SoftwareX 26 (2024) 101750

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

End to end navigation stack for nuclear power plant inspection with mobile robot

Janusz Będkowski

Institute of Fundamental Technological Research, Polish Academy of Science, ul. Pawińskiego 5b, Warsaw, 02-106, Mazowieckie, Poland

ARTICLE INFO

Keywords:
Navigation
Inspection
Mobile robot
SLAM
Localization

ABSTRACT

This paper describes a novel approach for nuclear facility inspection with novel automated 3D mapping system as an open source end to end navigation stack available at https://github.com/JanuszBedkowski/msas_enrich_2023. Incidents such as Fukushima, Majak or Chernobyl as well as the decommissioning and dismantling of old nuclear facilities (*e.g.* Sellafield, Asse or Murmansk) are showing great importance of the robotic technology. Rapid inspection requires reliable, accurate, precise and repeatable simultaneous localization and mapping. Proposed SLAM approach uses only non repetitive scanning pattern Lidar (Livox Mid360) and integrated inertial measurement unit. The novelty is based on feature less single core SLAM implementation. It fuses Normal Distributions Transform and motion model for simultaneous map building and current pose estimation. Motion model bounds an optimization result, thus it is stable and reliable. It requires less than 10 ms for pose update, trajectory tracking and emergency behavior. This method is a candidate for real time application since a calculation time is bounded and it uses only one core of Intel Celeron CPU G1840 2.8 GHz. It was tested both (i) during EnRicH 2023 https://enrich.european-robotics.eu/ — the European robotics hackathon, (ii) in laboratory conditions. This open source project provides also software of base station, thus it is first end to end solution available in literature.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00759
Permanent link to Reproducible Capsule	not available
Legal Code License	MIT
Code versioning system used	git
Software code languages, tools, and services used	C++, python
Compilation requirements, operating environments & dependencies	cmake, ROS1, https://github.com/JanuszBedkowski/observation_equations.git, https://github.com/nlohmann/json.git, https://github.com/LASzip/LASzip.git, https://github.com/xioTechnologies/Fusion.git, https://gitlab.com/libeigen/eigen.git, https://github.com/ocornut/imgui
Link to developer documentation/manual	https://github.com/JanuszBedkowski/msas_enrich_2023
Support email for questions	januszbedkowski@gmail.com

1. Motivation and significance

Simultaneous localization and mapping capability is an essential functionality of autonomous mobile robot working in GPS/GNNS denied environments. Mobile robot can be equipped with an odometry, Lidar, camera, IMU and CPU for performing data fusion algorithm. This algorithm suppose to retrieve current pose based on sensor readings and simultaneously build a map — a digital representation of an environment. Fusion algorithm should be reliable, robust and efficient to assure proper mission execution. Nuclear power plant inspection requires full autonomy, since the communication link is not always sufficient, therefore robot cannot be remotely controlled all over the mission task. Full autonomy in this particular scenario should guarantee

E-mail address: januszbedkowski@gmail.com.

https://doi.org/10.1016/j.softx.2024.101750
Received 8 November 2023; Received in revised form 1 April 2024; Accepted 20 April 2024
Available online 2 May 2024

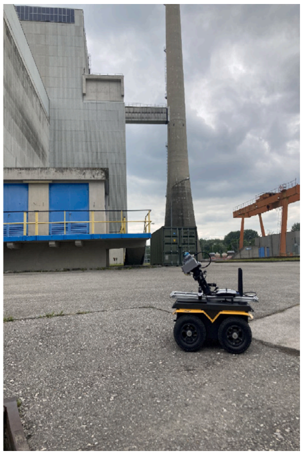


Fig. 1. Robot and nuclear power plant.

at least (A) path planning, (B) mission execution and (C) return to base station for a communication link retrieval. A, B, C require a SLAM capability since we do not conciser a digital representation of the nuclear power plant. Moreover, result of (B) should be a layout of the scenario with marked all important objects recognized by robot. We assume limited number of a dynamic obstacles, since Nuclear Power Plant is rather a static environment. Most important issue is related with limited time for the task execution, for this reason mobile robot suppose to move as fast as it is possible. Above mentioned mission characteristics and the need of lowering the cost of a mobile robot determine this research, since there is no commercial application or an open source project fulfilling this requirements. Proposed approach assumes limited communication link up-to 5 meters range so robot should always return to the base station for retrieving a communication link and for receiving next mission task. The mission considered in this paper is a task of providing layout of the environment acquired during traversing all consecutive goals provided by an operator. Autonomous mobile robot is capable performing such tasks when SLAM algorithm provides accurate map and accurate localization all over the time. For this reason novel approach is proposed. It is based on fused Normal Distributions Transform with motion model. Map is constantly updated during all missions. It is possible to retrieve the layout of environment thanks to an estimated pose of the robot and LiDAR data. This layout is used for manual path planning. It means base station provides functionality for placing goals that construct consecutive set of trajectory nodes that robot will trespass during mission. Robot performs way-point navigation using naive approach. Thus, once it reaches the goal it changes heading to another way-point. During navigation to the next way-point robot is trying to adjust heading to the next goal. In current implementation it is not capable to avoid obstacles, thus it simply stops once it detects obstacle in front or back. It is sufficient looking from presented application point of view.(see Figs. 1, 13, 15 and 18).

