have added a Markov Chain representing the likelihood of transition among the states. In this way we can model a turning pattern where agents are switching between the modes.

5 Conclusions

This document reports on the final mapping and localization module of Task T2.4, including socially annotated mapping.

To fulfill the requirements of this task we have structured it in three modules: Mapping, SLAM, and Localization. The Mapping module (Section 4) is in charge of the environment modeling. Our system employs NDT-OM for modeling static parts of the environment and CL-GMM to model the statistics of moving objects, such as flows of people.

Our SLAM module builds a geometrically consistent NDT-OM by tracking the vehicle pose using a frame-to-model registration approach that iteratively fuses the sensor data into the NDT-OM map. After having observed some localization failures in cases of extreme occlusion of the original 2D laser scanners, we have shifted to 3D mapping and localization using a 3D laser scanner at shoulder height, in order to guarantee correct localization also in extremely busy periods at the airport.

Finally, our localization module (Section 2) makes use of NDT-MCL to accurately localize the robot in the NDT-OM map. Using a map-based informed prior distribution for initializing MCL, we can quickly perform global localization when needed, compared to a standard uninformed prior.

References

[1] Tomasz Piotr Kucner, Martin Magnusson, and Achim J. Lilienthal. Where am i?: An ndt-based prior for mcl. In *Proceedings of the European Conference on Mobile Robots (ECMR)*, September 2015.

[2] Tomasz Piotr Kucner, Jari Saarinen, Martin Magnusson, and Achim J. Lilienthal. Conditional transition maps: Learning motion patterns in dynamic environments. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[3] Martin Magnusson, Narunas Vaskevicius, Todor Stoyanov, Kaustubh Pathak, and Andreas Birk. Beyond points: Evaluating recent 3d scan-matching algorithms. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3631–3637, 2015.

[4] Todor Stoyanov, Jari Saarinen, Henrik Andreasson, and Achim J Lilienthal. Normal Distributions Transform Occupancy Map Fusion: Simultaneous Mapping and Tracking in Large Scale Dynamic Environments. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4702–4708, 2013.

Beyond Points: Evaluating Recent 3D Scan-Matching Algorithms

Martin Magnusson,[1] Narunas Vaskevicius,[2] Todor Stoyanov,[1] Kaustubh Pathak,[2] and Andreas Birk[2]

***Abstract*— Given that 3D scan matching is such a central part of the perception pipeline for robots, thorough and large-scale investigations of scan matching performance are still surprisingly few. A crucial part of the scientific method is to perform experiments that can be replicated by other researchers in order to compare different results. In light of this fact, this paper presents a thorough comparison of 3D scan registration algorithms using a recently published benchmark protocol which makes use of a publicly available challenging data set that covers a wide range of environments. In particular, we evaluate two types of recent 3D registration algorithms – one local and one global. Both approaches take local surface structure into account, rather than matching individual points. After well over 100 000 individual tests, we conclude that algorithms using the normal distributions transform (NDT) provides accurate results compared to a modern implementation of the iterative closest point (ICP) method, when faced with scan data that has little overlap and weak geometric structure. We also demonstrate that the minimally uncertain maximum consensus (MUMC) algorithm provides accurate results in structured environments without needing an initial guess, and that it provides useful measures to detect whether it has succeeded or not. We also propose two amendments to the experimental protocol, in order to provide more valuable results in future implementations.**

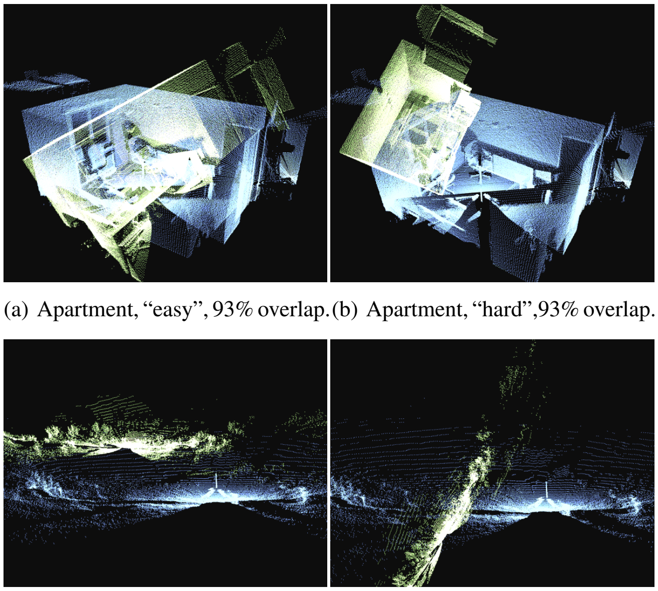
I. Introduction

Three-dimensional registration, or scan matching, is a crucial component of several robotics applications, such as mapping, object detection, manipulation, etc. Scan registration can be formulated as the problem of finding the relative transformation betwen two 3D point clouds that best aligns them.

A common problem with research papers presenting novel scan matching algorithms is that results are computed over a small number of scans from an application-specific environment. The need for standardised datasets for benchmarking registration algorithms has been recognised by the community and several recent works have been proposed [14, 10, 13]. However, in order to obtain the full benefits of such benchmarking efforts, it is critical that a sufficient number of registration algorithms are evaluated in a systematic manner, which is often not the case.

In this work we evaluate several recent registration algorithms on the challenging benchmarking dataset proposed by Pomerleau et al. [20]. We use this benchmark on two types of scan matching algorithms: local registration using NDT [11, 24] as well as the MUMC [15] algorithm for global matching. The purpose of this evaluation is two-fold: first, it aids the establishment of a scientific approach to comparing registration algorithms; and second, by applying a benchmark dataset and protocol designed originally for ICP-based algorithms to NDT and MUMC, we can identify the general validity of the proposed benchmark.

The two types of methods evaluated in this paper have different characteristics: MUMC decouples rotation and translation determination and does a global exhaustive search for the best alignment, while NDT registration uses local hill-climbing from an initial pose estimate. What is common for our algorithms is that they all take into account local surface structure around each point, and do not match individual points, in contrast to common ICP variants.


(a) Apartment, “easy”, 93% overlap. (b) Apartment, “hard”,93% overlap.

(c) Plain, “easy”, 37% overlap. (d) Plain, “hard”, 37% overlap.

Fig. 1: Examples from the data sets.

The main contribution of the present paper is a thorough evaluation of these recent registration algorithms on a scale that has not previously been attempted, using 210 scan pairs from different types of environments, over 100 000 registrations in total. We show that both NDT and MUMC achieve more robust registration than ICP indoors, and that NDT performs well also outdoors. In addition, we show that the Distribution-to-Distribution variant of NDT (D2D-NDT [23]) is significantly faster than the others, though slightly less robust than Point-to-Distribution variant (P2D-NDT[11]). Furthermore, the present evaluation is the first one using the proposed protocol for non-ICP methods.

A second major finding of this paper is that the proposed benchmark [20] has significant shortcomings. First, the selection of unique scan pairs from the datasets is small, thus limiting the applicability for global scan matching approaches like MUMC. Second, the initial offsets provided are often unrealistic, while the amount of scan overlap is generally low. While this does indeed make the dataset more challenging, it limits its discriminative power in practically more interesting cases.

In addition to these main contributions, this article also presents several recent advances for NDT registration and makes available a new open-source implementation of the P2D-NDT algorithm.

II. Related work

Recently, much effort has been devoted to the benchmarking of 3D scan-registration and 3D SLAM algorithms. Different benchmarking approaches differ in the types of scenarios, the sensor

[1] Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. firstname.lastname@oru.se

[2] Deptartment of EECS, Jacobs University, Bremen, Germany. f.lastname@jacobs-university.de

This work was funded in part by the EU FP7 projects SPENCER (ICT-2011-600877) and ROBLOG (ICT-270350), and the Swedish KK foundation under contract number 20110214 (ALLO).

used, and the kinds of ground-truth information available. In this discussion, we restrict ourselves to 3D mapping benchmarks – as opposed to the more common 2D benchmarks such as RADISH [7] and RAWSEEDS [4].

A good collection of 3D datasets is provided at [13]. The majority have been collected using a high-resolution, long-range, and large field-of-view Riegl VZ-400 scanner. In some cases, ground truth computed by manual registration using markers is provided. Some datasets have additional information like co-calibrated thermal and color data, and odometry. One of these datasets was used in Pathak et al. [14] to benchmark the performance of MUMC [15] and ICP. Most datasets in [13] are from outdoor urban scenarios. No protocol is defined to systematically evaluate new algorithms.

In Wulf et al. [26], a 1.2 km path in an outdoor urban (mostly flat) scene was captured in 924 scans of the RTS/ScanDriveDuo. A 2D ground-truth map was obtained from the land-registry office, and Monte Carlo localization (MCL) was used to compare the results of four 6D-SLAM strategies against this 2D reference, and some manual intervention was applied for quality control. The obvious limitation of this benchmark is the lack of full 3D ground-truth and providing only one type of environment.

Some authors have performed detailed studies of the "valleys of convergence" for registration algorithms [12, 10, 6], though typically only for a few scan pairs.

A tiltable SICK LMS 200 was employed in [10] to collect scans in an underground mining scenario. The two algorithms compared were ICP and NDT. No ground-truth was available, so the algorithms are compared with respect to accumulation of errors and the valley of convergence.

The common drawback of all the above mentioned benchmarks is the availability of only a single kind of environment and of typically only a few scan pairs. In contrast, the ETH benchmark used here [19, 20] has a range of environments, 6 × 35 scan pairs (chosen to have a uniform range of overlaps between 30% and 99%), with 3 × 64 pose offsets, sampled from a 6-DOF normal distribution. Hence, (for local registration methods at least) we have 40,320 tests in total for the benchmark, per algorithm.

III. Registration algorithms

Let $\mathcal{F}$ and $\mathcal{M}$ be two partially overlapping point clouds, taken from nearby poses. Additionally, we define $\mathcal{F}$ to be a *fixed* (or reference) scan, while $\mathcal{M}$ is a *moving* (or reading) point set. The registration task estimates the parameters Θ of a transformation function T, such that $T(\mathcal{M}, \Theta)$ is consistently aligned with $\mathcal{F}$.

