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

H2020-ICT-2016-2017
Grant agreement no: 732737

DELIVERABLE 1.3
Implementation of unsupervised semantic mapping system

Due date: month 48 (December 2020)
Deliverable type: R
Lead beneficiary: ORU

Dissemination Level: PUBLIC

Main author: Martin Magnusson (ORU)

Contents

1 Introduction

This report outlines an implementation of unsupervised semantic mapping that is specifically targeted for the main use case that drives the ILIAD project: fleets of intralogistics robots that are safe and efficient in spaces shared with people. The implementation is capable of automatically labelling shelves (pick-slots), non-traversable areas, and suggested places for temporary pallet storage. In all cases the system works without supervised learning. Shelf detection is treated as a geometric problem, and we show that it can be done so with high accuracy ($F_1 = 0.98$); traversability mapping that goes beyond simple occupancy projection uses frequency-based structure detection to better filter out clutter from the map; and we use long-term learned patterns of human motion to introduce non-binary traversability costs, and also suggest places for temporary pallet storage. In addition, we present a method for unsupervised place categorisation in a warehouse context. This method clusters 3D point clouds by appearance, and is able to automatically segment an unordered set of point clouds into semantically meaningful places, such as in-aisle or not, or different halls in the warehouse. We have shown that the performance on an available benchmark dataset is close to that of previously published supervised methods that use both range and intensity data.

We include data from three warehouses in the evaluation: primarily food manufacturer Orkla Foods, where the project's final demo is scheduled, and also grocery retailer Coop and dairy distributor Arla.

Our main method for structural mapping (the output of which is used in semantic mapping) is described in Section 2, and the semantic mapping components are covered in Section 3–6.

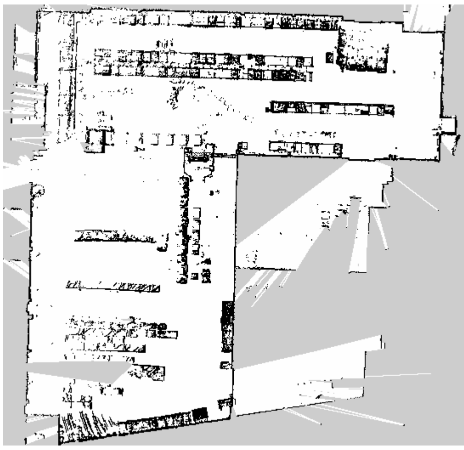
2 Structural mapping

The mapping system is based on the NDT-OM (normal-distributions transform occupancy map) representation [7], using the D2D-NDTSC scan registration method [1] to robustly estimate the lidar odometry. An example can be seen in Figure 1a. When the origin of the latest esimated sensor position $\mathbf{p}_t$ is located at a distance $> \delta$ m from the previous keyframe $\mathbf{p}_k$, the relative pose $\mathbf{p}_t^k = \mathbf{p}_k^{-1}\mathbf{p}_t$ is passed into a pose graph framework together with the matching covariance $\mathbf{\Sigma}_t^k$. A new keyframe $\mathbf{p}_k$ is then created from $\mathbf{p}_t$. This ensures low drift in the lidar odometry and keeps the number of scans in the pose graph to a minimum level. As drift is inevitably introduced when the robot traverses the environment, loop closure candidates are proposed when the current pose $\mathbf{p}_t$ is close a previouosly mapped area with a keyframe $\mathbf{p}_l$. Proposed candidates are registered using D2D-NDTSC in a coarse-to-fine fashion and the relative pose estimate $\mathbf{p}_t^l$ and covariance $\mathbf{\Sigma}_t^l$ are added as additional constraints in the pose graph. The optimal scan locations $\mathbf{x}$ given the constraints C can be obtained by solving $\operatorname{argmin}_{\mathbf{x}} \mathbf{F}(\mathbf{x}) = \sum_{\langle i,j\rangle \in C} \mathbf{e}_{ij}^T \mathbf{\Sigma}_{ij}^{-1} \mathbf{e}_{ij}$.

Using the refined lidar pose estimates and point cloud data, navigation maps can be built by classifying voxels in an interval $h_{\min} < h < h_{\max}$ above the ground plane as occupied or non-occupied. (The height of the scanner used to determine $h_{\min}$ can be taken from automatic calibration, T1.1.) The probability of a voxel being occupied is computed by performing raytracing between lidar keyframes and their observations, within the selected height interval, and applying an inverse sensor model to intermediate voxels. Voxels with sufficiently high probability of being occupied are then projected onto a standard 2d occupancy map as seen in Figure 1b.

The geometrical representation described above does not yet include higher-level information needed for actual fleet deployment; most importantly, where the pick-slots are (at what shelf in the warehouse is a particular product stored), and where activities

(a) A 3D point cloud map (black) and trajectory (red) estimated from lidar data in Orkla Foods' warehouse.



