Intra-Logistics with Integrated Automatic Deployment:
Safe and Scalable Fleets in Shared Spaces

H2020-ICT-2016-2017

Grant agreement no: 732737

DELIVERABLE 2.1

Report on Spatio-Temporal Representations for Long-Term Operations in Intra-Logistics

Due date: month 24 (December 2018)
Deliverable type: R
Lead beneficiary: UoL

Dissemination Level: PUBLIC

Main author: Tom Duckett (UoL)

1 Introduction

This report presents a summary of the work on spatio-temporal representations for long-term operation (task T2.1), and the application of these representations for long-term localisation and mapping (task T2.2) and learning of activity patterns (task T2.3), in the EU H2020 project ILIAD during its first 24 months. The research has been jointly developed by partners University of Lincoln (UoL) and Örebro University (ORU), with UoL being the main beneficiary.

The main objective of WP2 is to ensure long-term operation of the ILIAD system. The system should maintain and update its representations of the environment over time and learn site-specific information for each particular logistics warehouse. This includes monitoring both (i) the typical activity patterns of human workers and (ii) the effects of those actions, by learning the dynamics of expected activity and adapting to unexpected changes in the environment over time. It further includes (iii) self-monitoring the quality of localisation and mapping.

In summary, the main achievements of WP2 during the first 24 months of the project to meet these objectives have provided the ILIAD system with an ability to:

- efficiently represent the changing structure and semantics of the environment over time;
- model the environment dynamics;
- reliably self-localise in dynamic and changing environments; and
- predict the presence and movement of humans at specific times given the current context.

The rest of this report is structured as follows. Section 2 reviews the fundamental spatio-temporal representations developed for learning and prediction across the ILIAD system (T2.1). Section 3 describes the application of these representations within novel algorithms for persistent, performance-aware mapping and self-localisation (T2.2), building on the methods for sensor calibration, mapping and self-localisation developed in WP1, as reported under Deliverable D1.2. Section 4 describes the further application of the developed spatio-temporal representations for learning and predicting site-specific models of human activities, corresponding to the typical patterns of activity at a particular warehouse site (T2.3). Finally, Section 5 provides a summary and draws conclusions on the work so far, together with the outlook and future research directions for the ongoing research in WP2 of the ILIAD project.

2 Spatio-Temporal Representations for Long-Term Operation

In task T2.1 we have surveyed Artifical Intelligence (AI) approaches and representations for long-term operation [1], as part of a Special Issue of the IEEE Robotics and Automation Letters (RA-L) journal on AI for Long-Term Autonomy [2]. The survey [1] reviewed AI approaches for long-running robots, including key aspects of knowledge representations in the context of navigation and perception, such as the maintenance of multiple representations of possible environment states, robustness to appearance change and learning about dynamics, as well as reasoning about the locations of objects and people in the environment. We have further proposed, implemented and evaluated a number of complementary representation schemes, described in the following sections.

Spatio-temporal maps, as opposed to spatial-only, geometric maps, are attracting more and more interest in the field of robotic mapping. While traditionally, mapping has been performed using a static-world assumption, or including methods for actively filtering out moving objects, we are seeing a greater interest in map representations that explicitly represent time-dependent events: for example, temporal patterns of when a particular place is occupied or not, or the the motion patterns through an environment. For a detailed review of approaches, please see [3, 1].

2.1 Spatio-temporal representations in ILIAD

In the remainder of this section, we introduce the spatio-temporal representations that we have worked on in ILIAD's task T2.1. There are mainly four representations that have either been developed entirely, or substantially contributed to, as part of ILIAD; Circular–Linear Flow Field maps (CLiFF-map) [4], Spatio-Temporal Flow maps (STeF-map) [5, 6], Warped Hypertime (WHyTe) [7], and finally deep neural networks for spatio-temporal mapping [8, 9].

These representations are complementary, as each covers different aspects of spatio-temporal mapping. CLiFF-map (Section 2.2) represents local motion patterns in a continuous fashion, but does so at discretely sampled locations in a map. STeF-map (Section 2.3) similarly represents local motion patterns at discrete locations. It further adds information about temporal variability; i. e. how motion patterns change over time (such as different times of day). However, it represents motion patterns as a discrete set of directions only, and does not represent the speed or intensity of each pattern. Warped Hypertime (Section 2.4) is a generative model that is designed for predicting future states of the map, including occupancy, velocities, etc. Of these three representations, CLiFF will be preferable for global flow-aware motion planning when high spatial resolution is needed and STeF will be preferable in environments with strongly time-dependent motion patterns, while WhyTe is expected to be preferable when applied to motion prediction. In WP5, we are currently developing a hierarchical motion planner that can make use of several underlying maps (and their respective cost functions) and select the best available trajectory. Finally, the deep-learned representations (Section 2.5) are utilised for different purposes, namely to facilitate long-term updates of semantic map information and fast global localisation.

2.2 Circular–Linear Flow Field maps (CLiFF-map)

One map representation of environment dynamics that has been developed within ILIAD is the Circular–Linear Flow Field map (CLiFF-map) [4], which is a probabilistic approach for general flow mapping. It is designed to handle the motion of objects (e. g. people) as well as the flow of continuous media (e. g. air). CLiFF-map represents motion patterns using multimodal statistics to jointly represent speed and orientation, also enabling the reconstruction of a dense map from spatially and temporally sparse data.

The CLiFF-map model consists of a set of Gaussian mixture models (GMMs), each representing typical motion directions and speeds at a specific location. CLiFF-map also maintains variables describing the confidence of its model at each location, as well as the likelihood of observing motion there. To deal with the circular nature of the random variable representing direction, the probabilistic model of velocity assumes a *semi-wrapped* normal distribution, which can be seen as a distribution on a cylinder such that the orientation is wrapped around its circumference and the speed is directed along its height (see Figure 1).

Figure 2 summarises the main points in the CLiFF-map mapping procedure. The first step (data collection) is to record velocity measurements. For flows of people, this can be done either via people tracking with on-board sensors, or with data from an overhead camera. For air flow measurements, it can be done with an on-board anemometer. For

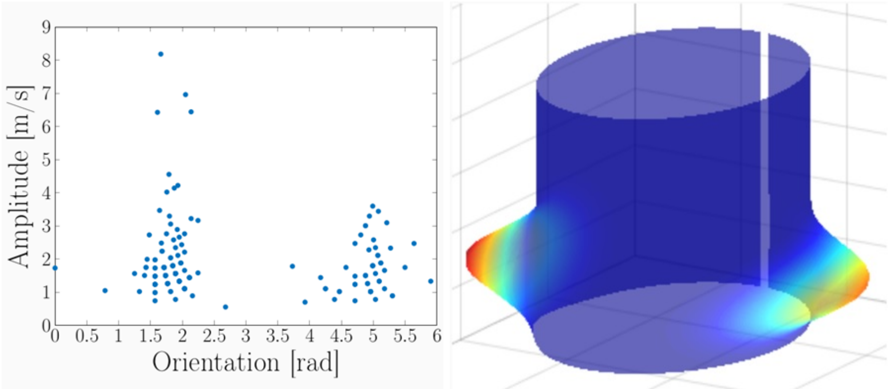


Figure 1: Illustrating the semi-wrapped Gaussian mixture model used by CLiFF-map. Left: velocity observations at one location. Right: the CLiFF-map component at this location. In this example, the model represents the raw data by a mixture of two Gaussian model components.

each location, sampled from the area covered during data collection, the raw velocity measurements are clustered and used to generate a semi-wrapped Gaussian mixture model (as shown in Figure 1 and the top-right part of Figure 2). From the potentially sparse locations, the map can be densified using a process called imputation, where expected data at new locations are computed from the surrounding observed locations. Finally, the quality of the map is estimated with a Gaussian process, which combines the observed motion ratio and observation ratio into a continuous measure of trust for each point in the map.

In summary, CLiFF-map builds a time-averaged model of the flow field at a discrete set of locations. For further details, please refer to the work of Kucner et al. [4, 10]. Section 4 in the present document illustrates how the representation can be used to learn site-specific patterns.

2.3 Spatio-Temporal Flow maps (STeF-map)

The human activities in an environment may change over time, e. g. pedestrian flows at the entrance of a work place at the start and end of a shift are likely to be in opposite directions. While the CLiFF-map representation described above builds a model of the average flow field, the aim of the STeF-map representation [5, 6] is to create a model of human motion which is able to predict the flow patterns of people over time, as well as where and when these flows are happening.

The underlying geometric space is represented by a grid, where each cell contains k temporal models, which correspond to k discrete orientations of pedestrian motion through the given cell over time (where $k = 8$ in our experiments as in Figures 3 and 17a). The temporal dimension of activities is modelled using periodic functions, by applying the Frequency Map Enhancement (FreMEn [11]) to recorded detections of human motions in each of the k directions in each cell of the grid. FreMEn is a mathematical tool based on the Fourier Transform, which considers the probability of a given state as a function of time and represents it by a combination of harmonic components. Then, transferring the most prominent spectral components to the time domain provides an analytic expression representing the probability of human motion (in a given direction for a given grid cell) at a given time in the past or future.