2. Software description

The core of an implementation is not using g^2o [1], gtsam [2], manif [3], ceres [4] or other well known libraries for solving non linear optimization problems. It is based on novel approach [5] that incorporates only SymPy [6] for automatic analytic Jacobian generation from observations equations available at https://github.com/JanuszBedkowski/observation_equations and Eigen library [7] for solving non linear optimization problem. Lie algebra is not used since the rotation matrix is parameterized using Tait–Bryan convention and all analytic Jacobians are generated automatically. This guarantee minimizing major issue of hand-crafted Jacobians (potential errors). Tait–Bryan convention enables intuitive uncertainty setup. Thus, pose is represented as 6 parameter vector (x, y, z, yaw, pitch, roll). In our case we set up an uncertainty for all angles (yaw, pitch, roll) with sigma equals 1 degree, thus our optimization is stable. Unfortunately, other implementations such as Fast-Lio2 [8] and successor Faster-Lio [9] do not guarantee bounded result for optimization. Thus, such implementation is not sufficient for our application. An implementation can be interpreted as factor graph [10] since it is based on a probabilistic approach commonly used in mobile robotics. General assumption is that local point cloud is transformed according to IMU readings processed my Madgwick filter https://github.com/bjohnsonfl/Madgwick_Filter [11]. Optimization problem incorporates several observation equations such as:

- point to point (basic observation equation for iterative closest point algorithm [12]).
- relative pose (fundamental observation equation for constructing pose graph SLAM [13]).
- Normal Distributions Transform (an extension to ICP incorporating uncertainty and using Mahanobis distance between query point and target [14]).
- fixed optimized parameter (to eliminate first node from optimization [5]).

These observation equations construct an optimization problem composed of 12 parameters (random variables) to solve (previous pose, current pose) at each iteration step. The pose graph SLAM is composed of two nodes and one edge between them, thus it is smallest instance possible. The SLAM algorithm is as follows:

- loop begin
- transform local point cloud with rotation matrix resulted from Madgwick Filter applied for raw IMU data
- form and solve optimization problem based on previous pose, current pose, map and local point cloud
- update map (update all covariances)
- set previous pose as current pose
- loop end

2.1. Observation equations

An observation Eq. (1) relates a target value y_i, a model function $\boldsymbol{\Psi}_{[\beta]}(\boldsymbol{x}_i)$ and its *residual* r_i. Minimizing a difference between the target value and the model function allows defining the optimization problem as (2) that is described in detail in Section 2.4.

$$\underbrace{\underbrace{r_i}_{residual} = \underbrace{y_i}_{target\ value} - \underbrace{\boldsymbol{\Psi}_{[\beta]}(\boldsymbol{x}_i)}_{model\ function}}_{observation\ equation} \tag{1}$$

$$\underbrace{\boldsymbol{\beta}^* = \min_{\boldsymbol{\beta}} \sum_{i=1}^{C} \left(y_i - \boldsymbol{\Psi}_{[\beta]}\left(\boldsymbol{x}_i\right)\right)^2}_{optimization\ problem} \tag{2}$$

where, $\boldsymbol{\beta}$ is the vector of n adjustable parameters by the optimization process resulting $\boldsymbol{\beta}^*$ and $\boldsymbol{x}$ is the vector of input variables of the model function. C is number observations.

2.2. Point to point

Point to point observation equation incorporates model function given by (3),

$$\boldsymbol{\Psi}_{[\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar}}(\boldsymbol{R}_{W\leftarrow Lidar}, \boldsymbol{t}_{W\leftarrow Lidar}, \boldsymbol{P}^l, \boldsymbol{P}^{nn,g}) = [\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar}\boldsymbol{P}^l - \boldsymbol{P}^{nn,g} \tag{3}$$

where $\boldsymbol{P}^{nn,g}$ is a nearest neighbor to $\boldsymbol{P}^l$ expressed in global reference frame as $[x^{nn,g}, y^{nn,g}, z^{nn,g}]^\intercal$. Thus, point to point observation equation is given in form of (4),

$$\underbrace{[\boldsymbol{\delta}^{3\times 1}]}_{residuals} = \underbrace{[\mathbf{0}^{3\times 1}]}_{target\ values} - \underbrace{([\boldsymbol{R},\boldsymbol{t}]^{3\times 4}_{W\leftarrow Lidar}\boldsymbol{P}^{l} - \boldsymbol{P}^{nn,g})}_{model\ function} \tag{4}$$

where target value (our expectation) is $[\mathbf{0}^{3\times 1}]$ interpreted as no difference between coordinates of nearest neighbors. This difference is denoted as residuals $[\boldsymbol{\delta}^{3\times 1}]$.

2.3. Relative pose

This constraint is a fundamental concept of Pose Graph SLAM. It allows modeling the relations between measurements and observations in form of a graph [15], where nodes are related with poses end edges that relate the difference between measures and observations. Relative pose $[\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar,12}$ from pose $[\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar,1}$ to pose $[\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar,2}$ is expressed in Eq. (5).

$$[\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar,12} = [\boldsymbol{R},\boldsymbol{t}]^{-1}_{W\leftarrow Lidar,1}[\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar,2} = [\boldsymbol{R},\boldsymbol{t}]_{W\rightarrow Lidar,1}[\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar,2} \tag{5}$$

To construct a relative pose observation equation it is necessary to introduce function $\beta = m2v([\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar,12})$ that retrieves parametric form of rotation matrix (Tait–Bryan angles, quaternion, Rodrigues etc.). The relative pose observation equation is given in (6).

$$\underbrace{[\delta]^{N\times 1}}_{residuals} = \underbrace{[\beta^{target}]^{N\times 1}}_{target\ values} - \underbrace{m2v_{[\beta]}([\boldsymbol{R},\boldsymbol{t}]^{12})}_{model\ function} \tag{6}$$

2.4. Optimization problem formulation

Optimization problem is defined as (7).

$$\beta^{*} = \min_{\beta}\sum_{k=1}^{K}\sum_{i=1}^{C^{K}}\left(\boldsymbol{y}_{i}^{k} - \boldsymbol{\Psi}^{k}_{[\beta]}\left(\boldsymbol{x}_{i}^{k}\right)\right)^{2} \tag{7}$$

It contains K observation equation blocks. Each block contains C^{K} instances of observation equations that are considered the same type. It is extended to weighted non-linear least squares by adding weights for each observation equation. Thus, it is possible to construct probabilistic approach for our problem. It will be discussed in the following sections.