A. P2D-NDT

The Normal Distributions Transform was originally developed in the context of 2D laser scan registration [2]. The central idea is to represent the observed range points as a set of Gaussian probability distributions. Assuming that a set of n point samples $\mathcal{P} = \{\boldsymbol{p}_i = (x_i, y_i, z_i)\}$ has been drawn from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, the maximum-likelihood estimates of the covariance and mean can be obtained from the observations:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{i=n}\boldsymbol{p}_i, \quad M = [\boldsymbol{p}_1 - \boldsymbol{\mu} \dots \boldsymbol{p}_n - \boldsymbol{\mu}], \quad \Sigma = \frac{1}{n-1}MM^T$$

The probability density function estimated in this manner might or might not be a good representation of the sampled points, depending on the extent to which the Gaussian assumption on $\mathcal{P}$ holds. At a sufficiently small scale, a normal distribution can be considered a good estimate of local surface shape, in that it can represent planar and linear patches. Thus, the basic principle of the NDT is to represent space using a *set* of Gaussian probability distributions.

The point-to-distribution (P2D) variant of NDT for 3D registration [9] maximises the likelihood of points from one scan, given the NDT model created from the reference $M_{NDT}(\mathcal{F})$. The likelihood that a point $\boldsymbol{x}$ is generated from $M_{NDT}(\mathcal{F})$ is then:

$$p(\boldsymbol{x}|M_{NDT}(\mathcal{F})) = \sum_{i=1}^{n_{\mathcal{F}}} w_i \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu_i}, \Sigma_i), \tag{1}$$

where $n_{\mathcal{F}}$ is the number of Gaussian components of the 3D-NDT model of point cloud $\mathcal{F}$. The weight of each Gaussian component w_i determines the influence of that component in the model set on the likelihood of a single point from the moving scan. Magnusson et al. [11] propose to use trilinear interpolation in order to take into account neighbouring Gaussians. The implementation used in this work, however, only considers the closest Gaussian to each point, thus setting w_i to zero for all other Gaussians. P2D-NDT minimises an approximation of the negative log likelihood function of $p(T(\mathcal{M}, \Theta)|M_{NDT}(\mathcal{F}))$, over the space of transformation parameters Θ. After re-organisation of terms, the registration problem is posed as minimising the objective function

$$f_{p2d}(\Theta) = \sum_{j=1}^{|\mathcal{M}|} -d_1 \exp \frac{-d_2}{2} \hat{\boldsymbol{m}}_{\boldsymbol{j}}{}^T \Sigma_m^{-1} \hat{\boldsymbol{m}}_{\boldsymbol{j}}, \tag{2}$$

where d_1 and d_2 are positive regularizing factors (values are calculated based on the current NDT model resolution as described in [8, 3]), j iterates over all points $\boldsymbol{m_j}$ in the moving scan $\mathcal{M}$, $(\boldsymbol{\mu_m}, \Sigma_m)$ are the parameters of the corresponding closest normal distribution in $M_{NDT}(\mathcal{F})$, and $\hat{\boldsymbol{m}}_{\boldsymbol{j}} = T(\boldsymbol{m_j}, \Theta) - \boldsymbol{\mu_m}$. The objective f_{p2d} is doubly differentiable with analytic expressions for the gradient and Hessian. Once $\mathcal{M}_{NDT}$ has been constructed, 3D registration can be performed by minimizing the objective function (2) using a numerical optimization technique, such as Newton's method.

The P2D-NDT implementation used here also employs a regularization step to avoid near singular Hessian matrices in the optimisation step. The procedure is the same as for the D2D-NDT algorithm, and is further described in Section III-B.

Important Parameters: In this work, the P2D-NDT implementation is a recent re-implementation as part of the `perception_oru` suite[1]. Several parameters govern the performance of the algorithm:

- **Discretization levels:** the algorithm performs registration on models reconstructed at different spatial resolutions. Typically, it is desirable to first register at a coarse resolution (e.g. 2 m cells), followed by finer registration steps (e.g., 1 m and 0.5 m). Note that the solver may go from finer to coarser grids, and back again, in a similar fashion as is common in multigrid methods for solving linear systems.
- **Maximum number of iterations:** controls the number of optimization iterations allowed at each resolution.
- **Subsampling grid size:** in order to avoid bias from uneven point distribution and to speed up computations, the moving scan $\mathcal{M}$ is subsampled using a regular grid of a given resolution. This resolution is another important parameter which affects overall performance.

The parameter selection is summarized in Table I.

[1] http://wiki.ros.org/perception_oru

B. D2D-NDT

The Distribution-to-Distribution (D2D) variant of the NDT registration algorithm, proposed by Stoyanov et al. [23], is an extension of P2D-NDT which operates solely on NDT models. The algorithm minimizes the sum of L_2 distances between pairs of Gaussian distributions in two NDT models. Formally, the transformation between two point sets $\mathcal{M}$ and $\mathcal{F}$ is found by minimizing:

$$f(\Theta) = \sum_{i=1,j=i}^{n_{\mathcal{M}},n_{\mathcal{F}}} -d_1 \exp\left(-\frac{d_2}{2}\boldsymbol{\mu_{ij}}^T (R^T C_i R + C_j)^{-1}\boldsymbol{\mu_{ij}}\right) \tag{3}$$

over the transformation parameters Θ, where: $n_{\mathcal{M}}$ and $n_{\mathcal{F}}$ are the number of Gaussian components in the NDT models of $\mathcal{M}$ and $\mathcal{F}$; R and $\boldsymbol{t}$ are the rotation and translation components of Θ; $\boldsymbol{\mu}_i, C_i$ are the mean and covariance of each Gaussian component; $\boldsymbol{\mu_{ij}} = R\boldsymbol{\mu_i} + \boldsymbol{t} - \boldsymbol{\mu_j}$ is the transformed mean vector distance; and d_1, d_2 are regularization factors (fixed values of $d_1 = 1$ and $d_2 = 0.05$ were used). The optimization over Θ can be done efficiently using Newton optimization with analytically computed derivatives.

Two modifications of the D2D-NDT algorithm, compared to the previously published version [23], have been included in this work. The prior version only considered the sum of pairwise closest Gaussian components when forming the objective function in Eq. (3), while the version tested here can be configured to use a neighbourhood of close components. The second modification is the addition of a regularization step to the computation of the Hessian matrix, prior to the computation of a Newton step. We perform an eigen-decomposition of $H = Q\Lambda Q^{-1}$ and check if the smallest eigenvalue λ_{min} is close to 0. If that is the case, we compute a regularized Hessian matrix $H_r = Q(\Lambda + diag(\lambda_r))Q^{-1}$, where $\lambda_r = 10^{-3}\lambda_{max} - \lambda_{min}$. This procedure ensures that the Hessian matrix is not excessively biased in one particular search direction and helps avoid local minima in the objective function.

For D2D-NDT, we have the additional parameter of the neighbourhood size; i.e., the number of Gaussian distributions used in the evaluation of the objective function. While our previous implementation [23] used size 0 (i.e., only the closest distribution) we currently use size 1 (i.e., the 8 neighbours in the closest layer). The other parameters (grid sizes and termination criteria) are selected in the same way as for P2D-NDT.

A similar idea, also performing registration with an objective function based on Gaussians, is used by the Generalized ICP method [21]. However, Generalized ICP assumes locally planar patches around each point, then calculates the normal direction to the local surface and uses it to bias the orientation of the covariance matrix, as opposed to P2D-NDT and D2D-NDT, which estimate the Gaussian parameters using points within a local neighborhood.

C. Plane Matching (MUMC)

The "Minimally Uncertain Maximum Consensus" (MUMC) algorithm [15] is a global alternative to the local methods like ICP and NDT. This approach consists of a pre-processing step in which plane patches are extracted from the 3D scans [25].

Especially for scenes containing man-made structures with large planar surfaces, this leads to large data-compression: the memory required for the extracted "plane-cloud" can be as small as 2.5% of the original point-cloud [16]. Each plane patch is represented by the unit normal $\hat{\mathbf{n}}$ and the distance d from the origin, along with the polygonal boundary. Using a noise model of the 3D sensor model, a 4×4 covariance matrix of the plane parameters is also computed [17] for each patch.

TABLE I: Summary of parameter selections.

	Step	Description
P2D-NDT	Data filtering of moving	grid sampling, cell size 40 cm
	Data filtering of fixed	use full point cloud
	Grid resolution	1 m, 2 m, 1 m, 0.5 m
	Termination criteria	5 iterations, $\Delta\Theta < 10^{-3}$
D2D-NDT	Data filtering (both)	use full point clouds
	Neighbor layers	1
	Grid resolution	1 m, 2 m, 1 m, 0.5 m
	Covariance scaling	$d_1 = 1, d_2 = 0.05$
	Termination criteria	5 iterations, $\Delta\Theta < 10^{-3}$
MUMC	Sensor noise model	Gaussian, $\sigma = 7$mm
	Size-similarity threshold	8
	Diversity constraint	on/off

The MUMC registration method then works directly on the two "plane clouds" corresponding to the two scans to be registered. The registration is based on finding the set of patch correspondences in the two scans which leads to the most geometrically consistent 3D transformation – as measured by the determinant of the estimated covariance matrix of the transform. This covariance is a function of the aforementioned plane-parameter covariance matrices, which in turn are a function of the sensor range noise model. Hence, a cascade of the uncertainties is maintained. The search space for correspondences can be reduced by a set of consistency tests. Some examples of these tests are: threshold for the allowable variation in the size of a given patch between scans, or patch color-histogram consistency [18], when color is available, e.g. in RGB-D scans.