After computing the probabilities for every single orientation, we conclude that the

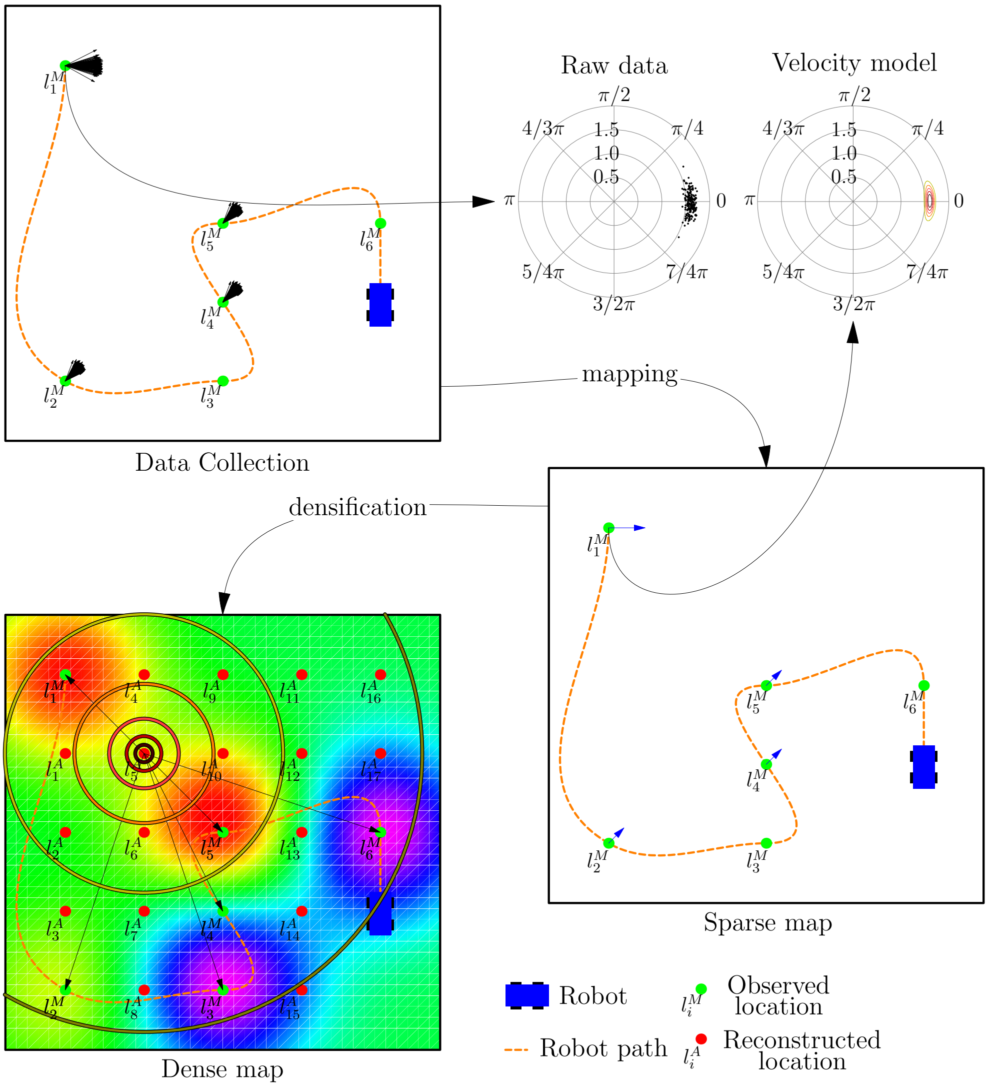
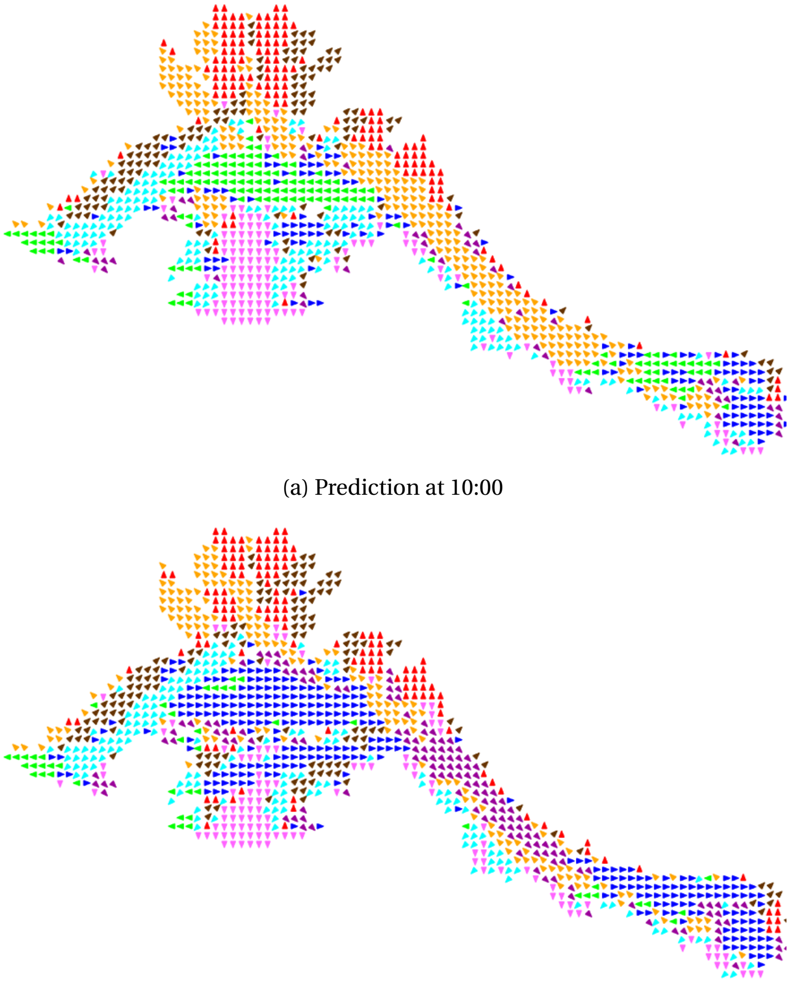


Figure 2: Overview of the CLiFF mapping procedure. *Top left, Data collection*: A robot is travelling through an environment collecting velocity measurements. In four locations dynamics were observed (black arrows), while in two locations there was no motion. (The velicity measurements from one location is shown in polar coordinates in the "raw data" plot at the top right.) *Bottom right, Sparse map*: Based on the raw data, to each observed location in the map a probabilistic velocity model (GMM) is associated (as shown in Figure 1. Based on the amount of collected data a Gaussian Process in constructed, estimating the trust in the estimated velocity models in unobserved locations. *Bottom left, Dense map*: To obtain a dense map, information from observed locations (green points) are combined to estimate the velocity model also in unobserved locations (red points). The influence of observed locations is proportional to the distance and intensity of motion. This figure is reprinted from Kucner [10].



(a) Prediction at 10:00

(b) Prediction at 18:00

Figure 3: STeF-map representation at two different times after some training.

dominant orientation a every cell for that instant of time t, corresponds to the orientation with the highest predicted probability:

$$\text{cell}_\theta = \text{argmax}(p_\theta(t)), \qquad \theta \in i\frac{2\pi}{k},\ i \in \{0,1,\ldots,k-1\}. \tag{1}$$

If we apply that approach for every single reachable cell in the map, we obtain a map representation of people flow that evolves over time if some rhythmic patterns are found over time (Figure 3).

2.4 Warped Hypertime

In ongoing work, we are also developing a "warped hypertime" spatio-temporal representation which facilitates learning and prediction of both discrete and continuous spatial representations [7, 12]. An overview of the approach is given in Figure 4.

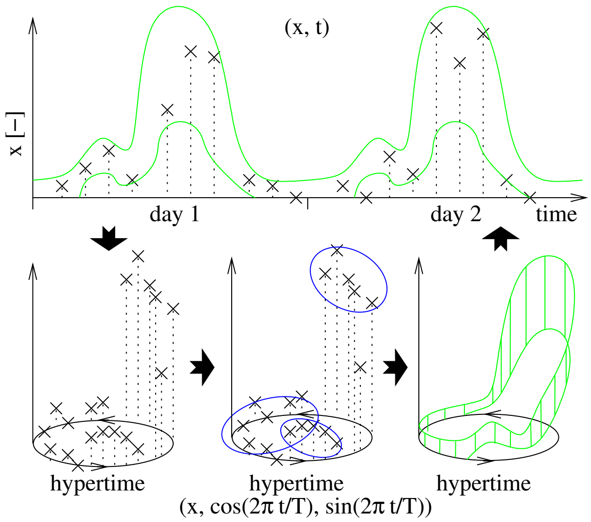


Figure 4: Warped hypertime representation: The data points (x,t) observed over time (top, black) are first processed by frequency analysis [11] to determine a dominant periodicity T. Then, the time t is projected onto a 2D space (called hypertime) and the vectors (x,t) become $(x, \cos(2\pi\, t/T), \sin(2\pi\, t/T))$. (See bottom, left). The projected data are then clustered (bottom, center, blue) to estimate the distribution of x over the hypertime space (green). Projection of the distribution back to the uni-dimensional time domain allows to calculate the probabilistic distribution of x for any past or future time.

This representation first uses FreMEn [11] to identify (temporally) periodic patterns in the data gathered. Then, it transforms each time periodicity into a pair of dimensions that form a circle in 2D space and adds (concatenates) these dimensions to the vectors that represent the spatial aspects of the modelled phenomena. Finally, a generalised model is built by applying traditional techniques like clustering or expectation-maximisation over the warped time-space representation. The resulting multi-modal model represents both the structure of the space and temporal patterns of the changes or events. In this way, the proposed method can turn a spatial representation into a spatio-temporal one by extending it with several wrapped dimensions representing time, with each pair of temporal dimensions representing a given periodicity observed in the gathered data. We hypothesise that since this model respects the spatio-temporal continuity of the modelled phenomena, it will provide more accurate predictions than models that partition the modelled space into discrete elements, or that models which neglect the temporal aspects.