2.5. Rotation matrix parametrization

Rigid transformation in SO(3) is expressed in parametric form as Tait–Bryan angles. Other parametric forms exist such as quaternions, Rodrigues, Cayley Formula [16], but they are not considered in this research since Tait–Bryan angles, sufficiently cover the desired scope of the investigation. Starting from the basic concept, in three-dimensional space rotations via each axis are given as rotation ω via x axis (8), φ via (9) and κ via z axis (10).

$$\mathbf{R}_{x}(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{bmatrix} \tag{8}$$

$$\mathbf{R}_{y}(\varphi) = \begin{bmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix} \tag{9}$$

$$\mathbf{R}_{z}(\kappa) = \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

There are two different conventions for the definition of the rotation axes — proper Euler angles ($\boldsymbol{R}_z$-$\boldsymbol{R}_x$-$\boldsymbol{R}_z$, $\boldsymbol{R}_x$-$\boldsymbol{R}_y$-$\boldsymbol{R}_x$, $\boldsymbol{R}_y$-$\boldsymbol{R}_z$-$\boldsymbol{R}_y$, $\boldsymbol{R}_z$-$\boldsymbol{R}_y$-$\boldsymbol{R}_z$, $\boldsymbol{R}_x$-$\boldsymbol{R}_z$-$\boldsymbol{R}_x$, $\boldsymbol{R}_y$-$\boldsymbol{R}_x$-$\boldsymbol{R}_y$) and Tait–Bryan angles ($\boldsymbol{R}_x$-$\boldsymbol{R}_y$-$\boldsymbol{R}_z$, $\boldsymbol{R}_y$-$\boldsymbol{R}_z$-$\boldsymbol{R}_x$, $\boldsymbol{R}_z$-$\boldsymbol{R}_x$-$\boldsymbol{R}_y$, $\boldsymbol{R}_x$-$\boldsymbol{R}_z$-$\boldsymbol{R}_y$, $\boldsymbol{R}_z$-$\boldsymbol{R}_y$-$\boldsymbol{R}_x$, $\boldsymbol{R}_y$-$\boldsymbol{R}_x$-$\boldsymbol{R}_z$). For the convenience, in this research the rotation in SE(3) is calculated in arbitrary chosen Tait–Bryan convention ($\boldsymbol{R}_x$-$\boldsymbol{R}_y$-$\boldsymbol{R}_z$) — Eq. (11),

$$\boldsymbol{R}(\omega,\varphi,\kappa) = \mathbf{R}_{x}(\omega)\boldsymbol{R}_{y}(\varphi)\boldsymbol{R}_{z}(\kappa) = \begin{bmatrix} c(\varphi)c(\kappa) & -c(\varphi)s(\kappa) & s(\varphi) \\ c(\omega)s(\kappa)+s(\omega)s(\varphi)c(\kappa) & c(\omega)c(\kappa)-s(\omega)s(\varphi)s(\kappa) & -s(\omega)c(\varphi) \\ s(\omega)s(\kappa)-c(\omega)s(\varphi)c(\kappa) & s(\omega)c(\kappa)+c(\omega)s(\varphi)s(\kappa) & c(\omega)c(\varphi) \end{bmatrix} \tag{11}$$

where s is sin and c is cos functions.

2.5.1. Weighted non-linear least squares optimization

We start by rearranging Eq. (7) to form (12) by concatenating all residual blocks into sum square errors for the clarity of further notation.

$$SSR = \sum_{i=1}^{C} r_i^2 = \underbrace{\sum_{i=1}^{C}\left(y_i - \boldsymbol{\Psi}_{[\beta]}(\boldsymbol{x}_i)\right)^2}_{objective\ function} \tag{12}$$

The model function $\boldsymbol{\Psi}_{[\beta]}(\boldsymbol{x}_i)$ contains $\boldsymbol{n}$ parameters of $\boldsymbol{\beta}$, thus there are n gradient Eqs. (13) derived from (12).

$$\frac{\partial SSR}{\partial\beta_j} = 2\sum_{i=1}^{C}\frac{\partial r_i}{\partial\beta_j}r_i = -2\sum_{i=1}^{C}\frac{\partial\boldsymbol{\Psi}_{[\beta]}(\boldsymbol{x}_i)}{\partial\beta_j}r_i = 0 \quad (j = 1,\ldots,n) \tag{13}$$

Such a system can be solved in some cases by so-called closed form solution [17,18], thus the solution is obtained in a single step, otherwise it has to be solved using an iterative approach by solving Eq. (14),

$$\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}\boldsymbol{\Delta\beta} = \overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{\Delta y} \tag{14}$$

where $\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}$ is the Jacobian of the model function $\boldsymbol{\Psi}$. Extending Eq. (12) by weights w provides a mechanism to control the impact of each observation equation into the optimization process and it is formulated as (15).

$$SSR = \sum_{i=1}^{C} w_i r_i^2 \tag{15}$$

The optimal solution that minimizing SSR can be found with solving formula (16),

$$\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{W}\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}\boldsymbol{\Delta\beta} = \overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{W}\boldsymbol{\Delta y} \tag{16}$$

where $\boldsymbol{W}$ is a diagonal weight matrix, thus it is easy to provide the independent impact for each observation equation. Once we introduce rotation matrix $\boldsymbol{R}$ that provides a correlation mechanism to $\boldsymbol{W}$ the final Weighted non-linear Least Squares Optimization is formulated as (17).