(b) Initial 2D occupancy map created by performing raytracing above the ground plane.

Figure 1: Output of structural mapping (see D1.2 and Section 2).

tend to happen (so that the robot can plan safely and efficiently without disturbing other's activities).

The 3D point cloud map which is the output after D2D-NDTSC scan registration and loop closure is used as the input for shelf detection (Section 3) and the 2D occupancy map described above is input to further analysis about traversability and storage in Sections 4 and 5. Place categorisation (Section 6) uses individual 3D point clouds without registration.

3 Unsupervised shelf detection

3.1 Method

To find candidate shelves in the map, we take the following approach. The input is a 3D pointcloud map $\mathscr{X}$ of the warehouse; e. g., as depicted in Figure 1a. In essence, the method searches for the poles of the racks used for the warehouse shelving. The poles are clustered into segments, one for each lane of shelves, after which the pick-slots that are the output of the method are distributed along the shelves. The detected pick-slots are included in the semantic map, and in a post-processing step, warehouse staff can assess the results to remove falsely detected shelves and add missing ones.

First, the map is pre-filtered to remove the floor, by filtering those points that are lower than h_{min}. The point cloud is segmented using region-growing segmentation [2] to generate an initial set of "obstacle chunks".

After this, the points of each segment are classified using the local shape, as follows.

1. For each remaining point $\mathbf{x} \in \mathscr{X}$, find all neighbouring points within a certain radius r.

2. Compute the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of the point positions found in the previous step.

3. Compute the eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$ and eigenvectors $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ of $\mathbf{\Sigma}$. If $\lambda_2/\lambda_1 < t_l$, the surrounding surface shape is approximately distributed along a line. The direction if the linear distribution is the largest eigenvector $\mathbf{e}_3 = (x, y, z)$. In this application, we are looking for vertical lines (poles) that have $|z| > |2y|$ and $|z| > |2x|$. From these values, the class computed from $\mathbf{x}$ is either *line* or *pole*. (If $\lambda_1 \geq t_t$, the surface is considered *uneven*. If $\lambda_1 < t_t$, the surface is planar. The normal vector of the plane approximating the local surface is $\mathbf{e}_1$. Depending on the orientation of $\mathbf{e}_1$, assign the class *floor*, *wall*, or *slope*.)

4. A weighted vote for the class selected in the previous step is assigned to $\mathbf{x}$ and each of its neighbours. The weight is determined by a Gaussian kernel centred at $\mathbf{x}$ with variance $\sigma = r/3$, so that it is close to zero at the edges of the neighbourhood.

5. After all points in $\mathscr{X}$ have been evaluated, each point is assigned the class for which it has the highest vote.

After all points that are not classified as *pole* have been removed, the remaining points are again clustered using region-growing segmentation (using r as the Euclidean distance threshold). Clusters with very few points, less than a parameter m, are removed, as provide they little support for the presence of a pole.

The remaining pole clusters are segmented again, with a larger threshold w_a, in order to create one segment per rack of shelves. The parameter w_a is warehouse-specific and should be set according to the smallest aisle width.

Finally, this set of “rack segments” is filtered to remove segments that are too small or too large. Each rack segment should be no wider than two pallets. A common setup of shelf racks is to have access from two sides, which means that the rack is two pallets wide. A rack that is three or more pallets wide makes little sense (since the middle pallets could not be reached) and is therefore likely the result of undersegmentation; e. g., if w_a is set too large.

The remaining segments constitute a plausible set of the racks present in the warehouse. For each of these racks, pick-slots are placed at regular intervals along the length of the rack. An oriented bounding box is computed for the rack, and pallet-sized pick slots are distributed along the longer sides of the bounding box. The size of each pick slot is a user-defined parameter – typically corresponding to the size of a standard pallet.

3.2 Data sets

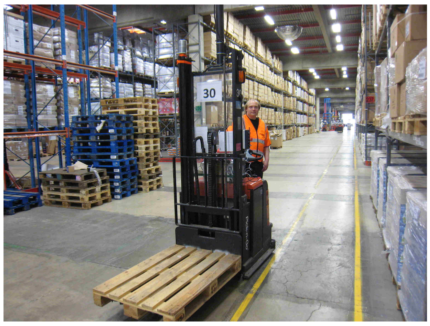
The methods are validated using real-world warehouse data from two sites: one from food manufacturer Orkla Foods, where the project’s final demo is scheduled, and one from grocery retailer Coop. The Orkla warehouse is smaller and less structured, while the Coop warehouse is large and highly structured. Figure 2 shows photos from the two sites.