2.5 Deep Learned Representations

In contrast to the above hand-crafted representations, we have also investigated deep neural networks for spatio-temporal representation fusing quantitative and qualitative information, e. g. for predicting human motion patterns [8] and long-term 3D semantic mapping [9].

A Convolutional Neural Network (CNN) is a partially connected neural network where the neurons are connected to the features within a set of spatial constraints. This architecture can be achieved through a "convolution" operation, which is highly parallelisable with GPU implementation. Due to this partial connectivity, CNN-based architectures have been widely-used in image-processing applications. Compared to hand-engineered features, CNNs can learn task-specific visual features from low-level geometric features to

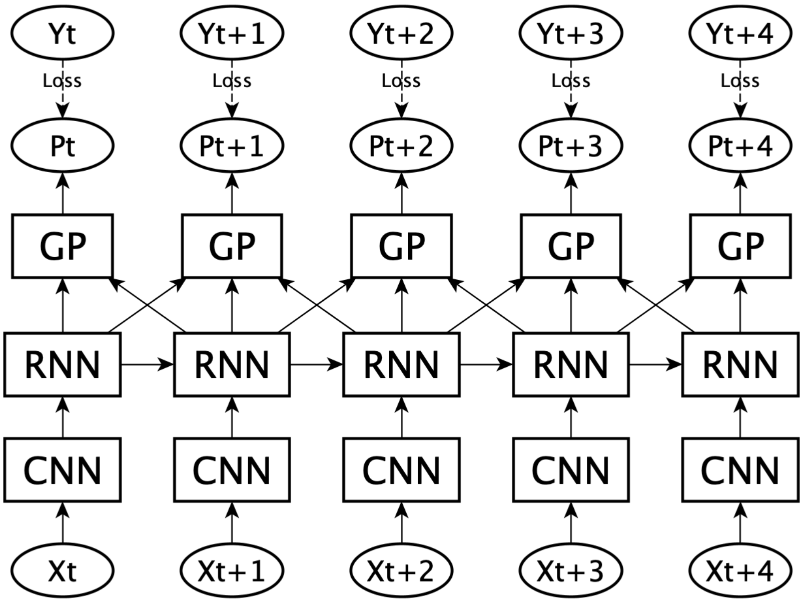


Figure 5: The deep learning architecture for learning from sequential data.

high-level semantic features with end-to-end learning. A Recurrent Neural Network (RNN) is a class of neural networks, which can learn the transitions of the hidden state given a sequence of observations. Hence, RNNs are widely used for temporal prediction from consecutive observations. Complementary to neural network architectures, e.g. CNN and RNN, a Gaussian Process (GP) is a Bayes-based non-parametric model to approximate the posterior of the data under the assumption of a multi-variate Gaussian prior for regression and classification problems. As a non-parametric method, GP measures the similarity of all the training examples from the kernel function and predicts the unknown examples with both mean value and its deviation (uncertainties).

Within ILIAD, we leverage the advantages of different types of neural networks and Gaussian processes for different tasks. That is, we use CNN for visual feature learning, RNN for sequential prediction and GP for uncertainty modelling. As a consequence, we use a sequence-to-sequence encoder-decoder model for learning from sequential data. The inputs are a sequence of robot observations $O_t, O_{t+1}, ..., O_{t+n}$ and the learning target $P_t, P_{t+1}, ..., P_{t+n}$ (this can be any learning target, e. g. robot position, semantics, pedestrian trajectory, etc.) For each observation, a CNN or multi-layer perception is used to learn an effective feature embedding. Then the CNN-learned features are used as the input of the recurrent neural network (RNN), and the RNN can learn the transitions of prediction targets in a hidden state space. With the consideration of consecutive observations, the RNN is likely to make more robust predictions with a properly learned “remember” or “forget” mechanism. Moreover, a (non-parametric) Gaussian Process can be applied to the outputs of an RNN to learn the uncertainties for applications where prediction uncertainties are important to know.

This architecture is generic and can be used for solving multiple robotic problems. For example, we have adapted this architecture for long-term self-localisation in Section 3.1, long-term semantic mapping in Section 3.2, and the prediction of pedestrian trajectories in Section 4.4.

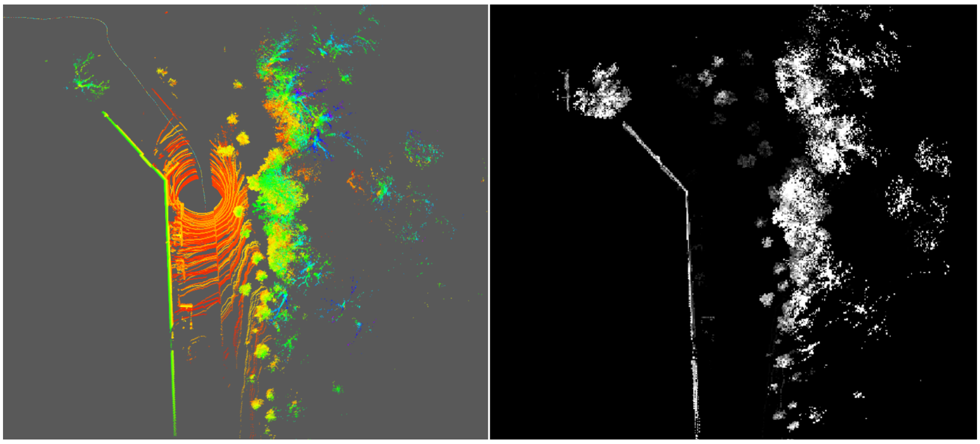


Figure 6: The left image shows a superimposed point cloud from ten Velodyne scans using NDT Fuser and the right image is the birds-view image of the superimposed point cloud by projecting the z values on the $x-y$ plane. We set a visual scope of 100x100x10 meters and resolution of 0.4 meters per pixel to generate a 400x400 gray-scale image for learning.

3 Reliability-aware long-term localisation and mapping

In T2.2 we have surveyed the background literature on long-term mapping and localisation [1]. We have further developed new approaches for long-term localisation and mapping. This includes a new 'front-end' for long-term mapping and localisation, which includes algorithms for detecting and tracking people in the environment, in collaboration with the EU-funded project FLOBOT [13], which can then be excluded from the 'back-end' optimisation.

3.1 Long-term localisation

Reliability-aware long-term localisation and mapping consists of two components: long-term localisation and persistent reliability-aware mapping. For long-term localisation, the robot should be able to self-localise with a priori knowledge of the environment when the "kidnapping" happens. For example, during long-term operation, robot localisation is likely to fail when the robot is switched off or moved by a human. Conventional Lidar-based global localisation methods either register the local Lidar scan with the global map or deploy Monte-Carlo Localisation (e. g. NDT-MCL as described in D1.2) initialised with uniformly distributed particles. However, these geometry-based methods are not scalable for large-scale applications. Even for small maps, it may take hundreds of seconds to converge to a correct pose estimate [14].

In ongoing work, we are experimenting with employing a deep neural network, instead of using geometry-based methods, to estimate the distribution of the robot's 6DOF global pose to initialise the particles in NDT-MCL. More specifically, we first superimpose ten frames of consecutive Velodyne 3D lidar scans using NDT Fuser (as described in D1.2), and encode the superimposed point cloud as a gray-scale birds-eye image as shown in Figure 6. Then a deep regression neural network is used to learn the 6DOF global pose, using ground-truth data from a previously recorded map. We use the first ten convolutional layers of the VGG-16 architecture as the base net and another two triple-layer fully-connected layers to learn the three-dimensional $t^p = (p_x, p_y, p_z)$ position and four-dimensional rotation (quaternion) $r^p = (q_x, q_y, q_z, q_w)$, respectively. We propose the

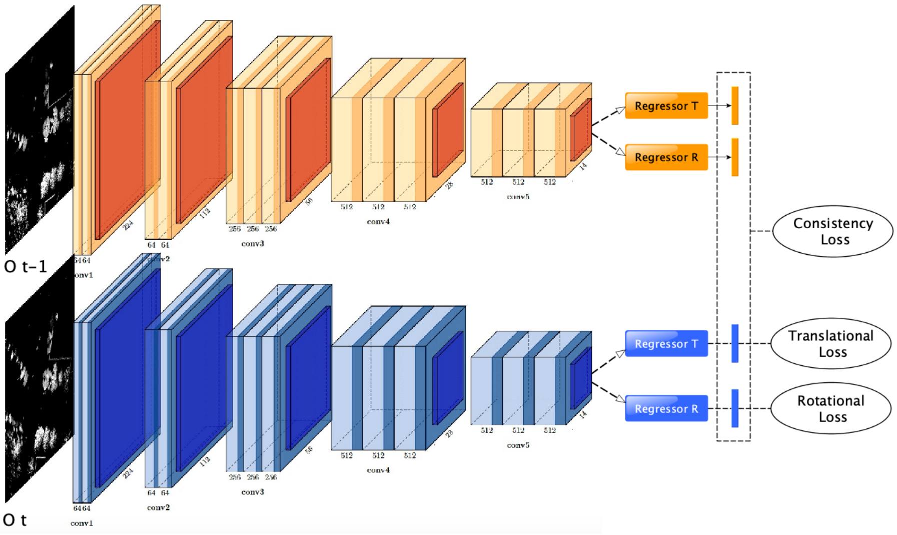


Figure 7: The deep learning architecture used for global localisation. In the mini-batch training, a batch of paired images is used. The weights of the pair of images are shared. Currently, we use the convolutional layers of the VGG-16 network architecture pretrained by ImageNet for dense feature extraction, and two multi-layer fully-connected regressors are used for predicting the global position and orientation.

following loss function for minimising the position loss and rotation loss.

$$\text{trans\_rot\_loss} = ||t^p - t^{gt}||_2 + \lambda\,(1 - < r^p, r^{gt} >^2) \tag{2}$$

Here, $< r^p, r^{gt} >$ is the inner product of the predicted quaternion and ground truth quaternion, and the second term indicates the distance between two normalized quaternion vectors.