MUMC also exploits the fact that for planes, the determination of rotation is decoupled from translation. Thanks to this property, combined with the vast data reduction in the number of patches compared to the number of points, and due to the pruning of the correspondence search-space by many consistency tests, MUMC can afford to do a *global, exhaustive* search for the most consistent set of patch correspondences. Hence, one of its advantages is that, unlike local methods like ICP and NDT, it does not necessarily need an initial guess for the transform. In fact, it has been shown to be able to register scans taken far away from each other and with considerable occlusion, without odometry [14].

Although the original paper [15] lists 8 parameters to be chosen, in the latest version of the algorithm only 2 parameters are selected explicitly. The others are estimated automatically based on the sensor noise-level. Since the same sensor was used for all scans in this paper, the same parameters were used for all the datasets.

- **Sensor Noise Model**: We assume Gaussian sensor noise in range measurements, with standard deviation $\sigma = 7$ mm.
- **Size-Similarity Threshold**: The determinant of the inverse of the 4×4 plane-parameter covariance matrix is proportional to N^4, where N is the number of points on the patch. The size-similarity threshold $\bar{L}_{\text{det}}$ [15] dictates the maximum allowable change in N in two potentially corresponding patches from the two scans. We set $\bar{L}_{\text{det}} = 8$, which allows the ratio of the number of points in two potentially corresponding patches to change by a factor of as much as $\exp(8/4) = 7.4$. This is a very permissive value.

In addition to the above numerical parameters, MUMC can be run in two modes, as explained below.

The Plane Diversity Constraint (DC): For MUMC to compute translation reliably, it needs to find plane correspondences in all directions. When only two or less translation components can be found reliably (e.g., the corridors in the ETH dataset desribed below) the uncertainty is automatically detected by MUMC because a certain matrix is effectively numerically rank-deficient [15, Eq. (25)].

In this case, there are two options for MUMC:

- **DC-OFF**: This option signals that we want to keep the rotation estimate and the reliably found translation components. For the translation component in the unreliable direction, we could either set it to zero or, if available, we could use the odometry component in that direction. In both cases, the error in this direction will be large and this will adversely affect the translation accuracy statistics of the algorithm.
- **DC-ON**: In this case, MUMC automatically declares the registration as having failed rather than venturing to fill-in unavailable translation components with heuristics or odometry. This result will then not be considered for computing the accuracy statistics.

D. ICP

The iterative closest point (ICP) algorithm was first introduced in 1991 [5] and is still widely used for registration of 3D point clouds. The two seminal papers on ICP were written by Besl and McKay [1] and Chen and Medioni [5]. To summarize the algorithm concisely: ICP iteratively refines the relative pose of two overlapping scans by minimizing the sum of squared distances between corresponding points in the two scans. Corresponding point pairs are identified either by Euclidean point-to-point distance [1] or by a point-to-plane metric [5], which measures the distance from a point in one scan and the closest tangent plane in the other. Since its conception, a large number of variants have been developed, differing in, e.g., how points are selected and how to select point-to-point correspondences. However, the main structure of the algorithm remains. The point-to-plane variant has been shown to be more accurate in many cases, and Pomerleau et al. [20] show that it also performs better for the benchmark used here. As specified in the experimental protocol, we compare our algorithms to the baseline implementation of the well-established point-to-plane ICP variant.

The parameter selection for ICP is the same as in [20].

IV. Data and protocol

We have used the same six environments from the "Challenging Laser Registration" data sets [19] as in Pomerleau et al. [20]. One of the main advantages of these data sets is that the ground-truth poses of all point clouds have been tracked with millimeter precision using a total station. These six data sets cover both indoor and outdoor environments, cluttered and open, some with large planes and some with more variable surfaces.

Apartment: An apartment with five rooms. This data set has denser scans than the others: 365 k points per scan, compared to 100 k–200 k for the others. (See Fig. 1a–1b.)

Stairs: A staircase transitioning from indoor to outdoor.

ETH: Large hallway with pillars and arches. These scans have little constraints along the direction of the hallway, and also features repetitive structures in the form of the pillars.

Gazebo (winter): A public park with a gazebo.

Wood (summer): Dense vegetation around a small path.

Plain: Open field atop a mountain in the Alps. Includes very little geometric structure. (See Fig. 1c–1d.)

The data sets come with sets of initial pose offsets, to be used for assessing the algorithms' robustness to poor initial pose estimates. These pose offsets are indeed quite challenging, and a large part of them are much worse than what can be expected to be encountered by a mobile robot. The offsets categorized as "easy", "medium", and "hard", and are generated from zero-mean normal distributions, where the standard deviation is larger for the more difficult categories. The "easy" poses have a standard deviation of 0.1 m and 10°, while the "hard" poses have a standard deviation of 1.0 m and 45°. Although the translation offsets are not so severe, the rotation offsets are very large. Some of the "hard" poses have an initial rotation offset of more than 90°. This is very challenging for any local registration algorithm, which may more likely converge to a solution that is rotated 180° than the correct orientation. Even some of the "easy" poses are offset more than 30°, which is not particularly easy – especially not for scan pairs with little overlap from unstructured environments. Conversely, some of the "hard" poses may be quite close to ground truth.

A. Suggested amendments to the protocol

For future work, we suggest the following amendments to the protocol [20].

1) Fixed-magnitude pose offsets: The pose offsets should have fixed magnitudes (as in Magnusson et al. [11]) rather than being sampled from normal distributions with increasing variance. Given the scale of the scans, reasonable magnitudes might be 0.5 m and 10° for "easy" poses, 2.5 m and 20° for "medium" poses, and 5 m and 45° for "hard" poses.

2) More unique scan pairs: The large number of initial pose estimates are relevant only to the local methods. For global methods which do not necessarily need initial guesses, it is more important to have more unique pairs and their ground-truth transforms.

V. Results

The results of our evaluations are summarised in Figs. 2 and 3. The rotation and translation errors are plotted separately, but precise registrations should have small errors both in translation and rotation. Due to space constraints, we were not able to include plots about all pertinent aspects of the benchmark. The complete outcome[2] for the NDT-based methods, as well as additional tables and plots of the results[3] are available online.

For the local registration methods (ICP and the NDT-based methods), we provide separate plots for the scan pairs with "easy" pose perturbations only (Fig. 3c) and the large overlaps only (Fig. 3b), in addition to plots for the complete data sets (Fig. 3a).

Because MUMC does not make use of an initial estimate, the only relevant sub-category is the amount of overlap. Another consequence is that the statistics are computed only from the 35 unique scan pairs from each data set, as opposed to 6720 combinations of scan pairs and pose offsets for the local registration methods. In addition to the magnitude of the final pose error, the entries for MUMC also specifies the percentage of scans that were used to compute the errors. The remaining percentage of scans could not be registered, but were also successfully declared as having failed.

A. Accuracy

The accuracy of the final solution (after registration) is measured as described in Pomerleau et al. [20, Eqs. 1–3]. The translation error is the Euclidean norm of the difference between the ground-truth and the output translation vectors. The rotation error is defined as the geodesic distance from the rotation matrix that brings the moving point cloud from the output orientation to that of the ground-truth pose. Error statistics are discussed in terms of quantiles where Q50 is the median and Q95 is the 95th percentile.

[2] http://projects.asl.ethz.ch/datasets/doku.php?id=laserregistration:evaluations:home

[3] http://aass.oru.se/Agora/Benchmarks/

The unstructured data sets, Wood and Plain, are quite difficult for all of the algorithms. Because they do not contain enough geometric structure, they are challenging for MUMC in particular. Looking at the percentages of matches estimated by MUMC to be successful, we can see that it detects that it is not able to perform matching in most cases. For Wood, MUMC detects a lack of features in 86% of the 35 scan pairs with DC-ON (63% with DC-OFF), and only tries to match the remaining 5 pairs, which means 14% of the pairs from this dataset. Because of the lack of structure, the translation error is above 50 cm also for the remaining pairs.

For Wood scans with large overlap (over 75%) ICP is accurate (less than 10 cm error) up to Q50, while P2D-NDT is equally accurate up to Q75. P2D-NDT is the only algorithm that is able to register any outdoor scans with small overlap (30%–50%), and is significantly better than the others for the Plain dataset (accurate to Q50 for "easy" poses). D2D-NDT performs similarly to ICP for the unstructured data sets.

Gazebo is an outdoor data set, but it also contains a prominent built structure. In fact, this data set proved slightly easier than the indoor data sets for the NDT methods, judging by Fig. 3a. We believe that the reason for this is in the combination of large surfaces that the moving scans can "slide on" and good constraints from the gazebo itself, which are not planar enough to be useful for MUMC.

For the structured data sets (including Gazebo), P2D-NDT finds accurate solutions up to at least Q75 as long as the pose offset is "easy", and about Q45–Q55 for all poses. This is in contrast to ICP, which finds accurate solutions up to Q50 for the "easy" poses, and Q30–Q40 for the all poses. Another finding is that D2D-NDT has better overall accuracy for the complete datasets, but when looking at the "easy" poses only, P2D-NDT is better. In other words, D2D-NDT is less sensitive to poor initial pose estimates but sometimes less precise when a good pose estimate is available.

MUMC performs much better for Apartment and Stairs than the other datasets, because it can exploit the structure. Stairs in particular works well. Counting the pairs that the algorithm flags as successfully matched, MUMC DC-ON correctly registers 100% of the pairs with large overlap from that dataset, and 95% of all pairs. ETH, on the other hand, is difficult because it only has a strong planar component along one direction. MUMC finds the correct *orientation* in most cases, or detects that it cannot provide a certain result (see Fig. 2b).

Some of the "easy" poses are also quite difficult for the local methods. Practically no registrations converge to an acceptable solution for all these cases. The exception is P2D-NDT on the Gazebo dataset, with accurate solutions (within 5 cm and 2° from ground truth) even at the 95th percentile for the "easy" poses.

Comparing Figs. 3a and 3b, it can be seen that ICP is more sensitive to small overlaps than the NDT-based methods. Considering only scan pairs with large overlap, ICP's results are more similar to NDT's than when considering the "easy" poses only. NDT often succeeds up to at least Q25 also small overlaps (30%–50%), although scan pairs with small overlap from the unstructured datasets are very challenging, especially for D2D-NDT. As long as there is diverse enough planar structure in the environment, which is the case in Apartment and Stairs, MUMC is quite robust to small overlap ratios, which can be seen by the fact that the corresponding curves in Figs. 3d and 3e are similar.