The Orkla warehouse contains 105 pick-slots used in production plus 9 not used in production. In this warehouse, five lanes of shelf racks are placed in one area of the warehouse, adjacent to other structure such as machinery for pallet wrapping, conveyors from the factory, temporary pallet storage, etc. The warehouse also contains large open space used as marshalling lanes for outward shipping.

The Coop warehouse contains many hundreds of pick-slots with shelf racks aligned along a regular pattern of aisles. Our dataset only covers a part of this very large warehouse. Compared to the Orkla warehouse, the environment is much more regular, with the racks making up most of the structure.

The racks in both warehouses differ in size and configuration. For example, the vertical poles supporting the racks are spaced 3 m apart at Orkla, and 2 m apart at Coop. Aisles between racks are just over 2 m wide at Coop, but waries between 2–4 m at Orkla. The

(a) The Orkla warehouse. *Top:* racks of shelves, and additional structure from the warehouse. *Bottom:* Example of rack-like structures consituting true negatives.



(b) The Coop warehouse. Please note that the poles in the shelf racks are not always well visible; e. g., due to the number and size of packages stored on the shelves.

Figure 2: The two warehouse datasets used to evaluate automatic shelf detection.

common denominator between the racks where the pick-slots are located is simply that they consist of coherent sets of poles and shelves.

3.3 Results

The output of the method described above for the two warehouses can be seen in Figures 3 and 4.

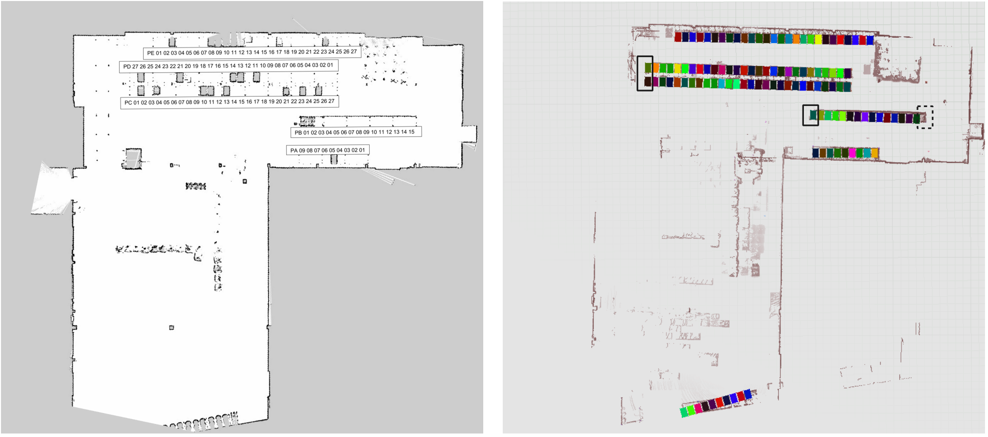
Orkla The number of pick-slots used in production in the Orkla warehouse is 105. A ground-truth labelling of pick-slots made by warehouse staff is shown in Figure 3a. Out of the 105 pick-slots, 104 are correctly detected by our method and there is 1 false negative. In addition, there are 3 false positives. This means that the recall rate is 99% and the precision is 97% ($F_1 = 0.98$), which is well above the target value of 80% recall at 80% precision set forth in the project's technical requirements.

The false positives are placed adjacent to two of the racks. At these positions, there are pole-like structures that are mistaken to be part of the shelf rack, causing 2 false positives for the double-rack and 1 false positive for a single-rack. These failure cases are labelled in Figure 3b.

In addition to the pick-slots counted above, the method detects another rack of shelves at one end of the warehouse. These are actual shelves, although they are not used in production and therefore not labelled by the warehouse staff. The method detects 10 pick-slots, of which 9 are true positives.

Coop Results from the Coop dataset are shown in Figure 4. One observation is that the method works best when the racks have been well observed: from both sides. In our quantitative results, we only count those racks that have been seen from both sides (inside the region thus marked in Figure 4).

In this dataset, we do not have ground truth labels by staff, but the number of pick-slots can be counted by visual inspection of the 3D map data. This should only be taken



(a) Ground-truth labelling of pick-slots. (b) Output of shelf detection.

Figure 3: Results of shelf detection from the Orkla dataset. *Left:* manual (ground-truth) labelling of the pick-slots used in production. *Right:* output of the shelf detection module. Each detected pick-slot is marked with a coloured box in (b). The background map shows other vertical obstacles not classified as shelves. The three false positives are circled with solid-line rectangles. The false negative is marked with a dotted-line rectangle.

as an indicative result, but counting 554 pickslots in total (distributed as 7 lanes with 62 slots each plus 2 lanes with 60 slots each), we have over 99% recall and precision within this region.

