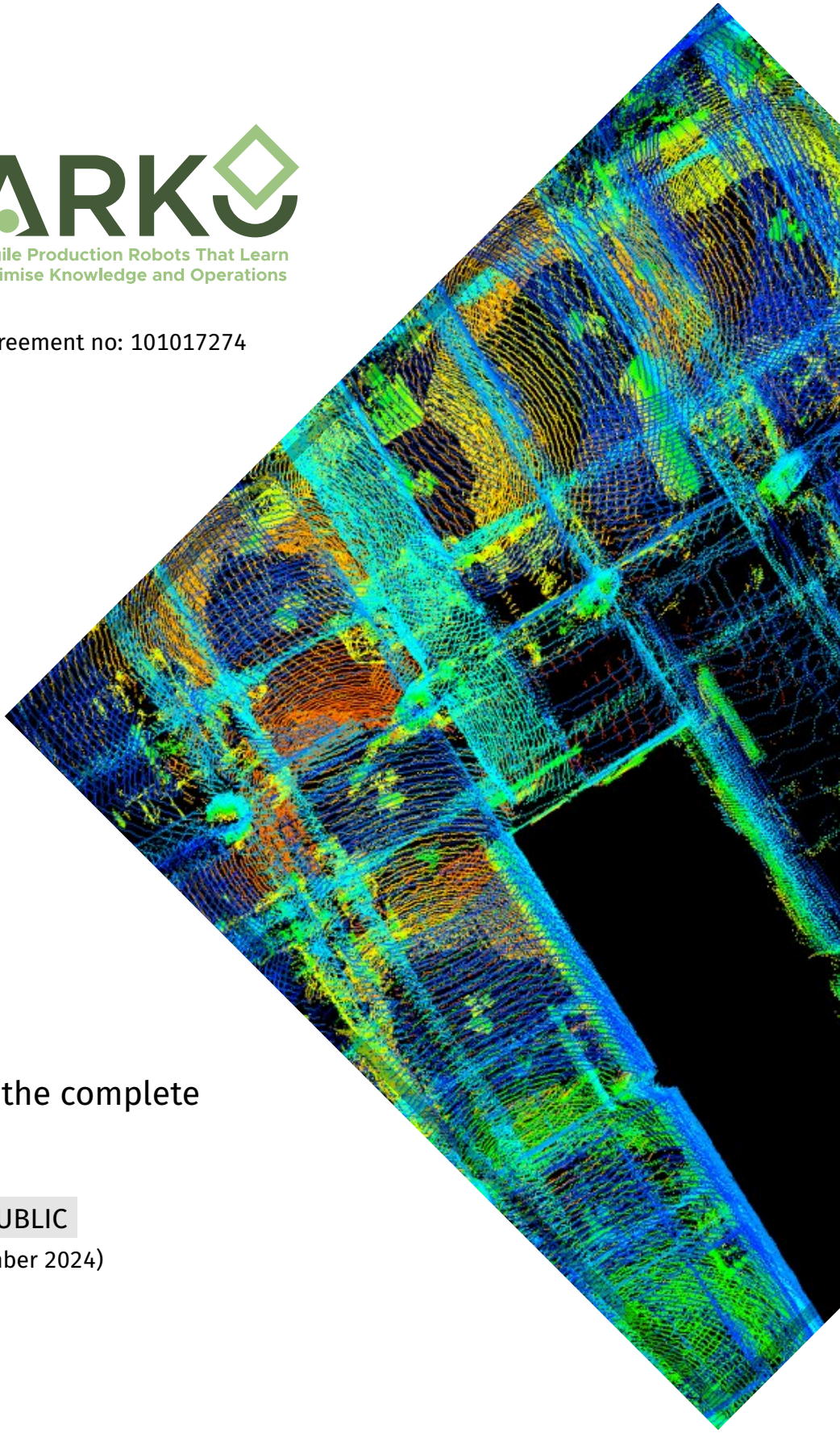
DARKO

Dynamic Agile Production Robots That Learn and Optimise Knowledge and Operations

H2020-ICT-2020-2 Grant agreement no: 101017274



DELIVERABLE 3.3

Implementation of the complete mapping system

Dissemination Level: PUBLIC

Due date: month 48 (December 2024)