B. Further discussion of MUMC results

The most important aspect of the MUMC results is that since MUMC is a global method, *no initial guess* was provided to it. The only categorisation pertinent to MUMC is the overlap level: large (easy to register), medium, and small (difficult to register). As noted above, the local algorithms (NDT and ICP) do require an initial guess that is close enough to the true solution.

In addition to the converged 3D transform, MUMC also returns a flag for whether the result can be considered successful (based on the number of correspondences found) and a covariance matrix of the transform.

The percentage of matches labelled by MUMC as successful is noted for each dataset in the plots. The stricter MUMC DC-ON will also consider a registration unsuccessful if certain components of translation cannot be found, as explained in Sec. III-C. Therefore, the success percentage of MUMC DC-ON for each dataset is less or equal to that of MUMC DC-OFF.

The covariance matrix of the transform clearly shows the dominant uncertain directions (the eigenvectors corresponding to high eigenvalues). It is clear from Fig. 2b that the rotation determination of both MUMC DC-ON and DC-OFF is very accurate for the cases that are flagged as successful – in particular for the ETH dataset.

At this point it should be noted that the Hessian of the NDT score function (2), (3) can also be used to construct a covariance estimate of the pose after registration, in order to flag unsuccessful matches [8, 22]. A detailed analysis of that is not within the scope of the present paper, but is a topic of ongoing work.

C. Timing

The execution times when running the benchmark are summarized in Figure 3. The reported times include all pre-processing (extracting plane patches for MUMC, building NDT representations, constructing kd-trees for ICP, etc.) but exclude the time required for loading the scan files from disk.

As aptly noted by Pomerleau et al. [20], it is difficult to make precise comparisons of execution times. Many uncertainties affect execution time: not only the hardware used, but also the compiler, the skill of the programmers, etc. The execution times provided here can only give a coarse estimate of the speed.

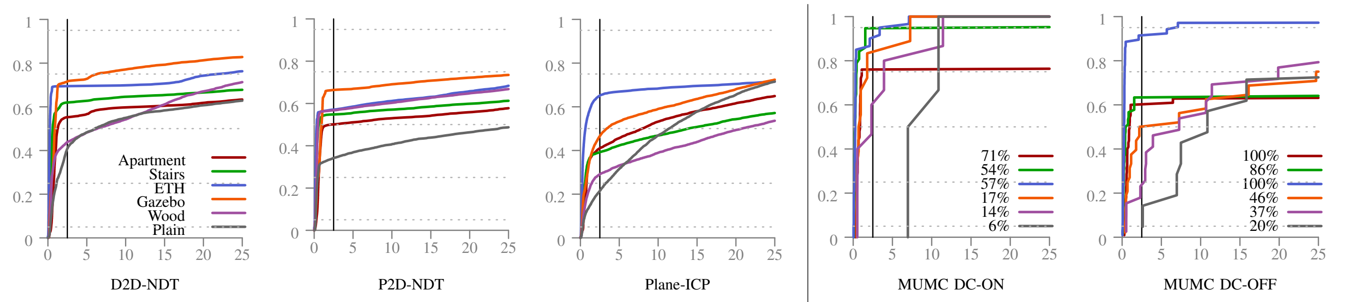
The execution times were measured on different computers. NDT was running on a quad-core 3.50 GHz Intel Core i7 CPU. MUMC was running on an 3.40 GHz Intel Core i7. Please note that these are slightly faster computers than the 2.2 GHz Core i7 used for ICP [20].

Although the implementations can make use of multiple cores, for these tests they were all running in a single thread. Typically four batches were being run in parallel (each on a separate CPU core) to reduce the overall processing time.

Implementation details: The difference between the execution speed of ICP and P2D-NDT is smaller than what has been reported in our earlier publications. One of the reasons is that the ICP implementation used here (libpointmatcher[4]) uses more efficient components (e.g., the kd-tree implementation, where most of the execution time for ICP is spent, uses libnabo instead of libann).Another reason is that the old P2D-NDT implementation[5] runs significantly faster than this version, from `perception_oru`. Most likely, this is because the old implementation uses the OPT++ library for the central optimization loop, while `perception_oru` uses its own optimization implementation. However, our tests with the old implementation has shown that it often is trapped in local minima when used in this benchmark, which results in poorer accuracy.

[4] https://github.com/ethz-asl/libpointmatcher

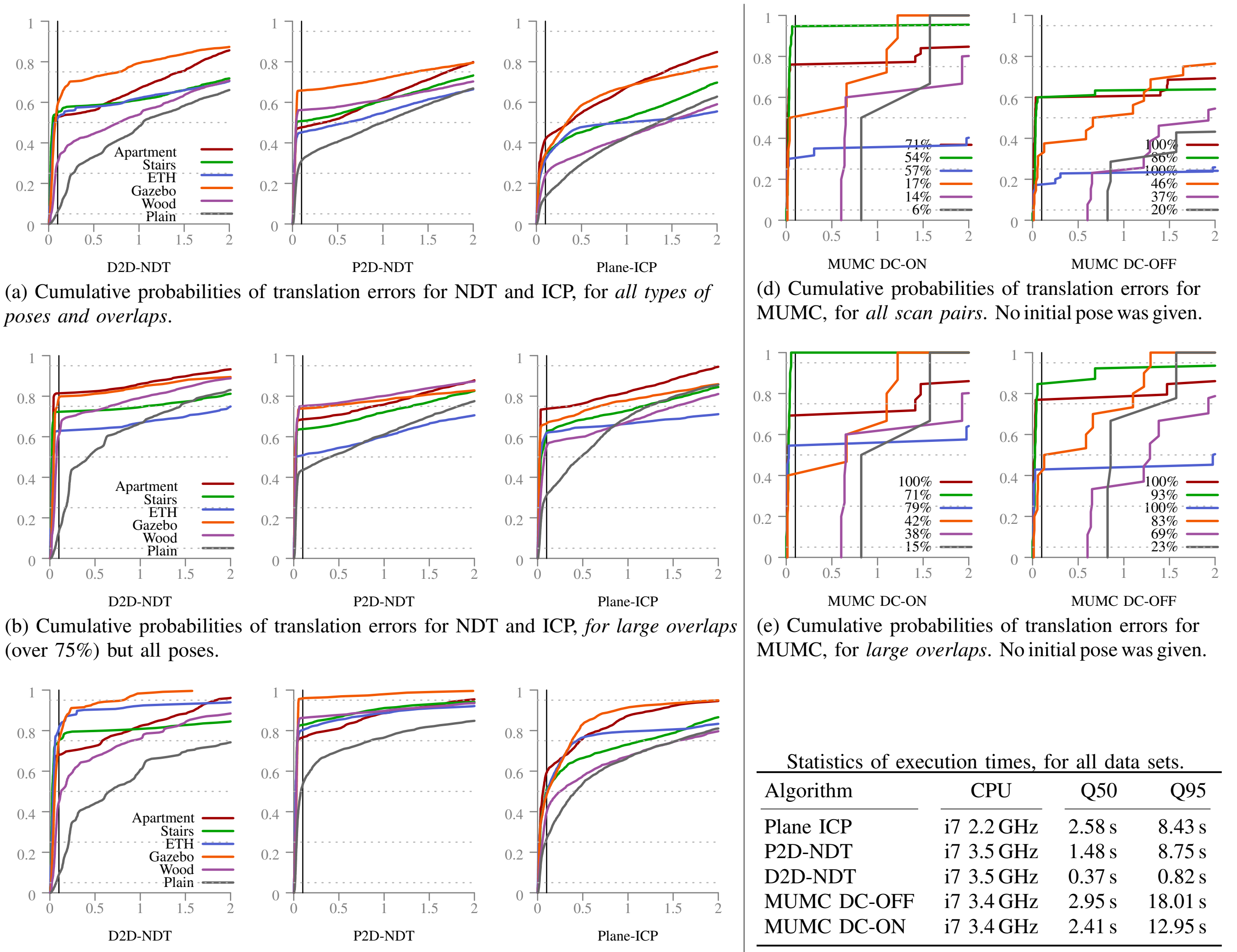
[5] http://aass.oru.se/~mmn/software/ndt_test_suite_v0.1.zip



(a) Cumulative probabilities of rotation errors for NDT and ICP, for all types of poses and overlaps.

(b) Cumulative probabilities of rotation errors for MUMC. No initial pose was given.

Fig. 2: Cumulative probability plots of rotation errors after registration. Rotation error (in degrees) is on the horizontal axis. The MUMC plots also include the percentage of scan pairs from which the plots are generated. A suggested threshold for successful matches (2.5°) is marked with a vertical bar.



(a) Cumulative probabilities of translation errors for NDT and ICP, for *all types of poses and overlaps*.

(b) Cumulative probabilities of translation errors for NDT and ICP, *for large overlaps* (over 75%) but all poses.

(c) Cumulative probabilities of translation errors for NDT and ICP, *for "easy" poses*, but all levels of overlaps.

(d) Cumulative probabilities of translation errors for MUMC, for *all scan pairs*. No initial pose was given.

(e) Cumulative probabilities of translation errors for MUMC, for *large overlaps*. No initial pose was given.

Statistics of execution times, for all data sets.

Algorithm	CPU	Q50	Q95
Plane ICP	i7 2.2 GHz	2.58 s	8.43 s
P2D-NDT	i7 3.5 GHz	1.48 s	8.75 s
D2D-NDT	i7 3.5 GHz	0.37 s	0.82 s
MUMC DC-OFF	i7 3.4 GHz	2.95 s	18.01 s
MUMC DC-ON	i7 3.4 GHz	2.41 s	12.95 s

Fig. 3: Cumulative probability plots of translation error after registration. Cumulative probabilities are on the vertical axes. Translation error (m) is on the horizontal axis, On the left are plots for the local methods (NDT and ICP), and on the right are plots for the global methods. The MUMC plots also include the percentage of scan pairs from which the plots are generated. A suggested threshold for successful matches (10 cm error) is marked with a vertical bar.