The method also detects many of the shelves that are seen from a distance. However, since not all poles have been seen, the oriented bounding boxes that are computed from the lane segments somtimes have an orientation error, which causes the pickslots to be misplaced. A potential workaround for this issue, left for future work, is to only consider scan data within a local region (e. g., 10 m) around the sensor frame when constructing the point cloud map used for shelf detection.

3.4 Parameter selection

Two main parameters need to be selected depending on the environment: the size of each pick-slot (usually the same as a standard pallet, but depends on what goods are stored in the warehouse) and the minimum aisle width w_a (which is used when clustering, to avoid that poles that are nearby get clustered to the same rack segment).

For all results presented here, we have based the pick-slot size on the size of a EUR pallet (80 × 120 cm). For the Orkla example, the aisle width w_a is set to 4.5 m, and for Coop, where aisles are generally narrower, it is set to 2.2 m.

There are also three parameters that govern the pole detection step: the radius r for classifying the shape, the threshold t_l for determining which point distributions are “linear enough”, and the threshold m which is used to remove stray point clusters with too few points to reliably provide evidence for a pole.

For both data sets, we have used $r = 0.35$ m and $m = 5$. However, t_l differs between the two, with $t_l = 0.25$ for Orkla and $t_l = 0.75$ for Coop. The point clouds in the Coop data set are not as precisely registered as they are in the Orkla map, since they have been localised with a reflector-based system, rather than the mapping framework described in Section 2 and Deliverable D1.2. Therefore, the stricter linearity threshold $t_l = 0.25$ is too conservative, and leads to poles being undetected, because the (noisy) distribution of

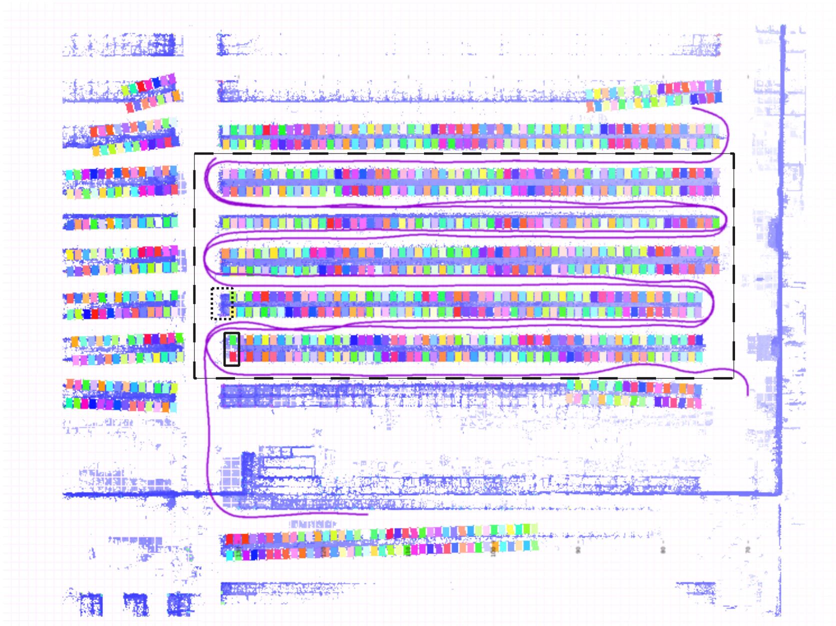


Figure 4: Shelf-detection output from the Coop dataset. The data comes from two runs through the warehouse, the trajectories of which are plotted with purple lines. Only those shelves that have been seen from both sides are included in the quantitative results (this region is marked with a long-dashed rectangle). For this data set, ground truth labels are not available. However, based on our visual assessment of where the pickslots should be, we count two false positives (at one end of one rack, circled with a solid-line rectangle) and two false negatives (at one end of another rack, circled with a dotted rectangle). This figure also highlights potential problems with misalignment and misdetections for shelves that are only seen from a distance.

points within r tends to be more "spherical" than linear. This, in turn, leads to gaps in the detected shelf racks, which manifests in more false negatives. At Orkla, there is more "clutter" in terms of other pole-like structures, and therefore a stricter threshold is better there.

4 Unsupervised labelling of traversable areas

In simple, flat environments, the problem of traversability can be addressed by answering a question: "Is this part of the map low enough?". Thus, utilising a simple height-based navigation map can yield usable results, such as the one shown in Figure 1b.

However, even though obstacles that have been observed as moving during map creation are removed by NDT-OM, some clutter may still remain, e. g., a parked truck or some traces of people. As part of T2.2, we have developed a method that can be used for automatic de-cluttering of 2D navigation maps [3] by analysing the dominant directions in the map and scoring each point of the map based on how well it is aligned to these directions.

Our method for structure extraction and decluttering is called ROSE [3]. ROSE exploits the fact that indoor environments usually contain walls and straight-line elements along a limited set of orientations. Therefore metric maps often have a set of dominant directions. ROSE extracts these directions and uses this information to segment the map into *structure* and *clutter*. The method is described briefly in the following. For more details, please see Kucner et al. [3].