Given a pair of images from time $t-1$ and t, the predicted global poses T^p_{t-1} and T^p_t, and the ground truth poses T^{gt}_{t-1} and T^{gt}_t, we can calculate the relative transform from the predictions and ground truth and make them geometrically consistent.

$$\text{consistency\_loss} = \text{trans\_rot\_loss}(\,(T^p_{t-1})^{-1}T^p_t,\ (T^{gt}_{t-1})^{-1}T^{gt}_t\,) \tag{3}$$

More specifically, T^*_t is the transform matrix which can be obtained from the translation t and rotation r at time t. We convert the transform matrices back to pose vectors to compute the translational and rotational losses. We find that with the assistance of geometric consistency loss, the neural network can learn temporally consistent features, thereby enhancing the robustness of global pose estimation.

The proposed deep neural network can learn site-specific and spatio-temporally consistent features. However, the inference (prediction) is not fully probabilistic with L2 loss. The drawback is that the uncertainty of the prediction cannot be provided. In other words, the neural network cannot give the confidence of the re-localisation. In our localisation pipeline, the confidence of global localisation is very important, which determines whether to initialise the particles and switch to Monte-Carlo localization. In order to mitigate this issue, we combine Gaussian process regression with the deep localisation network to estimate the robot global pose with uncertainties; that is, we apply a hybrid probabilistic regression technique where the deep neural network is used to extract features and build the kernel of a Gaussian Process.

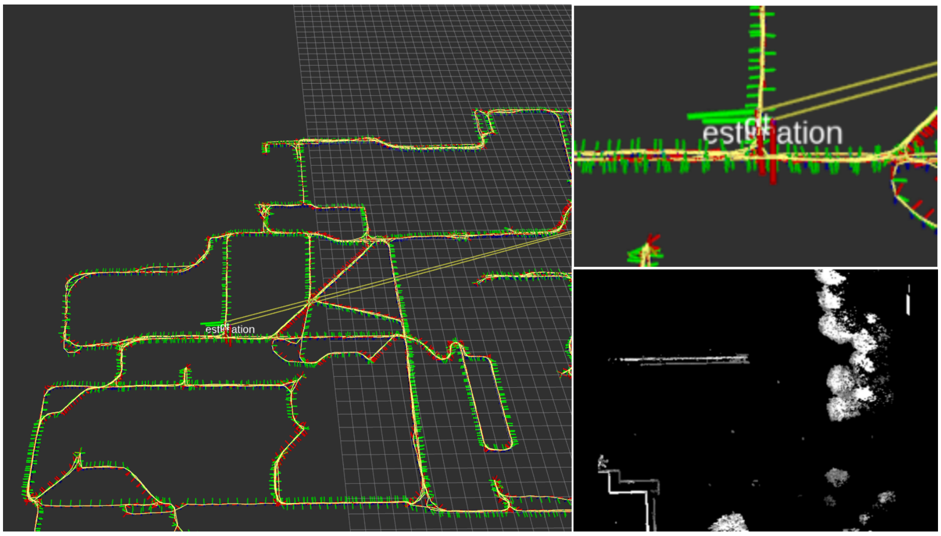


Figure 8: A qualitative result of global localization using our deep learning architecture. Given the bottom right image (created from a 3D lidar scan), the method provides an accurate localisation estimate quite close to the ground truth, as shown to the left (full map) and top right (zoomed in). The two green/red coordinate frames show the ground truth and estimated position, respectively. This example is from the Michigan NCLT dataset (http://robots.engin.umich.edu/nclt/). We use session 2012-01-08, 2012-01-14, 2012-01-22 for training and session 2012-05-11 for testing.

In the inference of Gaussian process regression, the conditional probability of the latent variable of testing example $f*$ given the training data $\{X, y\}$ is:

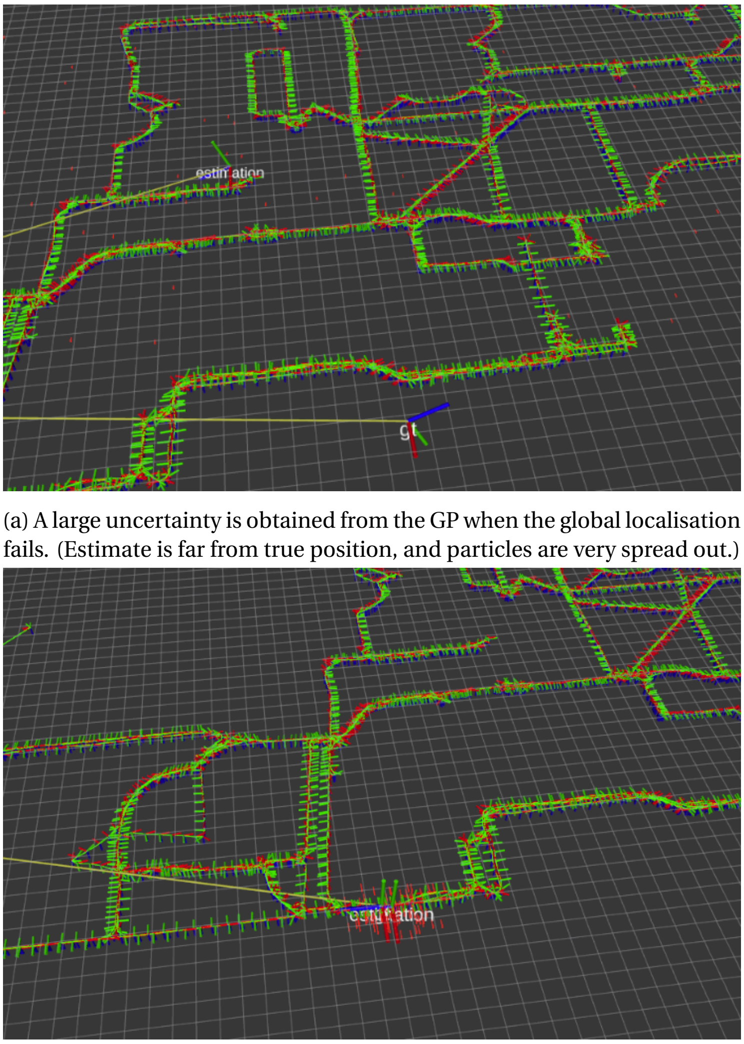
$$P(f*|X,y) = \mathcal{N}(K_{*n}(K_{nn}+\sigma^2 I)^{-1} y, K_{**} - K_{*n}(K_{nn}+\sigma^2 I)^{-1} K_{n*}) \tag{4}$$

This conventional inference formula is not scalable as the computation of $(K_{nn}+\sigma^2 I)^{-1}$ is $O(n^3)$. Instead of using all the training examples for the prior K_{nn}, we use a reduced set of points $Z \in \mathscr{R}^{m*F}$ (i. e., the inducing points) to approximate the whole training set, where m is much smaller than n. Given the latent variables of the inducing points, u, the posterior $p(u|y)$ can be estimated by a variational prior distribution $q(u)$, and then in order to get the optimal Z, the marginal log likelihood $p(y)$ is maximised via maximising the evidence lower bound (ELBO):

$$\begin{aligned} L(p(y)) = \int p(u)p(u|y)\,\mathrm{d}u = \int p(u)\ \mathrm{ELBO}(u)\,\mathrm{d}u \\ = \int p(u)E[log\ p(y|u)] - KL(q(u)||p(u))\ \mathrm{d}u, \end{aligned} \tag{5}$$

where KL refers to the Kullback–Leibler divergence. Titsias et al. [15] prove the final formula of the optimal inducing points Z and the mean and covariance of $q(u)$. In order to further make the training scalable, we train the SVIGP [16] (Stochastic Variational Inference Gaussian Process) from mini-batch data.

We use an RBF kernel in the Gaussian process and the kernel is constructed from deep neural network features. To be more specific, we only use the Gaussian process for the position (translation) prediction, and the feature of the last layer (shown in Figure 7)



(a) A large uncertainty is obtained from the GP when the global localisation fails. (Estimate is far from true position, and particles are very spread out.)

(b) A small uncertainty is obtained from GP when the global localisation succeeds. (Estimate is close to true position, particles are concentrated around the estimate.)

Figure 9: A qualitative result of Gaussian Process localisation with uncertainties. Particles for initialising the MCL filter are shown with small red arrows. The ground truth pose and the one estimated by our network are shown with two labelled coordinate frames.

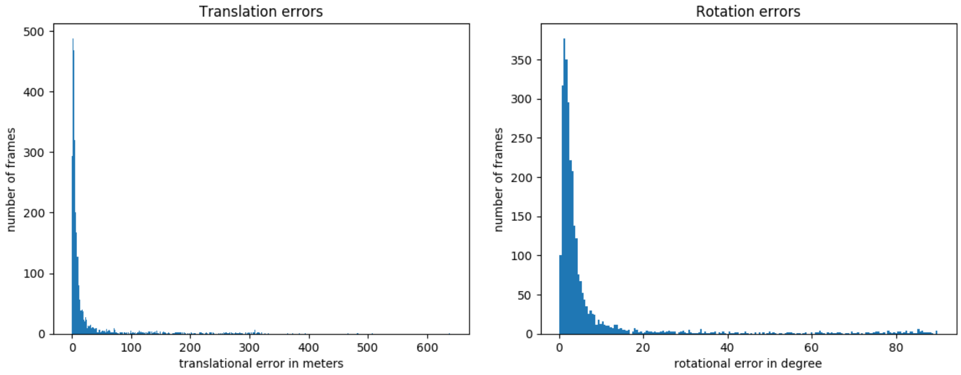


Figure 10: Histograms of translational and rotational error of the output from the global localisation network. We have found that most of the translation errors are smaller than 20 metres and most rotation errors are smaller than 10 degrees. The median is 3.5 metres and 2.3 degrees.