VI. Conclusions and future work

Judging by the results summarized in Figs. 2–3, which is the result of over 120 000 scan matches, we conclude that MUMC and NDT generally provides more robust registration than point-to-point or point-to-plane ICP when faced with scan pairs that have either small overlap or a poor initial alignment. Plots for point-to-point ICP have been omitted due to space constraints. The main advantage of MUMC is that it performs well even when no initial pose estimate (from odometry) is available, as long as there is sufficient structure in the environment. The two NDT-based methods (P2D-NDT and D2D-NDT) do require initial estimates, but are much less sensitive to errors in this estimate than ICP. D2D-NDT and, in particular, P2D-NDT work better than MUMC outdoors, because they have less strict assumptions on planarity. D2D-NDT is the fastest of the evaluated methods, with a median execution time that is about 7 times shorter than non-NDT methods.

For situations where no initial pose estimate is available, MUMC provides a good solution in common indoor environments, and can also provide information about whether it has successfully matched two 3D scans or not. For real-time use on a robot, D2D-NDT is preferable because of its faster execution speed. For challenging data sets with little structure or little overlap, P2D-NDT provides the best accuracy.

The benchmark protocol used here was designed for local registration methods. In future work, we will adapt the benchmark to more fairly assess the performance also of global methods, by including more unique pairs of scans from the given datasets.

Generating datasets of similar quality (with precise 6-DOF ground truth poses) for other sensors types (e.g. Velodyne scanners or RGB-D cameras) and environments would also be very useful.

More sophisticated algorithms also return the covariance of the registration transform in addition to the transform itself. Some way to automatically assess the accuracy of this estimation, at least in terms of the primary uncertainty directions would be useful.

References

[1] Paul J. Besl and Neil D. McKay. “A Method for Registration of 3-D Shapes”. In: *IEEE Trans. Pat. Analysis and Mach. Intelligence* 14.2 (Feb. 1992), pp. 239 –256.

[2] Peter Biber and Wolfgang Straßer. “The Normal Distributions Transform: A New Approach to Laser Scan Matching”. In: *IROS*. Las Vegas, USA, Oct. 2003, pp. 2743–2748.

[3] Peter Biber, Sven Fleck, and Wolfgang Straßer. “A Probabilistic Framework for Robust and Accurate Matching of Point Clouds”. In: *26th Pattern Recognition Symposium (DAGM 04)*. 2004.

[4] Simone Ceriani, Giulio Fontana, Alessandro Giusti, Daniele Marzorati, Matteo Matteucci, Davide Migliore, Davide Rizzi, Domenico G. Sorrenti, and Pierluigi Taddei. “RAWSEEDS ground truth collection systems for indoor self-localization and mapping”. In: *Autonomous Robots* 27 (2009), pp. 353–371.

[5] Yang Chen and Gérard Medioni. “Object modeling by registration of multiple range images”. In: *ICRA*. 1991, pp. 2724–2729.

[6] A Das and S.L. Waslander. “Scan registration with multi-scale k-means normal distributions transform”. In: *IROS*. 2012, pp. 2705–2710.

[7] Andrew Howard and Nicholas Roy. *The Robotics Data Set Repository (Radish)*. 2003. URL: http://radish.sourceforge.net/.

[8] Martin Magnusson. “The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection”. Örebro Studies in Technology 36. PhD thesis. Örebro University, Dec. 2009.

[9] Martin Magnusson and Tom Duckett. “A Comparison of 3D Registration Algorithms for Autonomous Underground Mining Vehicles”. In: *ECMR*. Ancona, Italy, Sept. 2005, pp. 86–91.

[10] Martin Magnusson, Andreas Nüchter, Christopher Lörken, Achim J. Lilienthal, and Joachim Hertzberg. “Evaluation of 3D Registration Reliability and Speed — A Comparison of ICP and NDT”. In: *ICRA*. Kobe, Japan, May 2009, pp. 3907–3912.

[11] Martin Magnusson, Achim J. Lilienthal, and Tom Duckett. “Scan Registration for Autonomous Mining Vehicles Using 3D-NDT”. In: *J. Field Robotics* 24.10 (Oct. 2007), pp. 803–827.

[12] Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas. “Registration of Point Cloud Data from a Geometric Optimization Perspective”. In: *Proceedings of the Symposium on Geometry Processing*. 2004, pp. 22–31.

[13] Andreas Nüchter and Kai Lingemann. *Robotics 3D Scan Repository*. 2011. URL: http://kos.informatik.uni-osnabrueck.de/3Dscans/.

[14] Kaustubh Pathak, Dorit Borrmann, Jan Elseberg, Narunas Vaskevicius, Andreas Birk, and Andreas Nüchter. “Evaluation of the Robustness of Planar-Patches based 3D-Registration using Marker-based Ground-Truth in an Outdoor Urban Scenario”. In: *IROS*. Taipei, Taiwan, 2010, pp. 5725–5730.

[15] Kaustubh Pathak, Andreas Birk, Narunas Vaskevicius, and Jann Poppinga. “Fast Registration Based on Noisy Planes With Unknown Correspondences for 3-D Mapping”. In: *Robotics, IEEE Transactions on* 26.3 (2010), pp. 424–441. ISSN: 1552-3098.

[16] Kaustubh Pathak, Andreas Birk, Narunas Vaskevicius, Max Pfingsthorn, Sören Schwertfeger, and Jann Poppinga. “Online Three-Dimensional SLAM by Registration of Large Planar Surface Segments and Closed-Form Pose-Graph Relaxation”. In: *Journal of Field Robotics* 27.1 (2010), pp. 52–84.

[17] Kaustubh Pathak, Narunas Vaskevicius, and Andreas Birk. “Uncertainty analysis for optimum plane extraction from noisy 3D range-sensor point-clouds”. English. In: *Intelligent Service Robotics* 3.1 (2010), pp. 37–48. ISSN: 1861-2776.

[18] Kaustubh Pathak, Narunas Vaskevicius, Francisc Bungiu, and Andreas Birk. “Utilizing Color Information in 3D Scan-Registration Using Planar-Patches Matching”. In: *IEEE Int. Conf. on Multisensor Fusion and Information Integration*. Hamburg, 2012.

[19] François Pomerleau, M. Liu, Francis Colas, and Roland Siegwart. “Challenging data sets for point cloud registration algorithms”. In: *IJRR* 31.14 (Dec. 2012), pp. 1705–1711.

[20] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. “Comparing ICP Variants on Real-World Data Sets”. In: *Autonomous Robots* 34.3 (Apr. 2013), pp. 133–148.

[21] A. Segal, D. Haehnel, and S. Thrun. “Generalized-ICP”. In: *Proceedings of Robotics: Science and Systems*. Seattle, USA, 2009.

[22] Todor Stoyanov. “Reliable Autonomous Navigation in Semi-Structured Environments using the Three-Dimensional Normal Distributions Transform (3D-NDT)”. PhD thesis. Örebro university, 2012.

[23] Todor Stoyanov, Martin Magnusson, and Achim J. Lilienthal. “Fast and Accurate Scan Registration through Minimization of the Distance between Compact 3D NDT Representations”. In: *IJRR* 31.12 (12 2012), pp. 1377–1393.

[24] Todor Stoyanov, Martin Magnusson, and Achim J. Lilienthal. “Point Set Registration through Minimization of the L2 Distance between 3D-NDT Models”. In: *ICRA*. 2012.

[25] Narunas Vaskevicius, Andreas Birk, Kaustubh Pathak, and Soren Schwertfeger. “Efficient Representation in 3D Environment Modeling for Planetary Robotic Exploration”. In: *Advanced Robotics* 24.8-9 (2010), 1169—1197.

[26] O. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner. “Benchmarking urban six-degree-of-freedom simultaneous localization and mapping”. In: *J. Field Robotics* 25 (2008), pp. 148–163.

Where am I? An NDT-based prior for MCL

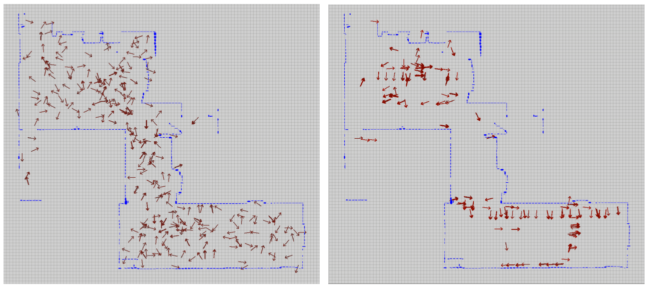
Tomasz Piotr Kucner, Martin Magnusson, Achim J. Lilienthal

Abstract—**One of the key requirements of autonomous mobile robots is a robust and accurate localisation system. Recent advances in the development of Monte Carlo Localisation (MCL) algorithms, especially the Normal Distribution Transform Monte Carlo Localisation (NDT-MCL), provides memory-efficient reliable localisation with industry-grade precision. We propose an approach for building an informed prior for NDT-MCL (in fact for any MCL algorithm) using an initial observation of the environment and its map. Leveraging on the NDT map representation, we build a set of poses using partial observations. After that we construct a Gaussian Mixture Model (GMM) over it. Next we obtain scores for each distribution in GMM. In this way we obtain in an efficient way a prior for NDT-MCL. Our approach provides a more focused then uniform initial distribution, concentrated in states where the robot is more likely to be, by building a Gaussian mixture model over potential poses. We present evaluations and quantitative results using real-world data from an indoor environment. Our experiments show that, compared to a uniform prior, the proposed method significantly increases the number of successful initialisations of NDT-MCL and reduces the time until convergence, at a negligible initial cost for computing the prior.**

I. Introduction

Localisation is a key component of most mobile robot systems today, e.g. in field robotics, intra-logistics or assistance robots. The main focus over the past years has been to increase localisation accuracy and efficiency. Multiple solutions are already widely employed. For large outdoor environments, localisation approaches are often based on GPS, while for indoor environments, industrial localisation systems are typically based on active or passive beacons. Such systems are capable of accurate localisation, although visibility constrains and the requirement for specific infrastructure in the form of installed beacons or GPS satellites is an important drawback. These localisation methods in current practice constrain robots to operate only within limited spaces, and additionally they impose additional deployment costs.