Deliverable type: Software

Lead beneficiary: ORU

Contents

1 Introduction **3**

2 Mapping and localization **3**
2.1 Baseline lidar mapping and localization 3
2.2 Efficient neural lidar map reconstruction 3
2.3 High-fidelity RGBD mapping and pose tracking 3

3 Maps of Dynamics **6**
3.1 CLiFF-LHMP 6
3.2 LaCE-LHMP 6

4 Reliability-aware mapping and safe localisation **6**
4.1 Localization risk prediction and mitigation 6
4.2 Map quality assessment 7

1 Introduction

This deliverable consists of the final implementations of mapping and localization functionalities for tasks T3.1, T3.2 and T3.3 in WP3 – including software for constructing and localizing in maps that are purely geometric as well as maps with high-fidelity rendering qualities, and constructing and using maps of dynamics.

We provide links to the software repositories used by the implementation, and some examples of the running system.

Please note that the technologies, and experimental results, are further described in D3.4.

2 Mapping and localization

2.1 Baseline lidar mapping and localization

The baseline mapping and localization stack used by the DARKO robot, as already reported in D3.1, uses a graph-based SLAM method [1, 2] based on NDT-OM sub-maps [3], implemented in the robust_mapping repository. For localizing in the NDT-OM map graph, we use a graph-aware version of NDT-MCL [4] (graph_map). The output of the baseline mapping system is a 3D NDT-OM map (for localization), a 3D point cloud map (mainly for visualization), and a 2D grid map that can be readily integrated with the motion planners from WP6.

This software was part of D3.1 ("Prototype mapping system implementation"). Example output can be seen in Figure 1.

2.2 Efficient neural lidar map reconstruction

Storing detailed large-scale maps is often memory-consuming. Motivated by recent neural implicit representation, we propose a novel data structure combined with neural networks to represent the map implicitly. This method achieves better or competitive quality of the reconstructed 3D surface, compared to recent baselines [5], while consuming minimal memory.

The core concepts of our work, named *3QFP* [6], are illustrated in Figure 2. At its core, 3QFP uses a data structure referred to as *Tri-Quadtrees*, where the whole scene is projected to multiple levels of axis-aligned planar quadtrees. When querying some point in the scene, the point feature will touch different levels of feature grids. The feature vectors surrounding the query point are interpolated and fed into a small neural network which decodes a signed distance value, which can then be used to reconstruct a surface mesh.

Our method achieved detailed descriptions of the scene while consuming less memory. As shown qualitatively in Figure 3, our method uses much fewer parameters but achieves better or competitive reconstruction quality compared to alternative methods, including the recent neural baseline SHINE-mapping [5] and a more traditional ball pivoting mesh reconstruction.

The implementation of 3QFP can be found at https://github.com/ljjTYJR/3QFP.

2.3 High-fidelity RGBD mapping and pose tracking

In addition to accurate *geometric* reconstruction as per above, we have also explored how to reconstruct the scene with high-fidelity *appearance*, aiming to enhance localization and enable additional applications based on rendering novel RGB views from unseen

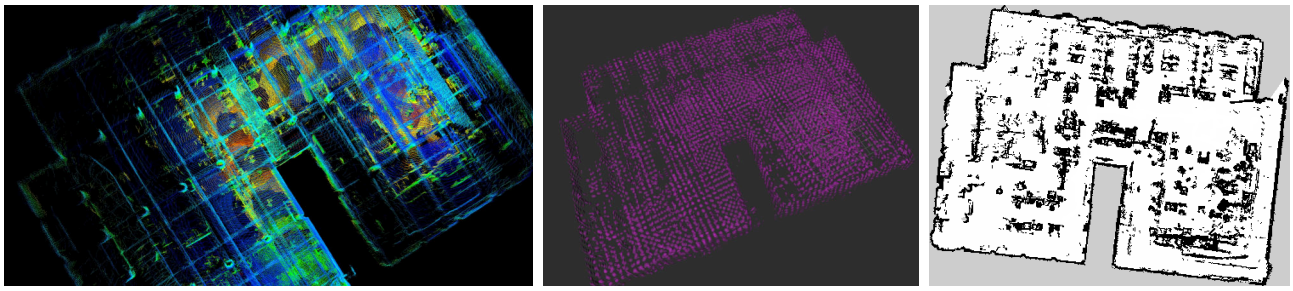


Figure 1: Geometric 2D and 3D maps from Milestone 3 at the Deutsches Museum in Munich, created with the DARKO prototype mapping system from T3.1. Left to right: 3D point cloud map (for visualisation), 3D NDT-OM map (for localization), 2D occupancy grid (for motion planning).