$$\boxed{\begin{aligned} \overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{R}\boldsymbol{W}\boldsymbol{R}^{\intercal}\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}\boldsymbol{\Delta\beta} = \overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{R}\boldsymbol{W}\boldsymbol{R}^{\intercal}\boldsymbol{\Delta y} &\overset{\boldsymbol{R}\boldsymbol{W}\boldsymbol{R}^{\intercal}=\boldsymbol{\Omega}}{\Rightarrow} \\ \overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{\Omega}\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}\boldsymbol{\Delta\beta} = \overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{\Omega}\boldsymbol{\Delta y} &\overset{\boldsymbol{\Omega}=\boldsymbol{\Sigma}^{-1}}{\Rightarrow} \\ \overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{\Sigma}^{-1}\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}\boldsymbol{\Delta\beta} = \overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Delta y} & \end{aligned}} \tag{17}$$

where $\boldsymbol{\Omega} = \boldsymbol{R}\boldsymbol{W}\boldsymbol{R}^{\intercal}$ is the information matrix and $\boldsymbol{\Sigma} = \boldsymbol{\Omega}^{-1}$ is the covariance matrix. This covariance matrix is crucial to form Normal Distributions Transform elaborated in Section 2.6 and for uncertainty of the motion model that bounds the result of the optimization process.

2.5.2. Solver

In this research, we consider Gauss–Newton optimization method with predefined number of iterations. For observation Eq. (1) the Gauss–Newton method is expressed as (18),

$$\beta^{k+1} = \beta^{k} + \left(\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}\right)^{-1}\overset{\boldsymbol{\Psi}}{\boldsymbol{J}}{}^{\intercal}\boldsymbol{r}(\beta^{k}) \tag{18}$$

where β^{k+1} is an updated vector state of the optimization step.

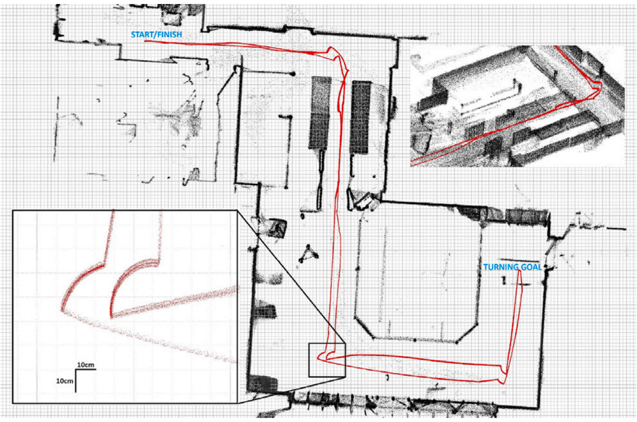


Fig. 2. Laboratory tests in 10 × 10 meters region. Mobile robot repeats executing the same mission (START/TURNING GOAL/FINISH) 20 times. All trajectories are marked by red color. It can be seen that repetitiveness is on decimeter level.

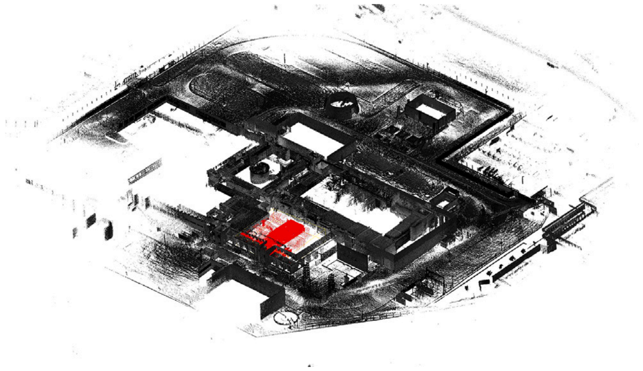


Fig. 3. Intersection of the nuclear power plant and region of EnRicH trial (red color).

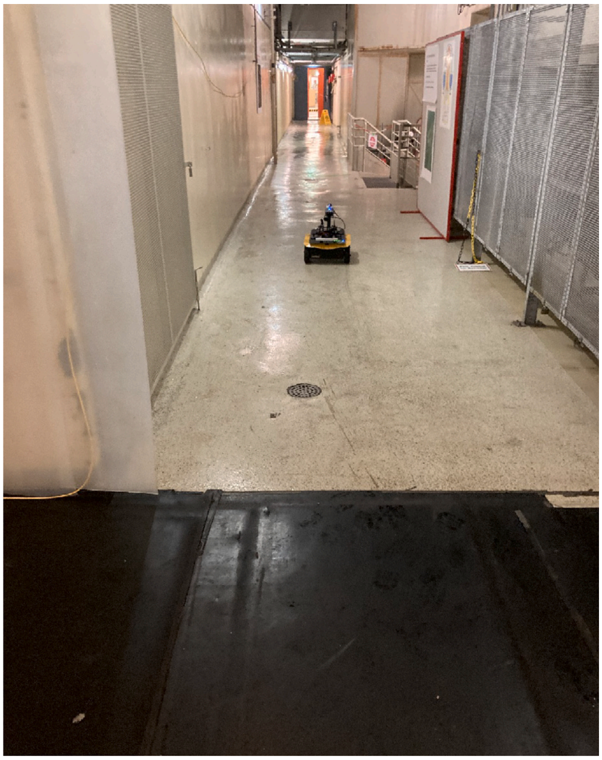


Fig. 4. Working conditions during EnRicH trial. The surface is rather flat. The distance between obstacles is large. The lightning conditions are satisfactory. Dynamic obstacles not evident. Challenge 1: limited communication. Challenge 2: limited time for task execution.