Conversely, it is very common in the robotics community to use map-based localisation approaches without additional infrastructure. Monte Carlo Localisation (MCL) [1] is one of the most popular map-based localisation approaches and has been shown to be robust in real-world scenarios [1]–[3]. Recent works of Saarinen et al. [4] and Valencia et al. [5] have shown that by using the Normal Distributions Transform (NDT) [6], [7] for representing the environment it is possible to obtain much higher localisation accuracy with lower memory and CPU requirements, compared to occupancy grid maps [8]. Thanks to the increased accuracy (localisation error along the path is less then 3 cm [5]) NDT-MCL provides localisation good enough to fulfil industrial requirements and to be used in commercial applications. This improvement enables the development of flexible autonomous robotics systems that are independent from external infrastructure while still achieving industry-grade accuracy. However a mechanism for accurate and fast initialisation and re-initialisation of the NDT-MCL variants was so far missing. In order to achieve accurate localisation from the very start of deployment, it is necessary that the localisation algorithm converges very quickly to the true pose, which is not the case when using a uniform prior.


(a) Uniform initialization

(b) GMM initialization

Fig. 1: Comparing the initial particle distribution using a uniform prior belief (a) vs an informed prior based on an NDT map (b).

The main contribution of this paper is a novel algorithm for constructing a prior for MCL from the robot's current sensor readings, exploiting the NDT map representation. In fact, the proposed initialisation method can be used together with any implementation of MCL. However, since it has been demonstrated [4] that NDT-MCL provides better pose estimates than occupancy grid based MCL and makes already use of an NDT map of the environment, we use NDT-MCL. We address the problem of initialisation and re-initialisation of NDT-MCL for cases where no external knowledge about the robot pose is available. We describe a method of building a Gaussian mixture model (GMM) representing the prior belief distribution of possible robot poses. The initial set of particles is then sampled from this GMM.

The remainder of the paper is organised as follows. Sec. II relates our work to the state of the art. Sec. III introduces our novel prior for NDT-MCL. Sec. IV describes the experimental setup and compares the localisation performance obtained with the proposed informed prior to initialisation with a uniform distribution.

Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. firstname.lastname@oru.se

This work was funded in part by the EU FP7 projects SPENCER (ICT-2011-600877) and the Swedish KK foundation under contract number 20110214 (ALLO).

978-1-4673-9163-4/15/$31.00 © 2015 IEEE

II. Related work

The Monte Carlo Localisation algorithm was first introduced by Dellaert et al. [1]. The MCL algorithm is a non-parametric Bayes filter where the belief distribution is represented as a finite set of particles. In contrast to parametric representations (e.g. Kalman filters) this has the advantage that it can represent also multimodal distributions. MCL is commonly implemented using occupancy grid maps [9] as the map representation [10].

Over the years, there have been several attempts to improve the quality of localisation with particle filters [2], [3]. Two notable recent improvements are NDT-MCL [4] and DT-NDT-MCL [5]. These approaches use NDT maps [6], [7] rather than occupancy grid maps for representing the structure of the environment and for evaluating the sensor model. The result is a significant improvement in accuracy even with rather coarsely discretised maps, which enables much more efficient mapping and localisation, both in terms of memory and CPU requirements [4].

Using standard MCL, initialisation is typically performed either by using a normal distribution centred around an initial guess of the robot pose, or by distributing particles uniformly all over the map (possibly with the addition of excluding poses that are known to intersect with obstacles in the map).

A noteworthy modification of the initial distribution of samples is presented in the work of Yee et al. [11]. In this work a regular grid of positions over the map is constructed. To compute the distance between grid points authors uses Monte Carlo tests to compute the error statistics as a function of separation. Next, the authors compute the most likely orientation for each grid point and compute the likelihood for it. This likelihood is later on used as a weight for a Gaussian associated to respective grid point. Moreover authors assume that each one has the same isotropic covariance. In this way they obtain a Gaussian mixture model which is later used for initialisation and re-initialisation. The basic idea of using Gaussian Mixture Model shows some similarities with method introduced in following paper, however idea of Yee et al. is closer to uniform initialisation. The grid points are distributed all over the map and later on the MCL filtering step is executed to implicitly define contributing particles. In contrast, our approach cuts the search space by defining areas of interest based on initial observations which later are evaluated. Moreover, the approach of Yee et al. uses the strong assumption that all the distributions are identical and isotropic.

Oh et al. [12] present a method to incorporate additional information in particles weights. Their method splits the map into regions and associates to each one a factor that describes the probability of the robot being inside that area. This approach introduces an additional bias that decreases the likelihood of particles in less likely areas (e.g., it is more likely that a robot is on the street than inside a wall). The major drawback of this method is the fact that this additional information is stored in separate static layer and has to be rebuilt each time some property of the environment changes. Moreover, any error in this layer might cause undesired behaviour of the filter by favouring some particles based only on their location even if they support the wrong hypothesis.

Dual MCL or Mixture MCL [2] suggests to invert localisation problem. Instead of first computing the new samples based on motion and then adjust its belief factor using observations authors suggest to sample form distribution based on observation and then adjust the importance factor based on the previous position of the robot. A similar but more recent approach is the observation-driven Bayes filter of He and Hirose [13]. Compared to these approaches, using NDT maps to generate poses from observations for the initial distribution is rather straight-forward. In comparison, He and Hirose [13] require pre-caching four meta-map representations and approximately one second of processing time per frame. Another interesting contribution towards implementing mixture MCL is the work of Elinas and Little [14]. However, in this work the authors employ stereo vision for localisation purposes and define a map as a set of SIFT features.

Instead of using SIFT features, we exploit the Normal Distributions Transform (NDT) environment representation. This method was introduced by Biber et al. [6] and later on extended to three dimensions by Magnusson et al. [7]. NDT is a piece-wise continuous representation, which represents space as a set of normal distributions, as opposed to occupancy grids, which represent space as a set of binary random variables. Moreover thanks to the extension by Saarinen et al. in [15] NDT Occupancy Map is able to store explicitly information about free and explored space within the environment, which is an additional asset in distributing samples over environment.

III. Prior distributions

We will now describe our informed NDT-based prior for MCL as well as the baseline uniform distribution.

A. Uniform distribution

As a baseline to evaluate the performance of our proposed approach we will use a uniform distribution of particles over the map. This approach was already discussed in the seminal work of Dellaert et al. [1].

To obtain a uniform distribution of particles over the map we use a two-step process. In the first step we uniformly draw one cell from the set of all unoccupied cells in the NDT grid (that is, all cells that do not contain a Gaussian representation of the local surface shape). In the second step we uniformly draw a position and orientation in the given cell. We repeat these two steps until we have acquired the desired number of particles. Thanks to this approach we make sure that all particles are placed in free space in the map, while keeping the execution time fixed.

B. Particle generation from GMM

In this section we will describe how to build the Gaussian mixture model (GMM) representing the prior belief distribution of the robot and how to obtain the initial set of particles from the GMM.

The procedure is as follows:

1) **Obtain promising poses -** NDT represents the global map as a set of Gaussians: $M_{NDT}^{G} = \{\mathcal{N}(\mu_j^G, \Sigma_j^G)\}_{j=1}^{N_G}$ To represent our current observation which is a set of n two-dimensional point samples $z = \{\mathbf{p}_i^z(x_i, y_i)\}_{i=1}^{n}$, we also use NDT: $\bar{Z} = \{\mathcal{N}(\mu_j^Z, \Sigma_j^Z)\}_{j=1}^{N_{\bar{Z}}}$ To build a set of possible poses we first compute the Cartesian product of the global map and the current observation: $M_{NDT}^{G} \times \bar{Z} = \{(g, \bar{z}) | g \in M_{NDT}^{G} \wedge \bar{z} \in \bar{Z}\}$ In this way combine each Gaussian from observation with each Gaussian from map. Next, for each pair $(g, \bar{z})$, we compute the pose $\mathbf{q}$ of the robot with respect to g. To solve this problem first we have to find the transformation, which will align eigen vectors with highest eigens value in $\bar{z}$ and g. Then we apply this transformation on robot pose $\mathbf{q}_{\bar{z}}$ and as a result we get $\mathbf{q}$. In this process we obtain a set $\mathcal{Q} = \{\mathbf{q}_i(x_i, y_i, \theta_i)\}_{i=1}^{N_z N_g}$ of possible poses of the robot in the coordinate frame of the global map. The size of $\mathcal{Q}$ depends on the size of the environment, the sensor field of view and the map resolution. In experiments we have observed that for an environment of size 25 x 25 m and resolution of 0.2 m number of the elements of $\mathcal{Q}$ is between 1869 and 3900, for 0.5 m is between 1755 and 2516, and for 1.0 m is between 803 and 1371.

In Fig. 3 we can see how initial set of hypotheses is generated. We can see that the current observation contains two distributions (blue and yellow). We align each Gaussian from the observation with a Gaussian in the map. Since we know what is the robot pose with respect to each Gaussian, we can transform robot pose to the global coordinate frame and obtain a set of possible poses. In Fig. 4 we can see a visualisation of all possible hypotheses obtained during one such initialisation.