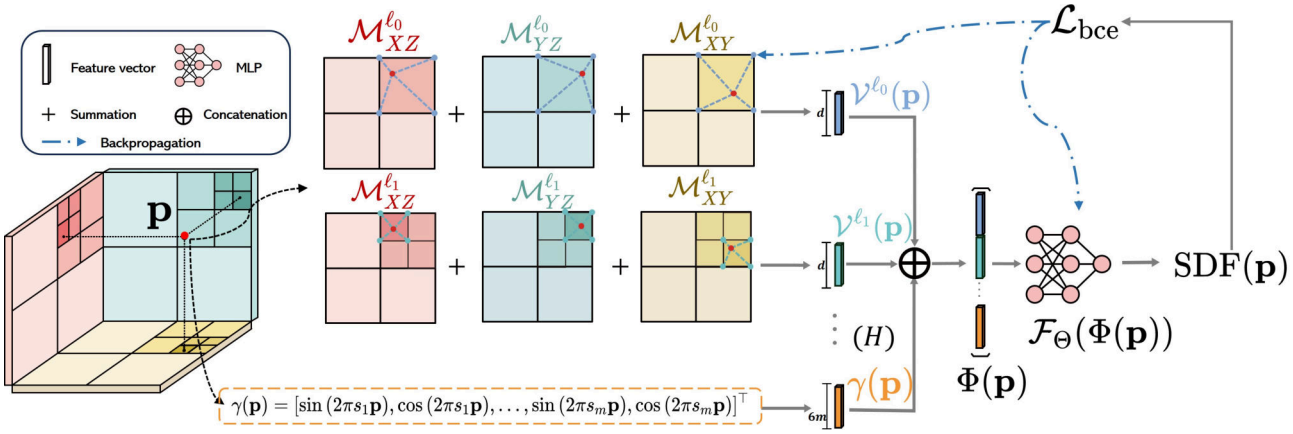


Figure 2: Overview of *3QFP* [6]. We represent the scene with three planar quadtrees $\mathscr{M}_i^\ell$, $i \in \{XZ, YZ, XY\}$ (where ℓ represents the quadtree depth). We store features in the deepest H levels of resolution of quadtrees. When querying for a point $\boldsymbol{p}$, we project it onto planar quadtrees to identify the node containing $\boldsymbol{p}$ at the level ℓ. The feature of $\boldsymbol{p}$ is then calculated by bilinear interpolation based on the queried location and vertex features. We add features at the same level and concatenate among different levels. Concatenated with the positional encoding $\gamma(\boldsymbol{p})$, $\boldsymbol{p}$'s feature ($\Phi(\boldsymbol{p})$) is fed into a small MLP ($\mathscr{F}_\Theta$) to predict the signed distance value (SDF). The learnable features stored in the quadtree nodes and the network parameters are learned by test-time optimization using the loss function $\mathscr{L}_{\text{bce}}$.

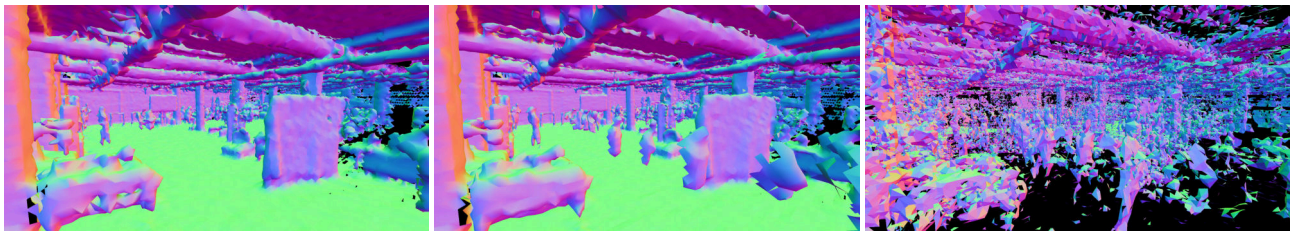


Figure 3: Examples of mesh reconstructions from Milestone 3 at the Deutsches Museum. From left to right: our 3QFP method [6] (< 1 h processing, 5.9 MB RAM), SHINE-mapping [5] (< 1 h processing, 49 MB RAM), ball pivoting reconstruction (> 4 h processing).