2.6. Normal distributions transform

Normal Distributions Transform [14] is an alternative technique to ICP for point cloud data registration and it is available in a well-known Point Cloud Library [19] open source project. It is limited to the pair-wise matching of two point clouds, thus a contribution of the proposed research is a novel approach to NDT enabling fusing it with pose graph SLAM. The key element of the NDT is the representation of the data as a set of normal distributions organized in the regular grid over 3D space. These distributions describe the probability of finding a 3D point at a certain position. The advantage of the method is that it gives a smooth representation of the point cloud, with continuous first and second-order derivatives. Thus, standard optimization techniques described in this paper can be applied. Another advantage of NDT over ICP is its much less computational complexity since the consumptive nearest neighborhood search procedure is not needed. Authors of [9] also elaborates this advantage. The 3D space decomposition into the regular grid introduces some minor artefacts, but in a presented experiment it is a negligibly small disadvantage. For each bucket from a regular grid containing a sufficient number of measured points NDT calculates the mean given by Eq. (19) and the covariance given by Eq. (20).

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{k=1}^{m} \boldsymbol{P}^g_k \tag{19}$$

$$\boldsymbol{\Sigma} = \frac{1}{m-1}\sum_{k=1}^{m} (\boldsymbol{P}^g_k - \boldsymbol{\mu})(\boldsymbol{P}^g_k - \boldsymbol{\mu})^\intercal \tag{20}$$

The likelihood of having measured point $\boldsymbol{P}^g_m$ is given by Eq. (21).

$$p(\boldsymbol{P}^g_m) = \frac{1}{(2\Pi)^{\frac{1}{2}}\sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{(\boldsymbol{P}^g_m - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1} (\boldsymbol{P}^g_m - \boldsymbol{\mu})}{2}\right) \tag{21}$$

Each $p(\boldsymbol{P}^g_m)$ can be seen as an approximation of the local surface within the range of the bucket. It describes the position $\boldsymbol{\mu}$ of the surface as well as its orientation and smoothness given by $\boldsymbol{\Sigma}$. Let $\boldsymbol{\Psi}([\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar}, \boldsymbol{P}^l_m)$ will be a transformation function of the local measurement point $[\boldsymbol{P}^l_m, 1]^\intercal$ via pose $[\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar}$ expressed as (22).

$$\boldsymbol{\Psi}([\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar}, \boldsymbol{P}^l_m) = \boldsymbol{P}^g_m = [\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar} \begin{bmatrix} \boldsymbol{P}^l_m \\ 1 \end{bmatrix} \tag{22}$$

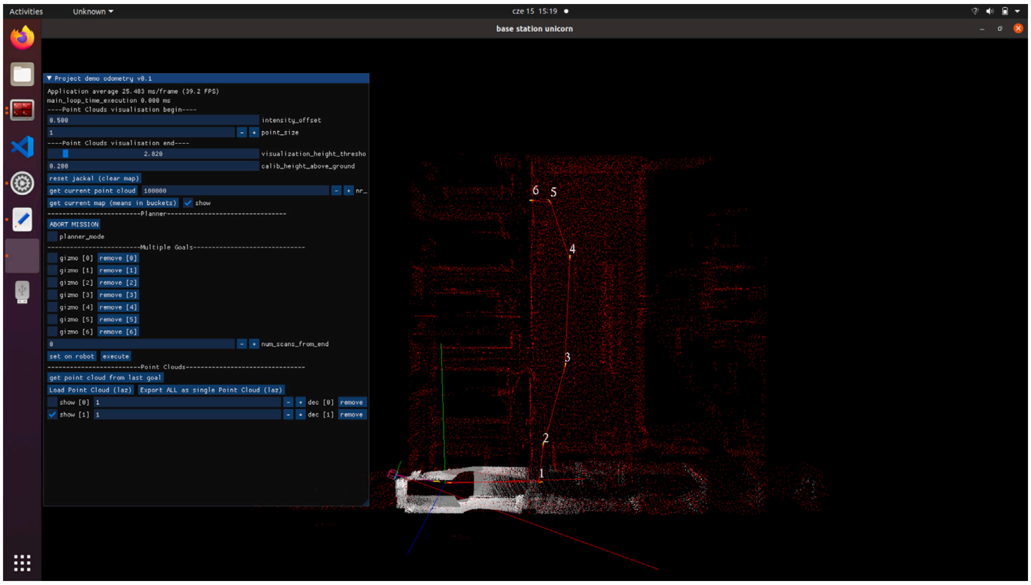
Thus, the NDT optimization problem is defined as the maximization of the likelihood function given in Eq. (23).

$$[\boldsymbol{R},\boldsymbol{t}]^{3\times4,*}_{W\leftarrow Lidar} = \max_{[\boldsymbol{R},\boldsymbol{t}]_{W\leftarrow Lidar}} \prod_{k=1}^{N} p(\boldsymbol{\Psi}([\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar}, \boldsymbol{P}^l_m)) \tag{23}$$

Furthermore, the optimization problem is equivalent to the minimization of the negative log-likelihood given in Eq. (24).

$$[\boldsymbol{R},\boldsymbol{t}]^{3\times4,*}_{W\leftarrow Lidar} = \min_{[\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar}} -\sum_{k=1}^{N} \log\left(p(\boldsymbol{\Psi}([\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar}, \boldsymbol{P}^l_m))\right) \tag{24}$$