The first step of ROSE is to compute the 2D Discrete Fourier Transform (DFT) $M = \{(m,n)\}$ of an input map $\mu = \{(u,v)\}$. The DFT image is "unfolded" around its centre with an unfolding function c; thus estimating the amplitude values for a discrete set of orientations ϕ and distances ρ from the centre of the frequency spectrum. Cumulative amplitude peaks are computed from the unfolded plot of orientations, and these peaks constitute the dominant directions of the input map. The set of peaks (Φ) can then be used to assess the structure of the map both at the global level and also for individual grid cells.

In order to use ROSE for decluttering, the next step is to identify to what extent the occupied map cells (u,v) belong to the dominant directions. For this purpose, we divide the frequency spectrum into two parts, a structure and a clutter part. The structure part (S) contains the frequency components along the peak directions in Φ:

$$S = \{(u,v)_s | (u,v) = c^{-1}(\phi,\rho), \phi \in \Phi_p, \rho \in (0, \rho_{\max})\}. \tag{1}$$

To obtain the cells in the frequency spectrum that correspond to the structure (S), we apply the folding function (c^{-1}). The folding function finds all the cells in the frequency spectrum that share the orientations (Φ_p) with the peaks. The remaining part of the frequency spectrum $N = S^C$ is then labelled as clutter.

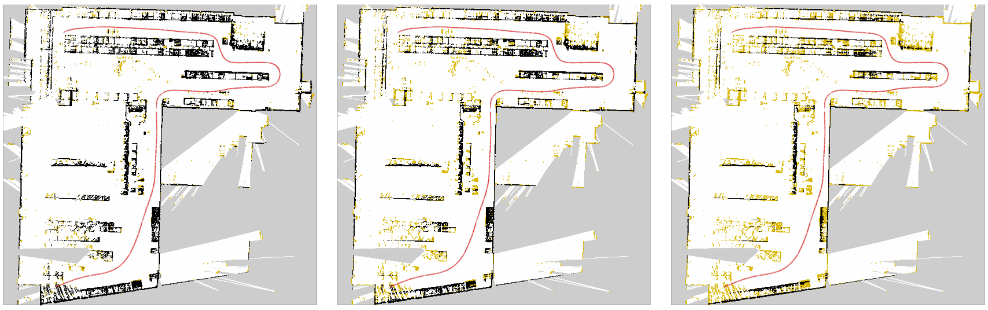
S is then used to reconstruct the structured elements of the map using the Inverse Discrete Fourier Transform (IDFT). This constitutes a *nominal reference map* μ_N; i. e., a representation of what we expect a ground-truth map to look like, in lieu of an actual reference map.

$$\mu_N(m,n) = \frac{1}{XY} \sum_{u=0}^{X-1} \sum_{v=0}^{Y-1} M(u,v) e^{j2\pi(um/X+vn/Y)}, (m,n) \in S \tag{2}$$

The pixel score computed in (2) can be further applied to label pixels as part of either structure or clutter. The split can be executed through simple thresholding. To automatically estimate the threshold value, we propose to use a Gaussian Mixture Model (GMM). To find the threshold we first run Expectation Maximisation (EM) over the list of pixel scores. In this way we obtain two normal distributions: $N_{\text{structure}}$ and N_{clutter}. The threshold is defined as the pixel score $s \in \mu_N$ for which the two Gaussians intersect: $\tau N_{\text{structure}}(s) = (1-\tau) N_{\text{clutter}}(s)$.

This automatic threshold can be given as an initial suggestion for assisting decluttering of the navigation map, and tuned based on the sensitivity required by the application. Figure 5 shows the output of ROSE decluttering of a navigation map from Orkla for the automatic threshold (0.52 in this case) and a selection of manual thresholds.

Going beyond the the traditional way of simply labelling robot map cells as "floor" vs "obstacle", in industrial environments shared with people there may be other – explicit or implicit – no-go zones. Even if an area is flat and – in principle – traversable, there may be many reasons why they *should* not be traversed by people or robots. As such, the maps of dynamics (MoDs) developed in WP2 can also be seen as a type of unsupervised semantic map, where the semantics in question are occurrence of activities and motion patterns. We use MoDs to steer motion planning so as to align with the expected flow in the warehouse using a hierarchical motion planner as described in D5.3 (Section 3). As these methods have been described also in other deliverables, they are not covered in detail here.



(a) Threshold 0.20 (b) Threshold 0.35 (c) Threshold 0.52 (auto-selected)

Figure 5: Decluttering traversability maps with ROSE unsupervised structure detection. The user-selected parameter is a sensitivity scale, which can be auto-computed as in figure (c) and manually adjusted depending on the context as in the remaining figures. The parts of the original occupancy grid map (from Figure 1b) that have been removed are coloured yellow.