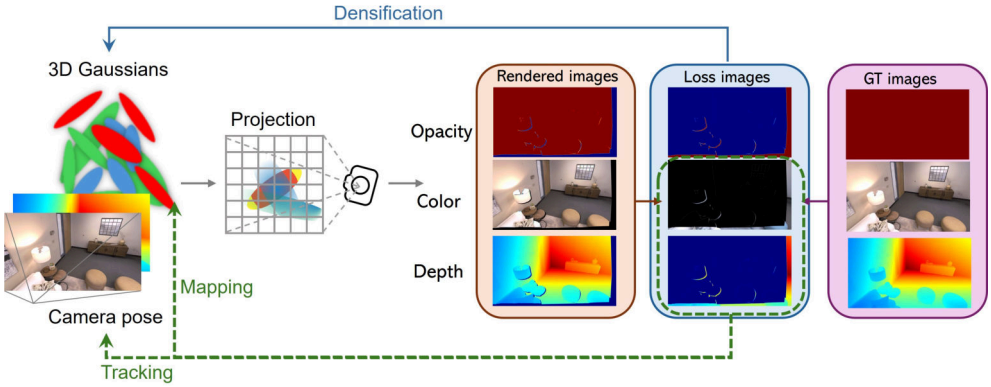


Figure 4: Overview of our mapping and localization framework using Gaussian splats [7]. Our method takes RGBD frames as inputs. During mapping, when given a posed RGBD frame, we first render the opacity image, color image and depth image. Then we compare them with the ground truth to densify the existed map. During tracking, we minimize the color and depth re-rendering loss to optimize the camera pose, and the pose is fed to the mapping pipeline.

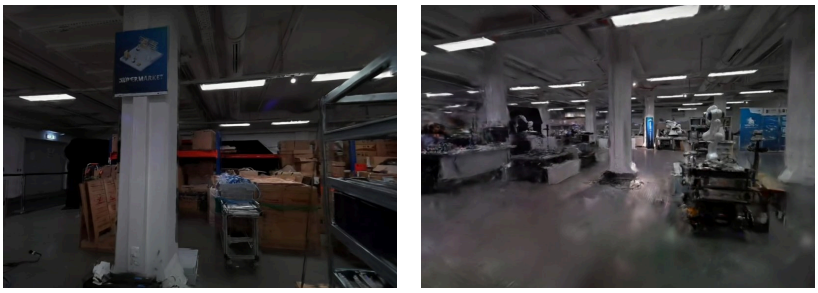


Figure 5: Example output of maps made with Gaussian splatting at Milestone 3 in Deutsches Museum.

viewpoints [7]. We apply 3D Gaussians [8] as representation primitives to create high-fidelity maps. An overview of our method is shown in Figure 4. This method consists of two parts: mapping and localization.

For the mapping part, given a new RGBD frame with known pose, we expand the map by evaluating the rendered image at the new pose. We add new Gaussians based on two criteria: where the rendered opacity is low (to complete unobserved regions) and where the difference between the rendered RGB/depth image and the live inputs is large (to improve the rendering quality).

For localization, we localize the camera by comparing the rendered image with the live input.

Second, we introduce extra regularization parameters to alleviate the “forgetting” problem during contiunous mapping, where parameters tend to overfit the latest frame and result in decreasing rendering quality for previous frames.

In quantitative experiments on benchmark datasets (see D3.4 and Sun et al. [7]), our method achieves better rendering qualities in appearance and geometry, compared to other NeRF-based and state-of-the-art 3DGS-based SLAM methods. Figure 5 shows some examples renderings from Milestone 3.

The code can be found at https://github.com/ljjTYJR/HF-SLAM.

3 Maps of Dynamics

In this section, we describe DARKO's implementation of maps of dynamics (MoDs) from T3.3. MoDs are a class of general representations of place-dependent spatial motion patterns, learned from prior observations). Within the scope of DARKO, we have in particular applied MoDs for long-term human motion prediction (LHMP).

In addition to the implementations described in the following, deliverable D3.4 further describes the underlying technologies and experimental validation, and includes additional work on maps of dynamics that is done within DARKO but not included in the implementation covered by this deliverable (D3.3).