2) **Build GMM -** The set of poses obtained in the previous phase implicitly mark regions of interest. To estimate the likelihood of those map regions we will generate a GMM in pose space. First we split the *state space* into a regular voxel grid $\mathcal{V} = \{v_j\}_{j=1}^{N_V}$. For simplicity of further discussion we assume that each voxel is a set of all possible poses within predefined ranges: $v_j = \{(x, y, \theta) | x \in [x_{min}^j, x_{max}^j) \wedge y \in [y_{min}^j, y_{max}^j) \wedge \theta \in [\theta_{min}^j, \theta_{max}^j)\}$.For each voxel v_j that contains pose particles we estimate the corresponding normal distribution in the following way:

$$\mu = \frac{1}{n} \textstyle\sum_{i=1}^{i=n} \mathbf{q}_i \tag{1}$$

$$M = [\mathbf{q}_1 - \mu ... \mathbf{q}_n - \mu] \tag{2}$$

$$\Sigma = \frac{1}{n-1} M^T M \tag{3}$$

In this way we obtain a Gaussian mixture model representing an informed prior on the poses: $M_{GMM}(\mathcal{Q}) = \{\mathcal{N}(\mu_j, \Sigma_j)\}_{j=1}^{N}$ where $N \leq |\mathcal{Q}|$ and $N \leq |\mathcal{V}|$. The next step is to estimate the weight w_j of each distribution in the set $M_{GMM}(\mathcal{Q})$. As a weight we will use the L_2 likelihood of the current observation at the mean pose, given the global map. The pose likelihood is computed as in Saarinen et al. [4]: $w_j = (\sum_{i=1}^{N} L_2^i)^{-1} L_2^j$ L_2 likelihood is the likelihood that the robot has a particular pose because it is consistent with several parts of the observation.

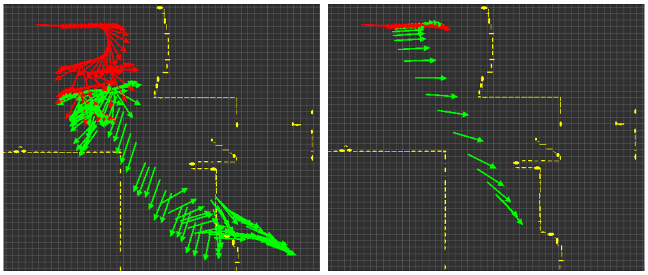
$$L_2^j(\bar{Z} | \mathbf{q}_j, M_{NDT}^G) = \tag{4}$$
$$\textstyle\sum_{i=1}^{N_G} \sum_{k=1}^{N_{\bar{Z}}} d_1 \exp(-\frac{d_2}{2} \mu_{ik}^T (R_j \Sigma_i^G R_j^T + \Sigma_k^Z)^{-1} \mu_{ik})$$

where $\mu_{ik} = R_j \mu_i^Z + t_j - \mu_k^m$ and d_1 and d_2 are scaling factors.

Each pose $\mathbf{q}_j$ can be represented as a rotation matrix R_j and translation t_j with respect to the global coordinate frame. L_2^j represents the likelihood of the current observation represented as NDT $\bar{Z}$ given the global map and the state $\mathbf{q}_j$.

Fig. 5 shows the means of all distributions of the GMM generated from the distribution shown in Fig. 4.

3) **Sampling -** The final step is to draw a set of initial particles from the GMM. The probability of drawing a sample $\mathbf{q}$ can be expressed as a sum of n weighted Gaussians: $p(\mathbf{q}) = \sum_{j=1}^{n} w_j \mathcal{N}(\mathbf{q} | \mu_j \Sigma_j)$. In this work we assume that obtained GMM is sparse therefore we can approximate this probability in the following way: $p(\mathbf{q} | \mathbf{q} \in v_j) = w_j \mathcal{N}(\mathbf{q} | \mu_j \Sigma_j)$ This approximation allows us to build a simple two step sampling algorithm. In the first step we draw a voxel according to its weight w and in the second step we draw the pose according to the normal distribution within the voxel.



(a) Uniform initialisation

(b) GMM initialisation

Fig. 2: Track of convergence (500 particles, cell size = 0.2[m]) - ground truth (red), NDT-MCL localisation estimate (green). We can see here how many localisation updates are necessary to reach correct pose estimate.

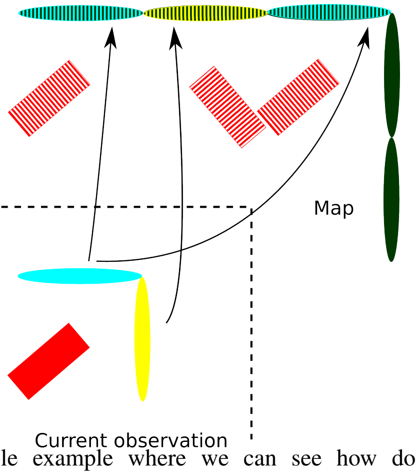
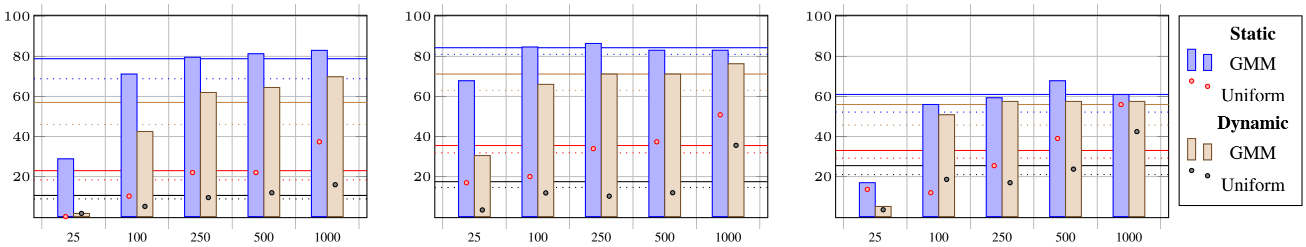
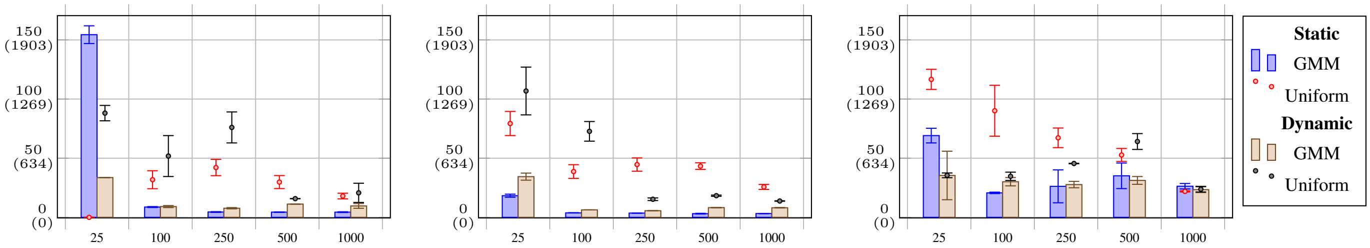


Fig. 3: Simple example where we can see how does aligning Gaussians from observation with ones from map leads to initial set of hypotheses.



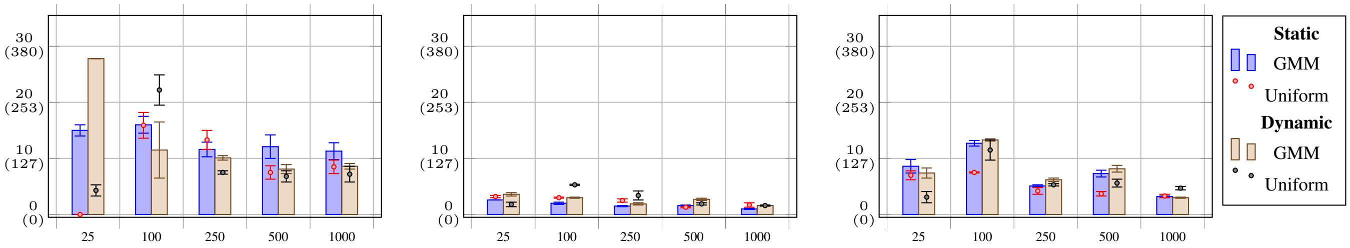
(a) map resolution 0.2 m (b) map resolution 0.5 m (c) map resolution 1.0 m

Fig. 6: Success rate [%] (dotted line - average success rate, solid line - the average success rate for 4 best cases)



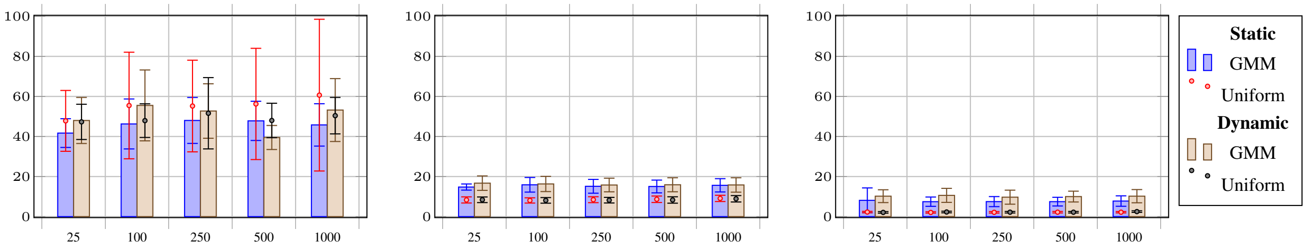
(a) map resolution 0.2 m (b) map resolution 0.5 m (c) map resolution 1.0 m

Fig. 7: Time until correct pose estimate [s (# of updates)]



(a) map resolution 0.2 m (b) map resolution 0.5 m (c) map resolution 1.0 m

Fig. 8: Time until convergence [s (# of updates)]



(a) map resolution 0.2 m (b) map resolution 0.5 m (c) map resolution 1.0 m

Fig. 9: Time for computing the prior distribution [ms]

IV. Experiments and Results

A. Experimental Setup

We have evaluated the approach presented in Section III-B using two data sets, recorded in a static and a dynamic environment. The static data set is publicly available[1]. In the test data sets, the robot traverses a closed loop (see Fig. 10) multiple times with velocity of 1 m/s in an indoor environment. Both data sets were collected in the basement of Örebro university using a commercial Automatically Guided Vehicle (AGV) system from Kollmorgen Automation AB. A Master Controller (VMC 5000) controls the vehicle along predefined trajectories. The ground truth was obtained with a commercial infrastructure-based positioning system, which tracks wall-mounted reflectors using a rotating laser. After setup and calibration, this system provides accurate (according to its specification accuracy should be approx. 1 cm or less) position information. For infra-structure free localisation we use a LIDAR with field of view of 270 degrees and range of 18 m. The data set covers a 25 m × 25 m area. In both cases the robot was travelling along the same predefined path with the same velocity. To emulate a dynamic environment, we have asked a group of people to not only move around in the environment, but also on purpose to disturb the localisation

[1] Data sets are available under:
http://mrolab/datasets.html.

process by changing the shape of the environment with panels or even to occlude the laser with them.