The proposed NDT implementation is using already discussed point-to-point observation equation. The target value is $[0,0,0]^\intercal$ and the model function is $\boldsymbol{\Psi}([\boldsymbol{R},\boldsymbol{t}]^{3\times4}_{W\leftarrow Lidar}, \boldsymbol{P}^l_m) - \boldsymbol{\mu}$. The information matrix $\boldsymbol{\Omega}$ is calculated as an inverse of the covariance matrix from Eq. (20).



```
▼ Project demo odometry v0.1
Application average 25.483 ms/frame (39.2 FPS)
main_loop_time_execution 0.000 ms
----Point Clouds visualisation begin----
0.500                                   intensity_offset
1                              - +      point_size
----Point Clouds visualisation end----
            2.820                       visualization_height_thresho
0.200                                   calib_height_above_ground
reset jackal (clear map)
get current point cloud  100000    - +  nr_
get current map (means in buckets)  ☑ show
------------------------Planner-----------------------------------
ABORT MISSION
□ planner_mode
------------------------Multiple Goals----------------------------
□ gizmo [0] remove [0]
□ gizmo [1] remove [1]
□ gizmo [2] remove [2]
□ gizmo [3] remove [3]
□ gizmo [4] remove [4]
□ gizmo [5] remove [5]
□ gizmo [6] remove [6]
0                              - +      num_scans_from_end
set on robot  execute
------------------------Point Clouds------------------------------
get point cloud from last goal
Load Point Cloud (laz)  Export ALL as single Point Cloud (laz)
□ show [0] 1                   - + dec [0] remove
☑ show [1] 1                   - + dec [1] remove
```

Fig. 5. Base station of autonomous mobile robot Jackal. Graphical user interface shows layout (sparse red dots) of the environment perceived by elaborated SLAM, current local map (dense gray dots), path and goals.

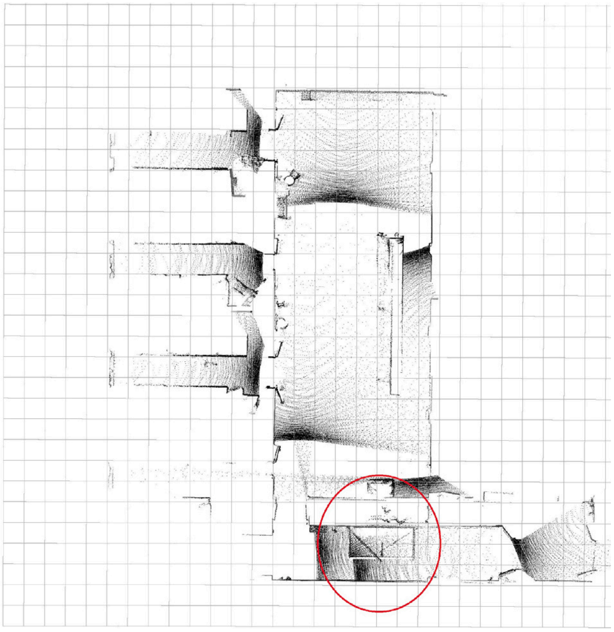


Fig. 6. Result of the MSAS EnRicH 2023 trial with marked most dangerous region of the environment (opened hatch to one level down, robot can be lost forever here). Grid 1 × 1 meter.

3. Laboratory tests

Laboratory tests were performed to verify the repetitiveness of the mission execution. The layout of the environment is shown in Fig. 2. Robot repeats the same mission (START/TURNING GOAL/FINISH) 20 times. All trajectories are marked by red dots. It can be seen that the system is capable follow predefined trajectory with the accuracy less than 10 centimeters (see zoomed bottom down rectangle in Fig. 2). Trajectory START to TURNING GOAL slightly differs from trajectory TURNING GOAL to FINISH. First robot is navigation forward. Once it reaches last goal of the trajectory (TURNING GOAL) it navigates backward to the initial pose (START).

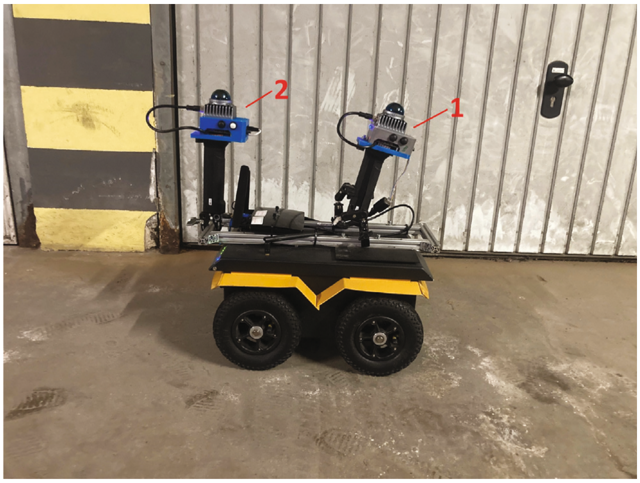


Fig. 7. System setup for accuracy, precision, robustness and repeatability assessment. 1: main LiDAR (3D map building and navigation), 2: physically mounted hand held mobile mapping system [20] for trajectory tracking and 3D map to ground truth data registration.