3.1 CLiFF-LHMP

Building on the code base from D3.1, we have developed a new MoD-informed human motion prediction approach, named CLiFF-LHMP [9], which has been shown to be data efficient, explainable, and insensitive to errors from an upstream tracking system.

This approach uses CLiFF-map, a specific MoD trained with human motion data recorded in the same environment. CLiFF-maps represent speed and direction jointly as velocity $\mathbf{V} = [\theta, \rho]^T$ using direction θ and speed ρ, where $\rho \in \mathbb{R}^+$, $\theta \in [0, 2\pi)$. For long-term human motion prediction, we bias a constant velocity prediction with samples from the CLiFF-map to generate multi-modal trajectory predictions.

In two public datasets we show that this algorithm outperforms the state of the art for predictions over very extended periods of time, achieving 45 % more accurate prediction performance at a 50 s prediction horizon compared to the baseline approach [10].

The code can be found in https://github.com/test-bai-cpu/CLiFF-LHMP/.

3.2 LaCE-LHMP

Detecting and identifying abnormal trajectories is a major challenge in motion modeling and prediction. Existing methods typically identify abnormal motions by comparing them to expected behaviours [11] or measuring deviations from normal motions [12]. However, these approaches require labelled data for supervised learning.

The CLiFF-map representation outlined above may struggle to differentiate dominant flow from irregular motion, and therefore the prediction accuracy may be affected by anomalous data. To address these limitations, we propose the Laminar Component Enhanced LHMP approach (LaCE-LHMP) [13]. This approach is inspired by data-driven airflow modelling, which estimates laminar and turbulent flow components and uses predominantly the laminar components to make flow predictions. Based on the hypothesis that human trajectory patterns also manifest laminar flow (that represents predictable motion) and turbulent flow components (that reflect more unpredictable and arbitrary motion), LaCE-LHMP extracts the laminar patterns in human dynamics and uses them for human motion prediction. The framework of LaCE-LHMP is presented in Figure 6.

The code can be found in https://github.com/test-bai-cpu/LaCE-LHMP.

4 Reliability-aware mapping and safe localisation

4.1 Localization risk prediction and mitigation

In addition to what was reported in D3.2, the final implementation of localization quality assessment includes an extended localization risk map representation that includes the level of dynamics in the risk assessment in addition to the level of alignability. We quantify

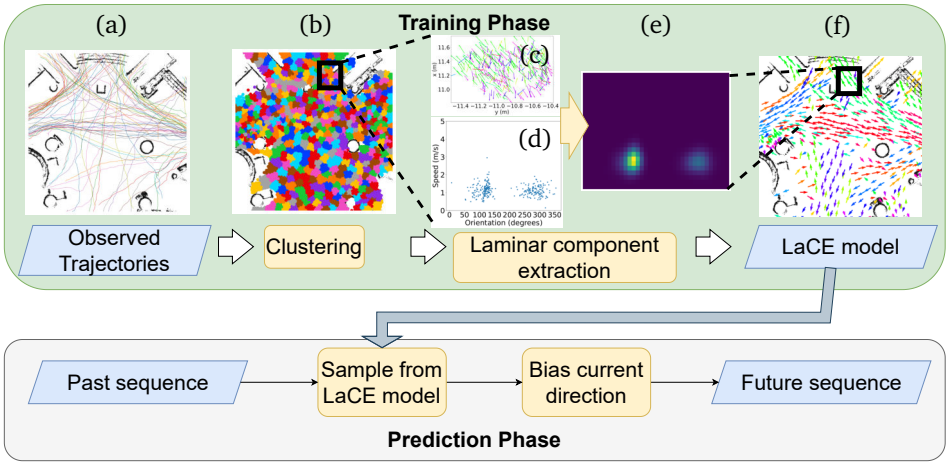


Figure 6: Diagram illustrating the training and prediction phases of the LaCE-LHMP approach. In the training phase, observed trajectories (a) are used. Velocity observations, which are depicted in **(c)** for (x, y) and **(d)** for ω-v distribution, are clustered using K-means into K clusters, shown in **(b)**. From each cluster's joint ω-v distribution, a discrete ω-v histogram Γ^R is estimated to extract the laminar component Γ^L, as shown in **(e)**. The directions with the highest likelihood in Γ^L are represented by colored arrows in the LaCE model **(f)**. The LaCE model is then utilized for prediction.

dynamics by relying on the independent Markov chain approach (iMac) [14], although our implementation differs in some aspects from the original implementation in the ndt_core_public repository listed below. We have also introduced a novel probabilistic model in the form of a Bayesian network that enables the prediction of localization errors given the conditions of the environment. This is further described in D3.4.