5 Unsupervised labelling of temporary storage locations

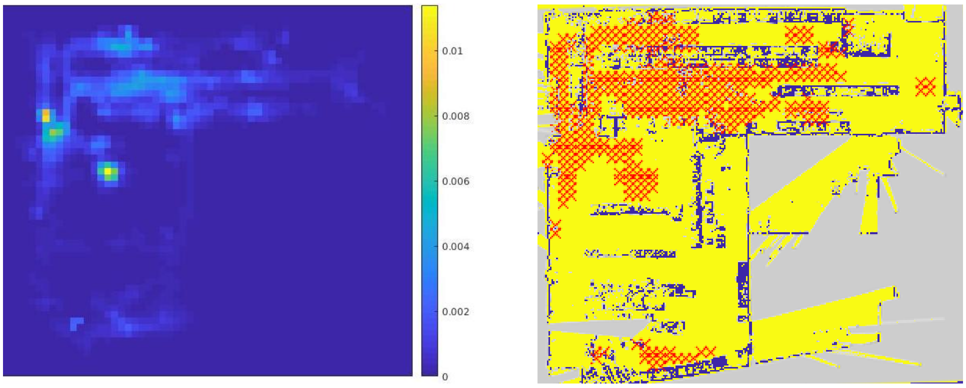
Different from the fixed pick-slots for certain types of goods, the need to temporarily store pallets on unused floor space often occurs; e. g., while preparing an order for shipment or while re-stocking the warehouse with new products from the factory. It is possible to identify locations preferable for temporary pallet storage by combining maps of structure and maps of dynamics. Our approach to unsupervised labelling of locations that are suitable for temporary pallet storage is to combine the traversability map (Section 4) with a map of dynamics (as described in D2.1). In essence, good places to store pallets should be places that are obstacle-free and where people usually do not walk.

In particular, for this implementation we use the CLiFF-map representation [4] which is a time-independent representation of the directions and intensity of motion. Concretely, CLiFF-map represents flow (using people detection from T3.3 as input) as a set of semi-wrapped Gaussian mixture models – one for each location in the map. In addition to the statistically plausible motion *directions* encoded by these mixture models, CLiFF-map also encodes the motion *intensity* for each location based on the number of times a person has been seen in motion there. Figure 6a shows the intensity component of a CLiFF-map recorded at the Orkla warehouse. The quantity that is visualised here is the p value from the CLiFF-map representation, which measures the ratio of activity in a cell (number of times a person has been observed in the cell) to the overall activity in the area (measured as number of times that persons have been observed in the area as a whole).

Through thresholding the flow-intensity map, it is possible to identify locations where pallets should and should not be stored. The result in Figure 6b shows a map of suggested places for storing pallets, where the top 10% most visited locations are forbidden for pallet storage.

6 Unsupervised place categorisation

Finally, we have shown how *place categorisation* from 3D point cloud data can be solved in an unsupervised way, without the supervised training that is required by the state of the art [6]. Place categorisation is the problem of assigning a type of place to a sensor reading



(a) Visualisation of people motion density in the Orkla warehouse. Bright areas represent highly active areas, while dark corresponds to locations with low activity.

(b) Combined map showing suitable storage locations. The yellow regions are suitable for temporary pallet storage. Red crosses denote regions with high expected density of people.

Figure 6: Unsupervised labelling of temporary storage locations by combining long-term maps of dynamics with de-cluttered traversability maps.

(e. g., an image or a 3D scan). What categories are relevant depends to some extent on the application. In the literature, place categories are often room types, such as “kitchen”, “office”, etc. In an intralogistics setting, relevant categories are, e. g., different warehouses or warehouse halls, and “in-aisle” vs “out of aisle”.

Our method for unsupervised place categorisation uses NDT histograms [5] as a global appearance descriptor of 3D point clouds. The appearance descriptors for a set of scans can be clustered using hierarchical k-means++ clustering to produce a hierarchy of semantically meaningful places. The only user-selected parameter is a sensitivity scale Δ that can be set according to how many categories are desired.

Figure 7 shows output from two warehouses: the Coop dataset described above, and a dataset from a central distribution warehouse of dairy producer Arla. In this figure, a Δ threshold that produces three categories is shown. As Δ tends to zero, more categories will be produced; in the limit, one category per point cloud. Figure 8 shows the locations of the scans belonging to the top-three categories in the map. It is diffucult to precisely quantify the performance, since there is no clear-cut ground-truth classification. (At what point is it meaningful to say that the robot is inside the aisle or next to it?) However, we have shown that the performance on an available benchmark dataset (not in a warehouse environment) is close to that of previously published supervised methods that use both range and intensity data. For more detail, please refer to Magnusson et al. [6].