4. EnRicH (THE EUROPEAN ROBOTICS HACKATHON) trial

EnRicH trials are important for qualitative and quantitative evaluation of the autonomous mobile robots. Qualitative and quantitative evaluation is performed by judges observing trials, thus many teams were compared. The mobile mapping system presented in this paper received "EnRicH 2023 3D Mapping award", thus it was best in class according to its capacity for performing mission with highest level of autonomy. Fig. 3 shows intersection of the nuclear power plant and region 25 × 25 meters of EnRicH trial (red color). Working conditions during EnRicH trial are shown in Fig. 4. It can be seen that the surface is rather flat. The distance between obstacles is large. The lightning conditions are satisfactory. There are no dynamic obstacles. The first challenge is limited communication, thus robot should perform mission in fully autonomous mode. The second challenge is a negative obstacle marked by red circle in 3D map provided by robot during

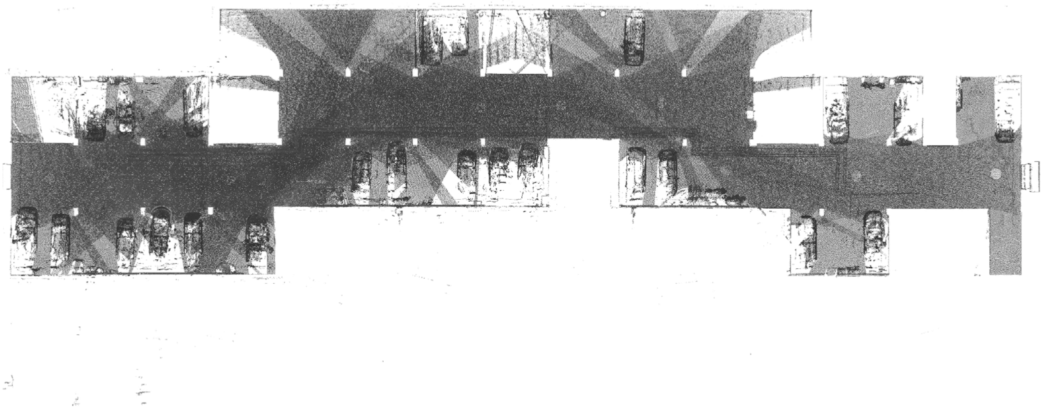


Fig. 8. Ground truth point cloud data acquired with TLS Z+F IMAGER 5010. Underground garage 100 × 25 [m], top view.

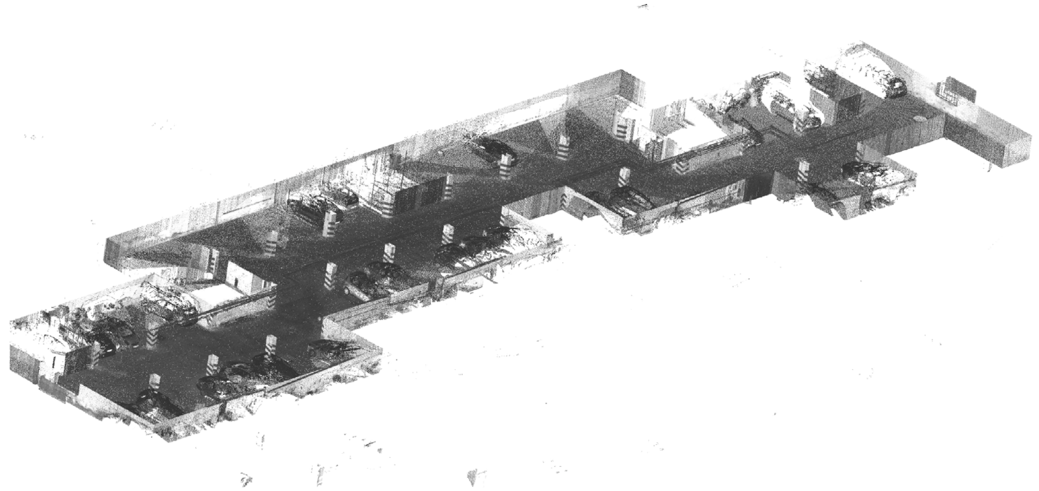


Fig. 9. Ground truth point cloud data acquired with TLS Z+F IMAGER 5010. Underground garage 100 × 25 [m], perspective view.

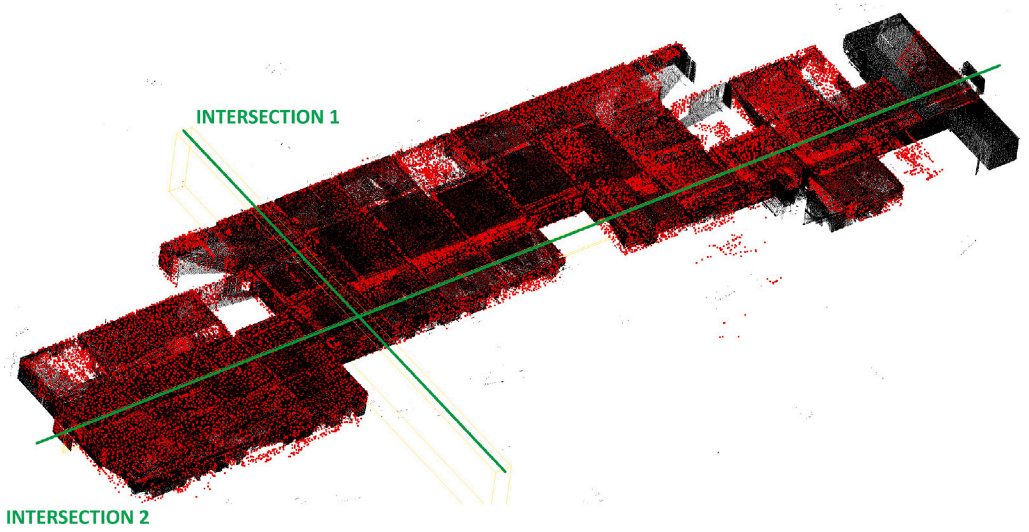


Fig. 10. 3D map accuracy assessment. Red color: 3D map obtained with proposed system, gray scale color: ground truth point cloud acquired with TLS Z+F IMAGER 5010.

autonomous drive close to it (Fig. 6). To reach such a performance robot should have robust localization. Another challenge is limited time up to 15 min for task execution. For this reason our robot moved at maximum speed of 2 m/s. Fig. 5 shows base station of autonomous mobile robot Jackal. Graphical user interface shows layout (sparse red dots) of the environment perceived by elaborated SLAM, current local map (dense gray dots), path and goals. Base station allows generating predefined trajectory that robot follows. Due to centimeter accuracy of the path execution, assumption of the non dynamic obstacles and accurate localization robot could perform mission repetitively during