The code can be found at:

- https://gitsvn-nt.oru.se/darko/software/alignability
- https://gitsvn-nt.oru.se/darko/software/risk_map.git
- https://gitsvn-nt.oru.se/software/ndt_core_public.git (branch *nice-devel*)

4.2 Map quality assessment

Our implementation for *reference-free map quality assessment* is based on a variational autoencoder that can assess 2-D occupancy grid maps, as also described in D3.2 and D3.4.

Based on an ablation study, we have that the following network design produces good results. The encoder network first downsamples the patch with two blocks of 2-D convolutions and ReLU layers with 32 and 64 filters, respectively. A kernel size of 3×3 and stride length 2 is used for each convolution layer. The latent parameter size is 16. The decoder network has two blocks of transposed convolution and ReLU layers each with kernel size 3×3 and stride length 2 with 64 and 32 filters, respectively. The Adam optimizer with learning rate of 0.001, patch batch size of 128 and 20 epochs is used for training.

The code can be found at https://gitsvn-nt.oru.se/darko/software/mqa.

References

[1] Daniel Adolfsson, Stephanie Lowry, Martin Magnusson, Achim J. Lilienthal, and Henrik Andreasson. "A Submap per Perspective - Selecting Subsets for SuPer Mapping that Afford Superior Localization Quality". In: *European Conference on Mobile Robots*. 2019.

[2] 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.

[3] 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* 32.14 (2013), pp. 1627–1644.

[4] Jari Saarinen, Henrik Andreasson, Todor Stoyanov, and Achim Lilienthal. "Normal Distribution Transform Monte-Carlo Localization (NDT-MCL)". In: *Proc. IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* 2013, pp. 382–389.

[5] Xingguang Zhong, Yue Pan, Jens Behley, and Cyrill Stachniss. "Shine-mapping: Large-scale 3d mapping using sparse hierarchical implicit neural representations". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 8371–8377.

[6] Shuo Sun, Malcolm Mielle, Achim J. Lilienthal, and Martin Magnusson. "3QFP: Efficient neural implicit surface reconstruction using Tri-Quadtrees and Fourier feature Positional encoding". In: *IEEE Int. Conf. on Rob. and Autom. (ICRA)*. 2024, pp. 4036–4044.

[7] Shuo Sun, Malcolm Mielle, Achim J. Lilienthal, and Martin Magnusson. "High-fidelity SLAM using Gaussian splatting with rendering-guided densification and regularized optimization". In: *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*. 2024.

[8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. "3d gaussian splatting for real-time radiance field rendering." In: *ACM Trans. Graph.* 42.4 (2023), pp. 139–1.

[9] Yufei Zhu, Andrey Rudenko, Tomasz P. Kucner, Luigi Palmieri, Kai O. Arras, Achim J. Lilienthal, and Martin Magnusson. "CLiFF-LHMP: Using Spatial Dynamics Patterns for Long-Term Human Motion Prediction". In: *IEEE Int. Conf. on Intell. Rob. and Systems (IROS)*. 2023.

[10] A. Rudenko, L. Palmieri, A. J. Lilienthal, and K. O. Arras. "Human Motion Prediction under Social Grouping Constraints". In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. 2018.

[11] W. Liu, D. Lian W. Luo, and S. Gao. "Future Frame Prediction for Anomaly Detection – A New Baseline". In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[12] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. "Soft + Hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection". In: *Neural networks* 108 (2018), pp. 466–478.

[13] Y. Zhu, H. Fan, A. Rudenko, M. Magnusson, E. Schaffernicht, and A. J. Lilienthal. "LaCE-LHMP: Airflow Modelling-Inspired Long-Term Human Motion Prediction By Enhancing Laminar Characteristics in Human Flow". In: *IEEE Int. Conf. on Rob. and Autom. (ICRA)*. 2024.

[14] Jari Saarinen, Henrik Andreasson, and Achim J. Lilienthal. "Independent Markov chain occupancy grid maps for representation of dynamic environment". In: *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. 2012, pp. 3489–3495.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017274