7 Summary and future work

In this report we have summarised the output of the ILIAD semantic mapping system, which contains unsupervised methods for detecting shelves, detecting clutter vs traversable areas as well as maps of dynamics showing which parts of the map are often traversed in practice, labelling areas that are suitable for temporary map storage, and finally place categorisation that can segment maps into similar-looking areas.

The shelf detection method in particular (in combination with structural mapping) is useful during deployment, much reducing the work load for surveying a warehouse and measuring the position of each shelf. Staff from the involved end users (Orkla Foods) have graded how many of the semantic labels are correct, how many are useful for production, and how many are missing. Our method for automatic shelf detection reached 99% recall and 97% precision in the, relatively, unstructured warehouse of Orkla Foods; and over 99% recall and precision in the highly structured warehouse of Coop, in the parts of the map that have been well observed (near the path traversed by the robot).

What remains within ILIAD is to fully integrate the methods for semantic mapping with the rest of the system and deploy on the fleet. One area of improvement would be to investigate a tighter integration between the decluttering and structure detection of ROSE with shelf detection, in order to further reduce the number of false positives and to bias the alignment of shelves to the main structural directions of the warehouse.

References

[1] Henrik Andreasson, Daniel Adolfsson, Todor Stoyanov, Martin Magnusson, and Achim J. Lilienthal. "Incorporating Ego-motion Uncertainty Estimates in Range Data Registration". In: *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*. Sept. 2017, pp. 1389–1395.

[2] K. Klasing, D. Wollherr, and M. Buss. "A clustering method for efficient segmentation of 3D laser data". In: *IEEE Int. Conf. on Rob. and Autom. (ICRA)*. 2008, pp. 4043–4048. DOI: 10.1109/ROBOT.2008.4543832.

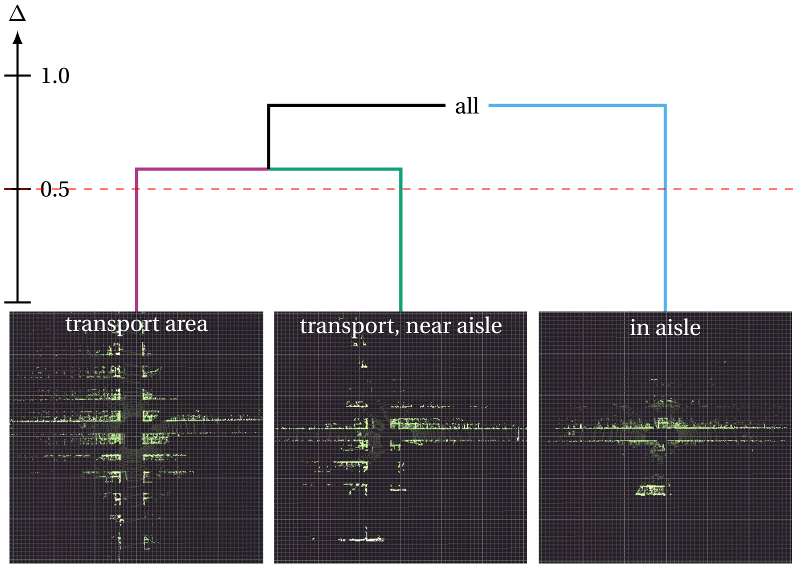
[3] Tomasz Piotr Kucner, Matteo Luperto, Stephanie Lowry, Martin Magnusson, and Achim J. Lilienthal. "Robust Frequency-Based Structure Extraction". In: *IEEE Int. Conf. on Rob. and Autom. (ICRA)*. 2021. arXiv: 2004.08794 [cs.RO].

[4] Tomasz Piotr Kucner, Martin Magnusson, Erik Schaffernicht, Victor H. Bennetts, and Achim J. Lilienthal. "Enabling Flow Awareness for Mobile Robots in Partially Observable Environments". In: *IEEE Robotics and Automation Letters* 2.2 (Apr. 2017), pp. 1093–1100. ISSN: 2377-3766. DOI: 10.1109/LRA.2017.2660060.