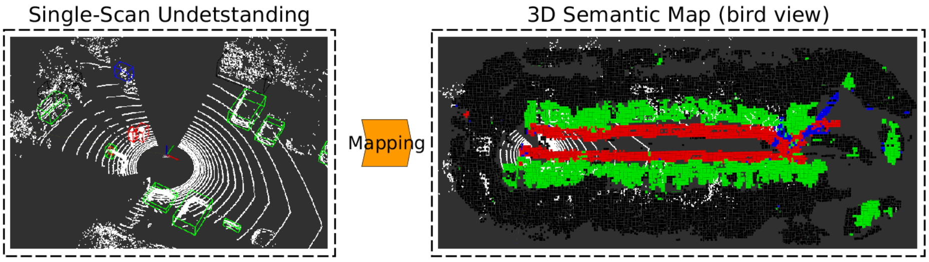
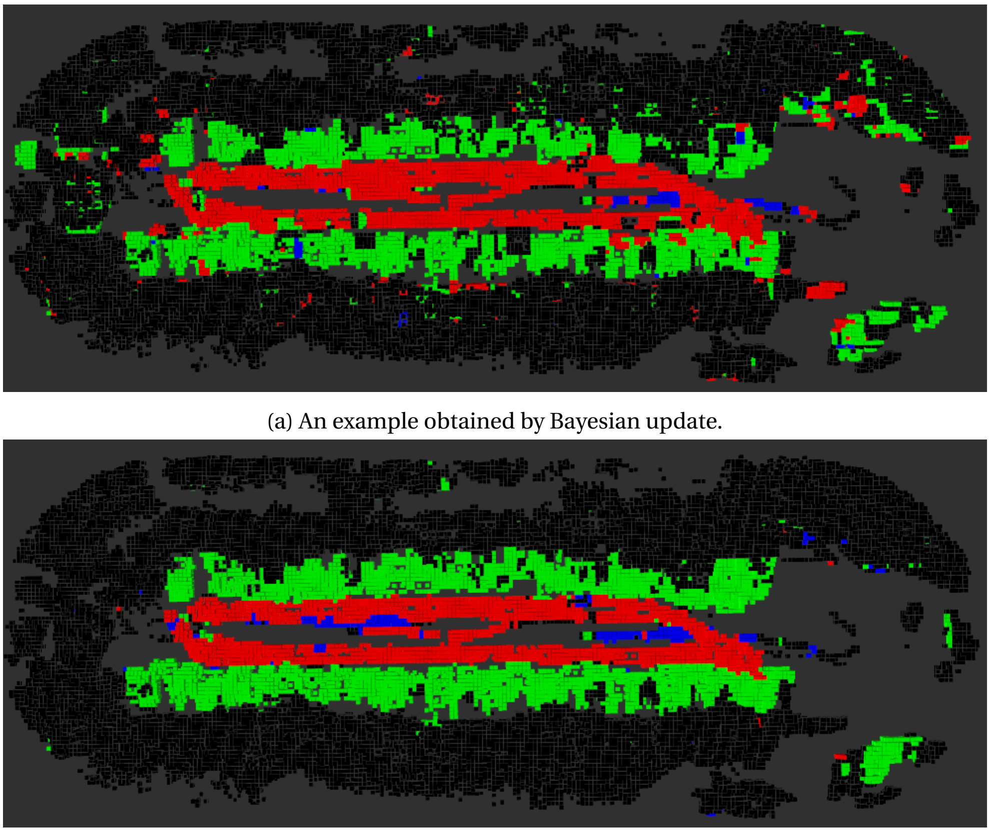


Figure 11: The proposed Recurrent-OctoMap is able to fuse a single-scan semantic understanding into a global 3D map.

is used. By this means, the parametric neural network can be integrated with the non-parametric Gaussian Process.

Since the creation of an ILIAD-specific long-term data set from warehouses has been delayed, the Michigan long-term dataset (http://robots.engin.umich.edu/nclt/) has been used for training and evaluation of the preliminary results shown in Figures 8 and 10. Here, we have used the data collected in January 2012 for training and that in May 2012 for evaluation.

Our ongoing research is investigating the integration of the learning method with the conventional geometry-based method, i. e. Monte-Carlo Localisation. We proposed to use our learning method to estimate the distribution of the robot's global pose, and thereby to initialise particle filters. Currently our global localisation network achieves a median translation error of 3.5 metres and a median translation error of 2.3 degrees for the Michigan data set (covering approximately 1 km^2). This is an encouraging result, and will likely result in fast convergence of an MCL filter for global "kidnapped robot" localisation. A publication on these results is planned for Autumn 2019.



(a) An example obtained by Bayesian update.

(b) An example obtained by Recurrent-Octomap

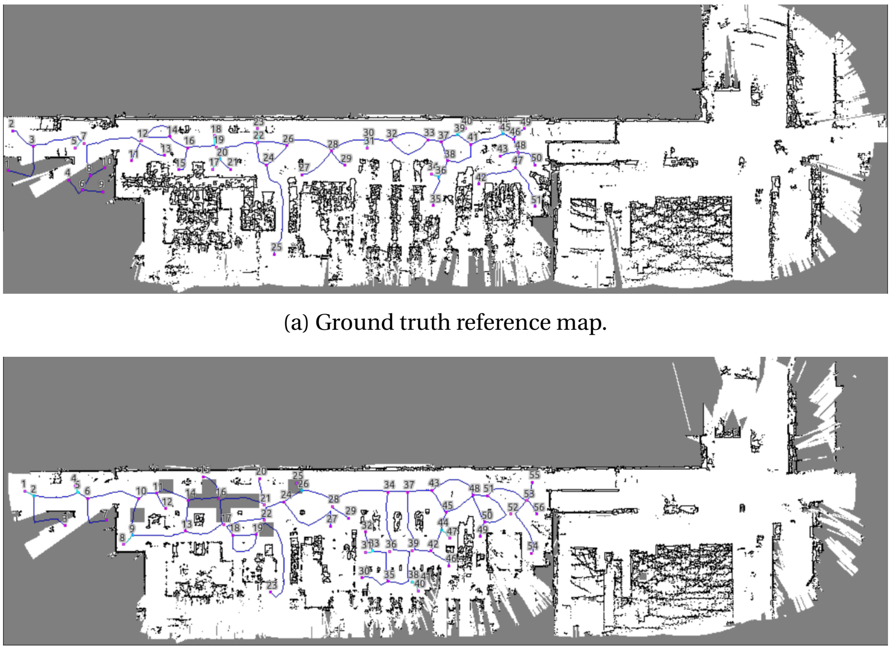
Figure 12: A qualitative result of semantic mapping on the ETH parking-lot dataset.

3.2 Reliability-aware long-term mapping

3.2.1 Long-term semantic map updating

We have developed a novel approach for long-term 3D semantic mapping based on a deep-learned Recurrent-OctoMap representation [9]. The most widely-used approach for 3D semantic map refinement is a Bayesian update, which fuses the consecutive predictive probabilities following a Markov-Chain model. In contrast, this research proposes a learning approach to fuse the semantic features, rather than simply fusing predictions from a classifier. In this research, we represent and maintain our 3D map as an OctoMap, and model each cell as a recurrent neural network (RNN), to obtain a Recurrent-OctoMap. In this case, the semantic mapping process can be formulated as a sequence-to-sequence encoding-decoding problem. It is worth noting that here we use a multi-layer perceptron instead of CNN, and Gaussian processes are not used in this research. Moreover, in order to extend the duration of observations in our Recurrent-OctoMap, we developed a robust 3D localisation and mapping system for successively mapping a dynamic environment using more than two weeks of data, and the system can be trained and deployed with arbitrary memory length.

We have validated our approach on the ETH long-term 3D lidar dataset. The experimental results show that our proposed approach outperforms the conventional Bayesian update approach with 16 % better overall accuracy for this dataset [9]. (See Figures 11 and 12.)



(a) Ground truth reference map.

(b) Map to be evaluated (constructed without reference localisation).

Figure 13: Map quality evaluation, showing the two maps to be compared, and their extracted topology graphs.

3.2.2 Reliability-aware mapping by map self-assessment

Per-map quality assessment T2.2 also contains work on quantifiable measures of map quality to understand where maps are "broken" or out of date. In the third and fourth years of ILIAD, we will develop such measures that are suitable also for industrial use-cases such as warehouses.

Work up until now has focused on evaluating state-of-the-art measures for map quality on warehouse data from ILIAD use cases. In particular, we have implemented and evaluated the metrics of Schwertfeger [17] (as also specified in the "technical requirements and performance measures" annex to D7.1). These metrics hinge on comparing the map in question to a ground truth reference map. The comparison relies on extracting the structure of the environment and representing it as a graph. In this deliverable we present three of Schwertfeger's metrics to evaluate the quality of the map: *coverage*, *accuracy*, and *local consistencies*.

The method relies on comparison of topology graphs based on the Voronoi diagrams of two 2D occupancy maps, one of which is the reference map. Figure 13 shows two maps from a dairy production warehouse: one produced using ground-truth localisation from Kollmorgen Automation, and one using the NDT Fuser mapping framework [18], which will be compared to the reference map.