The goal of the experiment was to investigate how using an informed prior (see Fig. 1) will affect global localisation. The comparison in this paper is done between uniform initialisation of NDT-MCL and GMM initialisation of NDT-MCL, as described in Section III.

We have chosen 60 random points along the path which represent different starting positions for the localisation process. We use the following four evaluation criteria:

1) Success rate - how many times the robot manages to localise itself correctly. We consider that the robot has localised it self correctly if the error is less then 10 cm.
2) Initial localisation time - how long does it take to minimise the localisation error with respect to ground truth. In case of localisation failure the measurement was discarded.
3) Convergence time - how long it takes before the value of one standard deviation is less then 10 cm and 5 degrees. This metrics was computed only for the cases when the localisation was performed successfully.
4) Computation time - how much time it takes to generate the prior.

We have tested five particle populations sizes (25, 100, 250, 500, 1000 particles) for maps of three different resolutions (0.2 m, 0.5 m, 1.0 m). For the coarsest map (resolution 1m), the resolution of the voxel grid in pose space was 1.5 m and $\frac{\pi}{2}$ radians, and for the other two map resolutions the pose voxel grid was 0.5 m and $\frac{\pi}{2}$ radians. Please recall that we have been performing our tests using NDT maps and evaluating two different priors for NDT-MCL. If we would use a regular occupancy or octomap it would be impossible to achieve accuracy below 10 cm for maps with grid cells as big as 0.5x0.5 m^2 or 1.0x1.0 m^2 [4].

B. Results

In Fig. 6 we can see that, as long as the map resolution is sufficient enough for accurate localisation, we achieve a high success rate using the GMM prior even with a relatively small number of particles. In the best case the success rate was as high as 86% for the static environment and 76% for dynamic. The best result obtained with a uniform distribution for population containing 1000 particles is only as high 56%. It is worth noticing that for a static environment above some threshold (in our experiments around 100 samples), the average success rate is stable and does not change much when changing the number of particles and depends mainly on the map resolution. In case of the dynamic environment we can observe that the success rate increases with the number of particles, however, the success rate is high for each population equal or bigger than 100 particles. Also for dynamic environments the success rate is higher for initialisation using GMM than with uniform distribution. In Fig. 6 the average success rate across all populations is marked for a given map resolution with a dotted line. The average success rate for the four biggest sample populations is marked with a solid line. We can see that the average success rate with an NDT-based prior is always higher then the initialisation with a uniform prior. This observation is true both in static and dynamic environment.

Another interesting feature of the NDT-based prior is that it allows to localise quickly (see Fig. 7). In the best case the average localisation time was as short as 3 seconds (or 38 updates of the particle filter), while initialisation based on a uniform distribution was never faster then 18 seconds (228 updates). All the timing plots in Figs 7–9 show the results only for the cases where the filter succeeded in localising. As a consequence, the plot for 25 uniformly distributed particles in Fig. 7(a) shows zero seconds, because none of those runs succeeded. In Fig. 7 it is visible that increasing the size of the particle population does not change significantly time before successful localisation. The time needed by both distributions to converge is comparable and usually low, despite of that time needed to localise correctly is significantly shorter for proposed prior then for uniform one. The result of this time difference is visible in Fig. 2. Where the pose estimate is *following* the correct estimate.

In Fig. 9 we can see that in most cases computing the GMM-based prior only takes a few milliseconds more than computing a uniform prior. If we compare this value against the average time between two laser scans in this data set, which is 70 ms, we can assume that it is possible not only to use this method

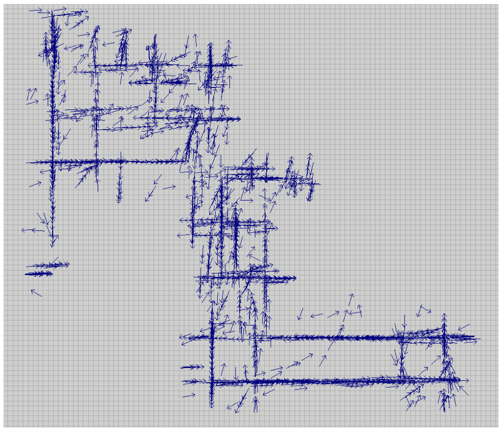


Fig. 4: Poses generated in step 1 of the initialisation algorithm, from which the GMM is created in step 2.

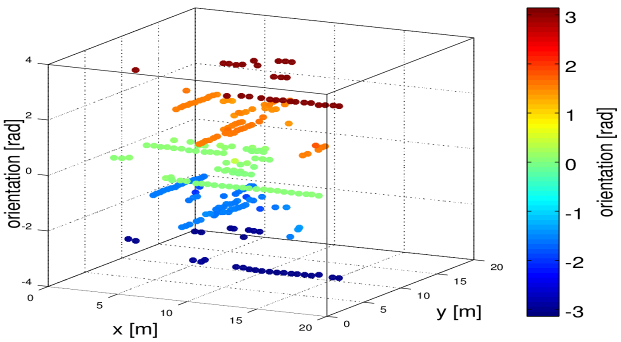


Fig. 5: An example of set of mean values for each component in GMM in map with resolution 0.2 m.

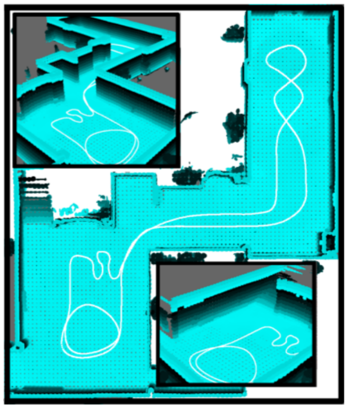


Fig. 10: Test environment with predefined path [4].

for initialisation but also for re-initialisation. If we want to use NDT-based prior for non-NDT MCL we have to remember that additional time will be required to build map of environment using NDT.

To evaluate the usefulness of the GMM method in case of re-initialisation, we have manually triggered resets of the localisation system during the robot runs in static environment. We have observed that after a high spike in the localisation error (at the moment of reset) the error drops again. During the test run we have reset the system 26 times and managed to recover in 20 cases when using the informed prior. For the system using uniform distribution for re-initialisation the success rate was only 7 recoveries for 26 resets.

V. Conclusions

In this paper we have introduced a method for constructing an informed prior for MCL. Based on a map of the environment we build a GMM which represents likely poses of the robot. Then we sample from the GMM distribution to obtain the initial set of particles.

The method introduced in this paper shows a way to implicitly define regions of interest, by removing areas that have no support from current observations. This is a major improvement in comparison to methods which are evaluating poses all over the given map. Moreover, the method builds and maintains a probabilistic model of the robot pose estimate on the fly. Therefore it does not require any additional process after obtaining the map of environment, such as assigning environment classes to regions of the map [12]. It also makes the method flexible enough to incorporate new information acquired by the sensor. Although such observation-driven priors have been used for other MCL implementations previously, this is the first implementation of a method for generating an informed prior for MCL. The main motivation for our method is the recent demonstrations [4], [5] of using NDT-MCL to achieve superior accuracy in dynamic industrial environments while maintaining a small memory footprint and low CPU requirements.

We also have performed a series of experiments both in a static and a dynamic environment demonstrating that the proposed method is able to perform global localisation with fewer particles, in comparison to the baseline uniform distribution. Moreover we have shown that an NDT-MCL particle filter initialised with NDT-based prior converges faster than when using a uniform distribution. We have also demonstrated that the significantly decreased localisation time (as measured in number of seconds or laser scans after initialisation) can be achieved with only a negligible one-time computational cost of a few milliseconds for generating the prior.

VI. Future work

In future work, we will extend the GMM method to 3D and also evaluate its performance with sensors that have a smaller field of view (e.g., RGB-D cameras).

References

[1] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1322–1328.

[2] S. Thrun, D. Fox, W. Burgard, and others, "Monte Carlo localization with mixture proposal distribution," in *AAAI/IAAI*, 2000, pp. 859–865.

[3] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012*. IEEE, 2012, pp. 3158–3164.

[4] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, "Normal Distributions Transform Monte-Carlo Localization (NDT-MCL)," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[5] R. Valencia, J. Saarinen, H. Andreasson, J. Vallv, J. Andrade-Cetto, and A. J. Lilienthal, "Localization in highly dynamic environments using dual-timescale NDT-MCL," in *IEEE International Conference on Robotics and Automation (ICRA), 2014*. IEEE, 2014, pp. 3956–3962.

[6] 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)*, vol. 3. IEEE, 2003, pp. 2743–2748.

[7] 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, Oct. 2007.

[8] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

[9] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," in *Proceedings of the Sixth Conference on Uncertainty in AI*, vol. 2929, 1990.

[10] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[11] M. Y. Yee and J. Vermaak, "A grid-based proposal for efficient global localisation of mobile robots," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 5. IEEE, 2005, pp. v–217.

[12] S. M. Oh, S. Tariq, B. N. Walker, and F. Dellaert, "Map-based priors for localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS).*, vol. 3. IEEE, 2004, pp. 2179–2184.

[13] T. He and S. Hirose, "Observation-driven Bayesian Filtering for Global Location Estimation in the Field Area: Observation-driven Bayes filters for global localization," *Journal of Field Robotics*, vol. 30, no. 4, pp. 489–518, Jul. 2013.

[14] P. Elinas and J. J. Little, "MCL: Monte-Carlo localization for mobile robots with stereo vision," in *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[15] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal, "Normal distributions transform occupancy maps: Application to large-scale online 3D mapping," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2233–2238.