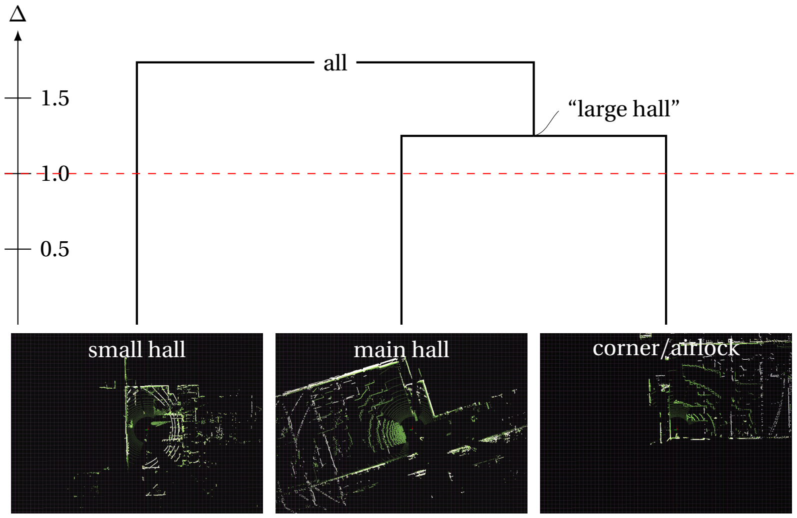
[5] Martin Magnusson, Henrik Andreasson, Andreas Nüchter, and Achim J. Lilienthal. "Automatic Appearance-Based Loop Detection from 3D Laser Data Using the Normal Distributions Transform". In: *Journal of Field Robotics* 26.11–12 (Nov. 2009), pp. 892–914.

[6] Martin Magnusson, Tomasz Piotr Kucner, Saeed Gholami Shahbandi, Henrik Andreasson, and Achim J. Lilienthal. "Semi-Supervised 3D Place Categorisation by Descriptor Clustering". In: *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*. Sept. 2017, pp. 620–625.

[7] Jari Saarinen, Henrik Andreasson, Todor Stoyanov, and Achim J. Lilienthal. "3D normal distributions transform occupancy maps: an efficient representation for mapping in dynamic environments". In: *The International Journal of Robotics Research* (Sept. 2013). DOI: 10.1177/0278364913499415.



(a) Coop



(b) Arla

Figure 7: Dendrograms showing the output of unsupervised hierarchical place categorisation from two warehouses. The leaf nodes show the average point cloud for the category. The Coop dataset (a) is dominated by aisles between rows of shelves, and as such the most relevant place categories are within vs out-of aisles. The Arla dataset (b) features more free-floor storage, and as such the dominant categories are two different halls in the warehouse. (Setting the threshold Δ lower than 1.0 subdivides the two halls into regions.)

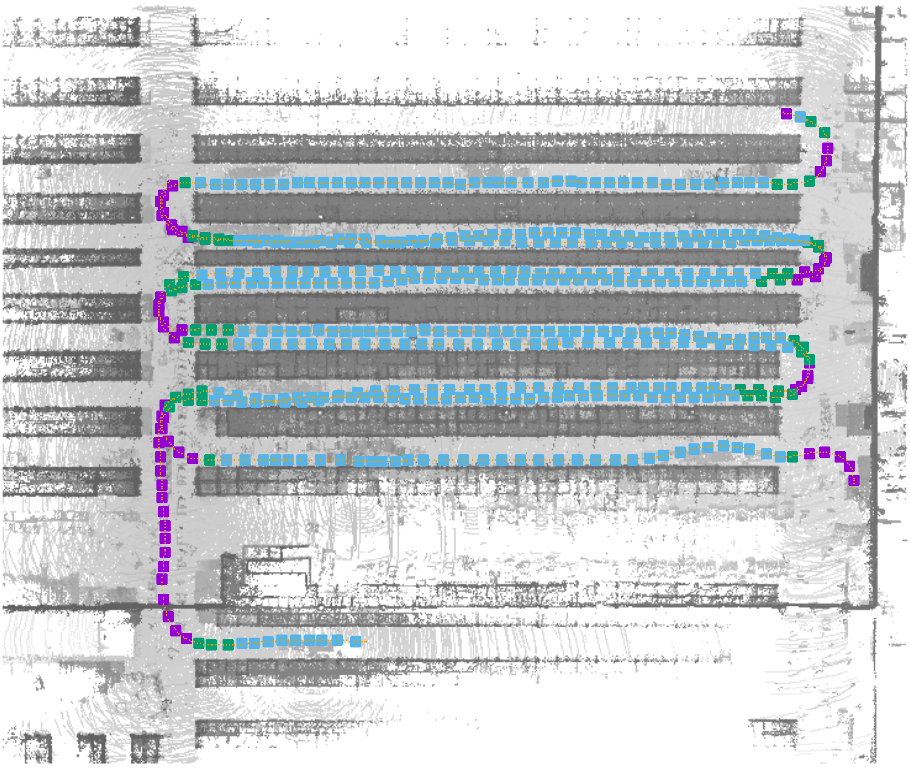


Figure 8: Place categorisation output on the Coop dataset, visualised on the map. The position of each scan is denoted with a coloured dot. The colours of the three detected place categories correspond to Figure 7a, with blue = in aisle, magenta = out of aisle, and green = border region.