There are some notable issues with Schwertfeger's approach for comparing maps that are visible in Figure 13. Firstly, the method is not able to fully handle an environment that is not fully connected (which is typically the case in warehouses) and is only partially able to extract a topology graph from the maps in this example. No graph is computed for the right room, on the other side of a door. Furthermore, some of the metrics are sensitive to small perturbations and noise in the maps, and do not necessarily reflect low-quality maps, as detailed below.

The *coverage* measure is computed as the ratio between matched nodes in the reference graph to the total number of nodes in it. In this example, the coverage measure is 58 %. That is caused not by low quality of the map, but by the fact that even small changes in the shape of the environment substantially change the resulting graphs. Furthermore, the graph only covers part of the environment, which substantially limits the quality of the evaluation.

The *accuracy* score is computed as the mean squared error of the position of the matched vertices. In this example, the evaluated map and the ground truth map are built from the same data set. Therefore both maps are aligned so that the *global* and *relative* accuracy are equal. The accuracy score is computed as the ratio of the mean squared error to a configured maximum error $d_{\max}^{\text{GlobalAccuracy}}$. Here the problem is that an arbitrarily chosen maximum error makes it impossible to objectively compare the quality of two maps. The mean squared error of locations of the nodes is equal to 81 pixels squared in the example from Figure 13. As $d_{\max}^{\text{GlobalAccuracy}}$ we have selected $(2w)^2$, with w the width of the longest wall; in this case $d_{\max}^{\text{GlobalAccuracy}} = 169$. This gives us a local and global accuracy of 47 %.

We have also attempted to compute the *local consistencies* measure. This metric is computed in a similar way as the relative accuracy, but the distance of one pair of connected nodes in V is compared to the corresponding inter-node distance in V'. However, since the two graphs are not equal, due to some clutter, the distance is computed only for the paired nodes, which in turn means that the average score for all of the evaluated nodes is very low. The consistency score for this pair of maps is 28 %, which is far from our target value of > 80 %.

In summary, computing map quality for this warehouse example using Schwertfeger's method, which we consider to be the closest thing to an established state of the art for map quality assessment, does not yield results that are consistent with intuition for our warehouse data sets. The map in Figure 13b looks quite similar to its ground truth reference in Figure 13a but still the coverage and accuracy are both less than 60 %, which is rather far from the ILIAD target values that were set out to be > 80 % coverage and > 95 % relative accuracy. However, the low numbers seem to be caused by the nature of the evaluation algorithm, rather then by the poor quality of the map.

In ongoing work, we are instead developing alternative methods for self-assessment of map quality, which furthermore are designed to work without a ground-truth reference map. Being able to estimate the quality of a robot-generated map without comparison to a reference map is important both for easy deployment and low-maintenance during long-term operation. This line of work will be presented in more detail in D2.2, but the general approach is to learn statistical relations between elements of the map and identify abnormal elements. Our approach is similar in spirit to that of Chandran-Ramesh and Newman [19] but we will study the use of unsupervised methods in order to better address ILIAD's objectives of easy deployment and long-term introspective monitoring.

Per-scan quality assessment While the work above pertains to quantifying map quality and identifying problem cases globally on the map level, we have also published work on quantifying the quality of scan registration, which can be used while the map is being built via scan-to-scan or scan-to-model registration [20]. We have performed a comparative evaluation of a number of methods for geometric consistency checking from the literature, in order to classify aligned versus misaligned 3D point cloud scans. In addition, we have trained an AdaBoost classifier from a combination of these classifiers. We have compared these methods on two data sets from qualitatively different environments in order to avoid overfitting the classifiers to any particular environment; one structured environment where building walls are the dominant features, and one unstructured

Name	Description
NDT1	NDT score
NDT2	NDT score with ground removed
NDT3	NDT score, only overlapping regions
NDT4	NDT score, with ground removed and only overlapping regions
RMS1	Root-mean-square point-to-point distance, with nearest-neighbour threshold 4 m
RMS2	RMS distance, threshold 2 m
RMS3	RMS distance, threshold 0.5 m
RMS4	RMS distance, threshold 0.25 m
RMS5	RMS distance, threshold 0.15 m
RMS6	RMS distance, threshold 0.05 m
RMS7	RMS distance, statistical threshold [22]
HEST	NDT Hessian translation parameters
HESR	NDT Hessian rotation parameters
PLEX	Plane extraction [23, 19]
NORM	Partitioned mean normals [24]
SIM1	Surface interpenetration measure [21]
SIM2	Surface interpenetration measure with overlap scaling
ADA	AdaBoost, combining the above classifiers

Table 1: The scan alignment methods evaluated in Figure 14.

outdoor environment. We also evaluated how well the methods generalise by training in one environment and testing in the other.

Figure 14 shows the accuracy of all the methods for the structured environment. The methods that have been evaluated are listed in Table 1. From these results, it is clear that using the NDT score as a classifier for detecting misaligned scans is a viable option, especially when only computing the score function for points in overlapping segments of the two point clouds (NDT3 and NDT4). This is true even when the classification threshold is learned from a very different environment. Combining several classifiers with AdaBoost did not lead to significantly better results. The best classifiers (NDT score and AdaBoost) detect close to 100 % of the large errors (50 cm) and 90 % of the small errors (10 cm) when trained on structured data, and 80 % of small errors when trained on an unstructured data set. The RMS (root mean squared) point-to-point error is not good for detecting small errors, and other methods such as the surface interpenetration measure (SIM) [21] do not generalise from one environment to the other.

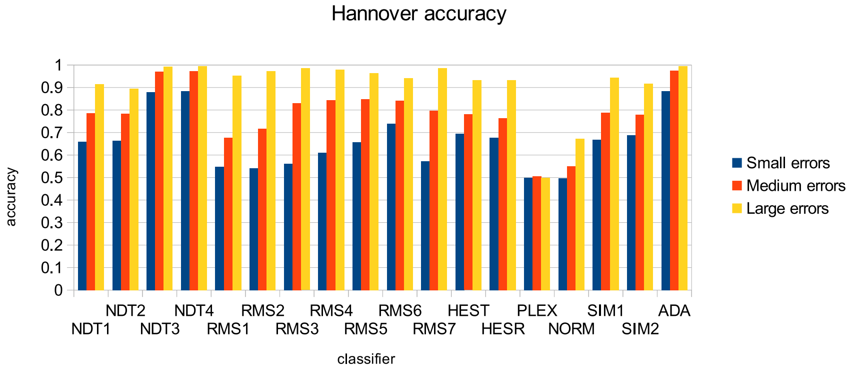
4 Learning and Predicting Site-Specific Activity Patterns

In T2.3 we have applied our spatio-temporal flow maps CLiFF-map and STeF-map to data from a human tracking system to model site-specific activity patterns [5, 6]. We have also applied deep-learned neural networks for predicting human motion patterns [8].

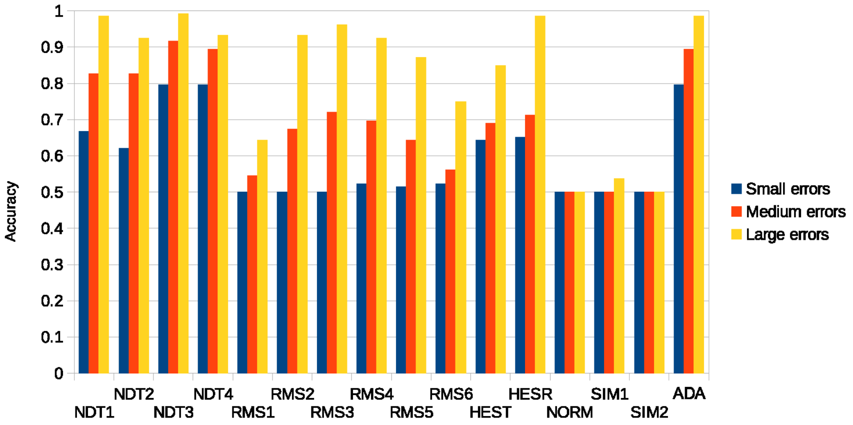
4.1 Learning activity patterns with CLiFF-map

Figure 15 shows an example CLiFF-map created from a people-tracking data set recorded at NCFM (the National Centre for Food Manufacturing in Holbeach, UK) during ILIAD's Milestone 2 demonstration in September 2018.

We have also published research showing how the CLiFF-map representation can be



(a) Accuracy on the structured Hannover data set (8-fold cross validation).



(b) Accuracy when trained on the unstructured Kjula data set and evaluated on the structured Hanover data set.

Figure 14: Accuracy of scan alignment classifiers for three ranges of error offsets. The best classifiers (NDT score and AdaBoost) detect close to 100 % of the large errors and 90 % of the small errors when trained on structured data, and 80 % of small errors when trained on an unstructured data set.

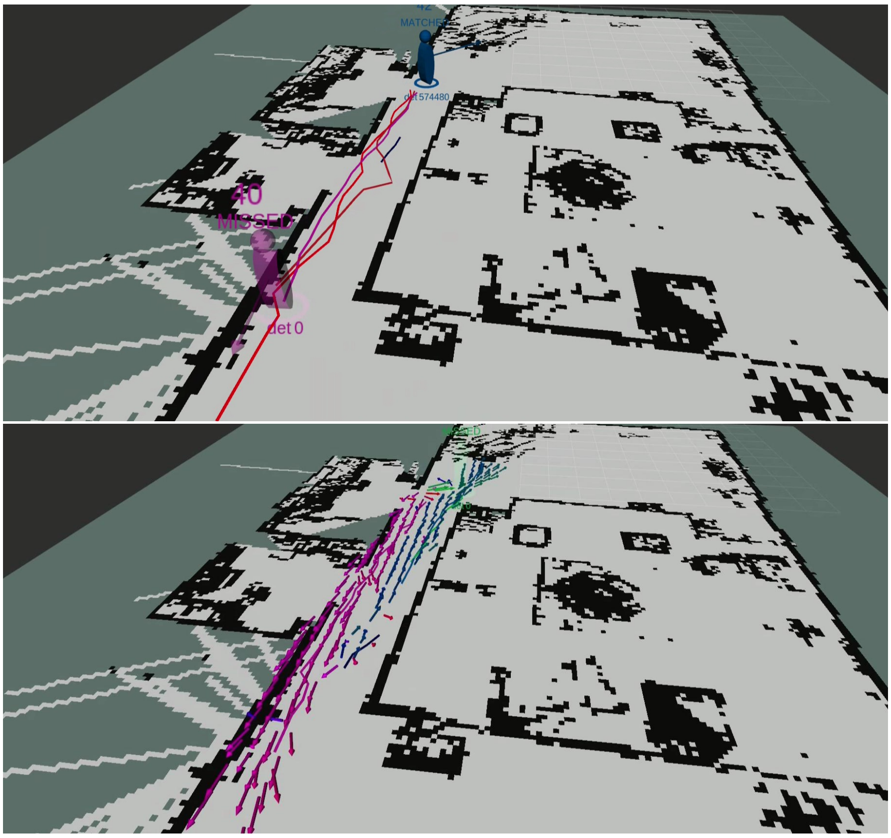


Figure 15: Example illustrating the generation of a CLiFF-map, taken from the MS2 demonstration at NCFM in September 2018. The top figure shows tracked pedestrians (from Task 3.3), and the bottom figure illustrates the modes of the CLiFF-map that was learned from long-term observation of pedestrian flows through this corridor. Each arrow in the bottom figure represents the mode of one GMM component. (The variances and mixing factors of the GMMs are not visualised, for clarity.)

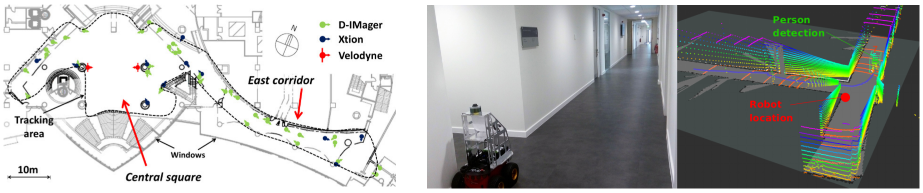
exploited by RRT-style motion planning [25, 26]. Qualitatively, using CLiFF-maps has been shown to enable robots to follow or avoid expected flows of people, depending on the application. Quantitatively, we have shown that by planning on a CLiFF-map, the planner finds a path quicker than when planning on a geometric-only map. These results will be reported under WP5 (in D5.3).

4.2 Learning and predicting activity patterns with STeF-map

So far, in order to evaluate the STeF-maps approach, we ran experiments using two real pedestrian datasets. Both feature complex human movement and enough days to train the models and perform the evaluation from a long-term perspective.

The first one is a pedestrian tracking dataset recorded at the **ATC** shopping center in Osaka (Japan), which covers an area of around 1000 m^2 (Figure 16a). From this environment we used 52 consecutive days (26 Wednesdays and 26 Sundays), taking the first 46 to perform exploration and the other 6 days as the evaluation data.

The second dataset was collected by a robot at one of corridors in the Isaac Newton Building building at the University of Lincoln (**UoL**) as shown in Figure 16b. The robot



(a) (b)

Figure 16: (a) ATC shopping center map - (b) UoL dataset: Robot location in the corridor and example of a person walking seen by the Velodyne scans.

was placed in one of the T-shaped junctions covering a total area of around 75 m^2 (see Figure 16b). Our dataset spanned from mornings to late evenings for 14 days, sparsely recorded over a four week period. From these, 12 days were used for training and the remaining two were used for testing.

Learning the Activity Patterns To do the learning we assume that the model is provided with people detections from the environment, containing the x, y coordinates together with the angle of movement α for every timestamp t. These x, y positions for each detection are discretised and assigned to a corresponding cell, and the orientation α is assigned to one of the k bins, whose value is incremented by 1. In other words, we count the number of people detections occurring in each orientation bin and cell. After a predefined interval of time, we normalise the bins, and use the normalised values to update the spectra of the temporal models (FreMEn models). Then, we reset the bin values to 0 and start the counting again.

Predicting of the Site-Specific Activities To predict the behaviour of human movement through a cell at a time t, we calculate the probability for each discretised orientation θ, ($\theta = i\frac{2\pi}{k}$ and $i \in \{0 \ldots k-1\}$), associated to that cell as

$$p_\theta(t) = p_0 + \sum_{j=1}^{m} p_j \cos(\omega_j t + \varphi_j), \tag{6}$$

where p_0 is the stationary probability, m is the number of the most prominent spectral components, and p_j, ω_j and φ_j are their amplitudes, periods and phases. The spectral components ω_j are drawn from a set of ω_s that covers periodicities ranging from 1 to 24 hours with the following distribution:

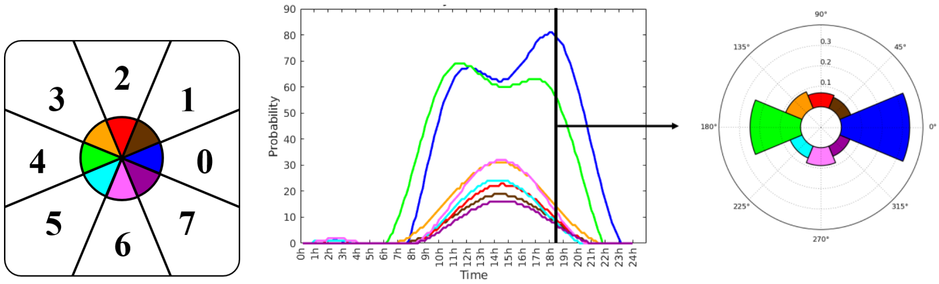
$$\omega_s = s \cdot 3600, \quad s \in 0, 1, 2, 3, \ldots, 24. \tag{7}$$

Computing the probability at multiple times we can obtain the time evolution of the people behaviour going through a certain cell, as shown in Fig. 20b.

4.3 Learning and predicting activity patterns with Warped Hypertime

We have validated the Warped Hypertime approach on 2-dimensional data indicating the positions of people in several corridors of the Isaac Newton Building at the University of Lincoln, using the UoL dataset described above.

To valuate the model quality, we split the gathered data into training and test sets, and learn the model from the training set only. Then, we partition the timeline of the test



(a)

(b)

Figure 17: (a) 8 bins discretising every 45 degrees the full circumference. (b) Model prediction over 24h with $m = 2$ and probability distribution of each orientation at t =18:00 of one cell.

data into a spatio-temporal 3D grid. For each cell g, we count the number of detections d_g that occurred and compare this value with the value p_g predicted by a given spatio-temporal model. To demonstrate the model's ability to estimate the spatio-temporal distribution over time, we let it predict the most likely occurrence of people for different times. Figure 18 shows that the predicted distributions of people depend on time and follow the shape of the corridor (which is not part of the training data).

In ongoing work we are evaluating and comparing this approach alongside the other reported spatio-temporal representations for learning and prediction in the ILIAD system.

4.4 Pedestrian trajectory prediction using deep learning

We have also used the sequence-to-sequence encoder-decoder architecture from T2.1 (Figure 5) to learn site-specific pedestrian pedestrian trajectories from long-term robot deployment data, as part of our work in T2.3. In the pedestrian trajectory prediction problem, the observations are global poses of the detected pedestrians. Hence we do not use the CNN for feature encoding, and GP is not used in this application. It is worth noting that we include the current time and day of the week (assuming a weekly periodicity in this case) as another input feature of the neural network for time- and site-specific trajectory prediction. An example of the 3DOF pose prediction is shown in Figure 19 and more quantitative results can be found in our paper [8].

5 Ongoing and Future Work

Ongoing work in Work Package 2 will enable introspective active learning, by continuously monitoring the acquired spatio-temporal models, making an appropriate assessment of the precision and uncertainty in these models, and then taking actions to resolve the uncertainty and remove the errors. This approach will also be used in turn to keep the maps up-to-date and the robots successfully localised despite changes to the environment structure and dynamics over time.

Furthermore, in order to evaluate and test the generalisation of the multiple spatio-temporal models presented in this section, new data collection is been carried out in scenarios that are similar to the ones the ILIAD robots will face in the final implementation. Those are the NCFM and Orkla warehouse environments (Figure 20), where recordings are being performed using both the pallets trucks and the forklifts.

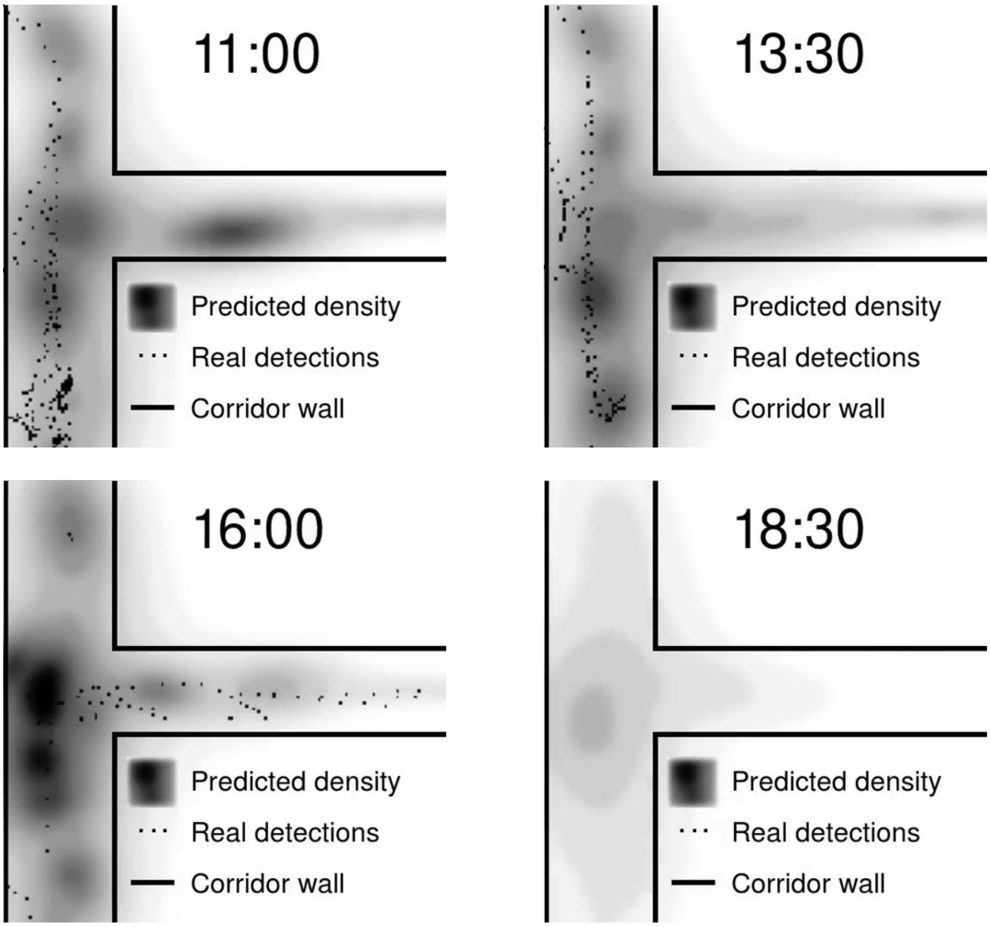


Figure 18: Predicted people presence in a corridor of the University of Lincoln, UK, at various times of the day using the learned representation in Figure 4.

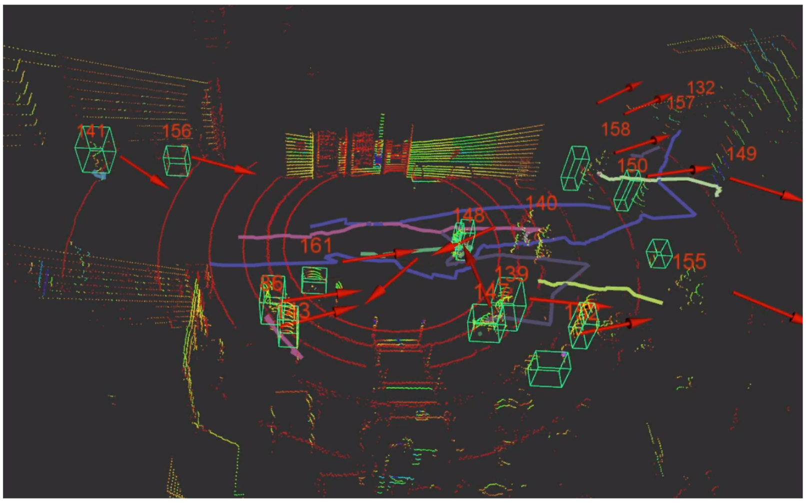
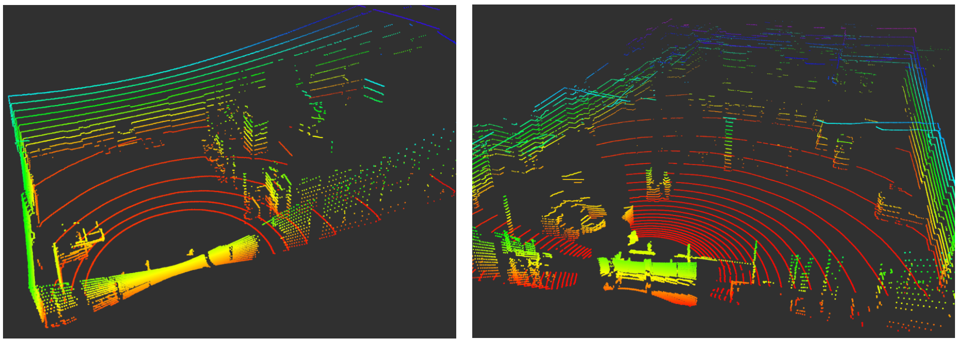


Figure 19: A screen-shot of our 3DOF pedestrian trajectory prediction in a 3D lidar scan. The detected people are enclosed in green bounding boxes with a unique ID. The coloured lines represent the observed people trajectories. The red arrows indicate the predicted poses for the next 1.2 s.



(a) NCFM, Holbeach, UK.

(b) Orkla Foods, Örebro, Sweden.

Figure 20: Environments currently used for long-term data collection.

References

[1] Lars Kunze, Nick Hawes, Tom Duckett, Marc Hanheide, and Tomas Krajnik. Artificial intelligence for long-term robot autonomy: A survey. *IEEE Robotics and Automation Letters*, 3(4):4023–4030, October 2018.

[2] L. Kunze, N. Hawes, T. Duckett, and M. Hanheide. Introduction to the special issue on AI for long-term autonomy. *IEEE Robotics and Automation Letters*, 3(4):4431–4434, October 2018.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[4] Tomasz Piotr Kucner, Martin Magnusson, Erik Schaffernicht, Victor Hernandez Bennetts, and Achim J. Lilienthal. Enabling flow awareness for mobile robots in partially observable environments. *IEEE Robotics and Automation Letters*, 2(2):1093–1100, April 2017.

[5] Sergi Molina, Grzegorz Cielniak, Tomas Krajnik, and Tom Duckett. Modelling and predicting rhythmic flow patterns in dynamic environments. In *UK-RAS Network Conference*, December 2017.

[6] Sergi Molina, Grzegorz Cielniak, Tomáš Krajník, and Tom Duckett. Modelling and predicting rhythmic flow patterns in dynamic environments. In *TAROS*, pages 135–146, 2018.

[7] Tomas Vintr, Sergi Molina Mellado, Grzegorz Cielniak, Tom Duckett, and Tomas Krajnik. Spatiotemporal models for motion planning in human populated environments. In *Student Conference on Planning in Artificial Intelligence and Robotics (PAIR)*. Czech Technical University in Prague, Faculty of Electrical Engineering, September 2017.

[8] Li Sun, Zhi Yan, Sergi Molina, Marc Hanheide, and Tom Duckett. 3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018.

[9] Li Sun, Zhi Yan, Anestis Zaganidis, Cheng Zhao, and Tom Duckett. Recurrent-OctoMap: Learning state-based map refinement for long-term semantic mapping with 3-d-lidar data. *IEEE Robotics and Automation Letters*, 3(4):3749–3756, October 2018.

[10] Tomasz Piotr Kucner. *Probabilistic Mapping of Spatial Motion Patterns for Mobile Robots*. PhD thesis, Örebro University, 2018.

[11] Tomas Krajnik, Jaime P. Fentanes, Joao M. Santos, and Tom Duckett. FreMEn: Frequency map enhancement for long-term mobile robot autonomy in changing environments. *IEEE Transactions on Robotics*, 33(4):964–977, August 2017.

[12] Tomás Krajník, Tomás Vintr, Sergi Molina Mellado, Jaime Pulido Fentanes, Grzegorz Cielniak, and Tom Duckett. Warped hypertime representations for long-term autonomy of mobile robots. *ArXiv (to appear in IEEE Robotics and Automation Letters with presentation at IROS'19)*, abs/1810.04285, 2018.

[13] Zhi Yan, Li Sun, Tom Duckett, and Nicola Bellotto. Multisensor online transfer learning for 3d lidar-based human detection with a mobile robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, October 2018.

[14] 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.

[15] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

[16] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

[17] Sören Schwertfeger. *Robotic mapping in the real world: Performance evaluation and system integration.* PhD thesis, Jacobs University Bremen, 2012.

[18] 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.

[19] M. Chandran-Ramesh and P. Newman. Assessing map quality using conditional random fields. In *Field and Service Robotics*, pages 35–48, 2008.

[20] Håkan Almqvist, Martin Magnusson, Tomasz Piotr Kucner, and Achim J. Lilienthal. Learning to detect misaligned point clouds. *Journal of Field Robotics*, 35(5):662–677, August 2018.

[21] Luciano Silva, Olga R.P. Bellon, and Kim L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Robotics*, 27(5):762–776, May 2005.

[22] Szymon Marek Rusinkiewicz. Efficient variants of the ICP algorithm. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[23] J. Weingarten, G. Gruener, and R. Siegwart. A fast and robust 3D feature extraction algorithm for structured environment reconstruction. In *Proceedings of the International Conference on Advanced Robotics*, 2003.

[24] Ameesh Makadia, Alexander Patterson, and Kostas Daniilidis. Fully automatic registration of 3D point clouds. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[25] Luigi Palmieri, Tomasz Kucner, Martin Magnusson, Achim J. Lilienthal, and Kai O. Arras. Kinodynamic motion planning on Gaussian mixture fields. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6176–6181, May 2017.

[26] Chittaranjan Srinivas Swaminathan, Tomasz Piotr Kucner, Martin Magnusson, Luigi Palmieri, and Achim Lilienthal. Down the CLiFF: Flow-aware trajectory planning under motion pattern uncertainty. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 7403–7409, 2018.