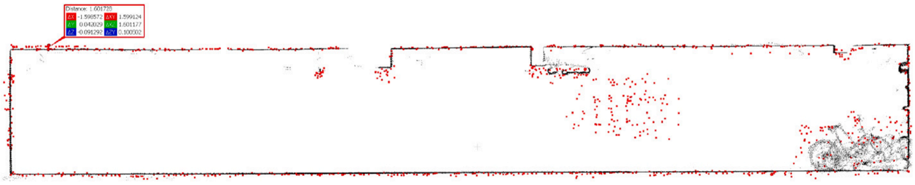


Fig. 11. Intersection 1 from Fig. 10. The maximum deviation of the map (red dots) to ground truth (black color) is ΔZ = 9.1 [cm].

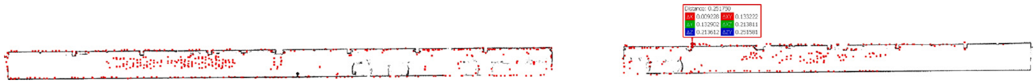


Fig. 12. Intersection 2 from Fig. 10. The maximum deviation of the map (red dots) to ground truth (black color) is ΔZ = 21.3 [cm].

entire 15 min trial. Robot delivered map (Fig. 6). Map is accurate on centimeter level.

5. Accuracy, precision, robustness and repeatability

Performance measures such as accuracy, precision, robustness and repeatability were performed with the mobile robot equipped additionally with physically mounted hand held mobile mapping system [20] shown in Fig. 7. Thanks to this it was possible to track the robot in indoor and outdoor environments. Two experiments were conducted: in underground garage and in the forest.

5.1. Map accuracy assessment (underground garage experiment)

Map accuracy assessment was performed with respect to ground truth data obtained with Terrestrial Laser Scanner (TLS) Z+F IMAGER 5010. This TLS is industrial grade measurement device assuring millimeter accuracy and precision of the registered point cloud data. Fig. 8 shows top view of the underground garage 100 × 25 [m]. Fig. 9 shows perspective view of this scene. The goal was to build 3D map with proposed system. The resulting RMS is 0.0937, thus the system is capable mapping with decimeter accuracy. Figs. 10–12 show 3D map accuracy assessment. Two intersections were chosen manually and the maximum deviation of the map to ground truth was measured. It is ΔZ = 9.1 cm for intersection 1 and ΔZ = 21.3 cm for intersection 2.

5.2. Navigation performance measures (underground garage experiment)

Mobile robot from Fig. 7 was deployed in underground garage shown in Fig. 7. The navigation system was set up to automatically traverse the garage from START → END and backward via certain number of goals. The performance measures were performed based on mobile mapping data registration (see Fig. 7) [20] to ground truth (see Figs. 8, 9). Mobile robot was navigating ten times repetitively from START → END and backward in underground garage. The maximum deviation of the trajectories is around 0.3 [m]. It can be seen the centimeter precision/repetitiveness of the START → END and END → START trajectories. It means the navigation system is capable to repeat the mission with decimeter accuracy. The repetitiveness in structured environment is confirmed.

5.3. Navigation performance measures (forest experiment)

This experiment was performed in forest environment (see Figs. 16 and 17). Mobile robot from Fig. 7 was deployed in START position. Similarly to previous experiment the navigation system was set up to automatically traverse the same path from START → END and backward via certain number of goals. The maximum deviation of the trajectories is around 52 [cm]. It can be seen the centimeter precision/repetitiveness of the START → END and END → START trajectories. It means the navigation system is capable to repeat the mission with decimeter accuracy. The repetitiveness and robustness in harsh environment is confirmed.

6. Software and hardware requirements

This project is designed assuming minimal need for 3rd party libraries. Basic functionality requires:

- https://github.com/JanuszBedkowski/observation_equations.git (observation equations)
- https://github.com/nlohmann/json.git (IO)
- https://github.com/LASzip/LASzip.git (IO)
- https://github.com/xioTechnologies/Fusion.git (IMU filter)
- https://gitlab.com/libeigen/eigen.git (math operations)
- https://github.com/ocornut/imgui (graphical user interface)

The software requires ROS1. GPU is not mandatory since all calculations are done on CPU (see Fig. 19). This software was successfully tested also on Raspberry PI 4B (8 GB RAM). Thanks to dr MichałPełka the upgrade to ROS2 is maintained at https://github.com/michalpelka/msas_enrich_2023.

7. Impact and applications

This software is dedicated for mobile robotic domain. It is first end to end solution for nuclear power plant inspection available in literature. Moreover current end users cover the following applications:

- culture heritage (creating virtual trips),
- urban search and rescue (research on creating a digital twin of training facilities),
- ground truth for AGV and forklifts,
- precision forestry (research on volume calculation and AI for tree classification),
- agricultural robotics (creating maps for navigation),
- underground mining (research on safety application, digital twin),
- critical infrastructure inspection.

8. Conclusions

This paper shows the novel approach for nuclear facility inspection. Autonomous mobile robot is equipped with non repetitive scanning pattern Lidar Livox Mid 360. This Lidar can cover 99% of the field of view within one second, thus robot can retrieve full 3D model of the environment even without motion. The implementation is based on feature less single core SLAM that can localize robot at 100 Hz rate. This method is the candidate for real time application since the calculation time is bounded and it uses only one core of Intel Celeron CPU G1840 2.8 GHz. It fuses Normal Distributions Transform and motion model for simultaneous map building and current pose

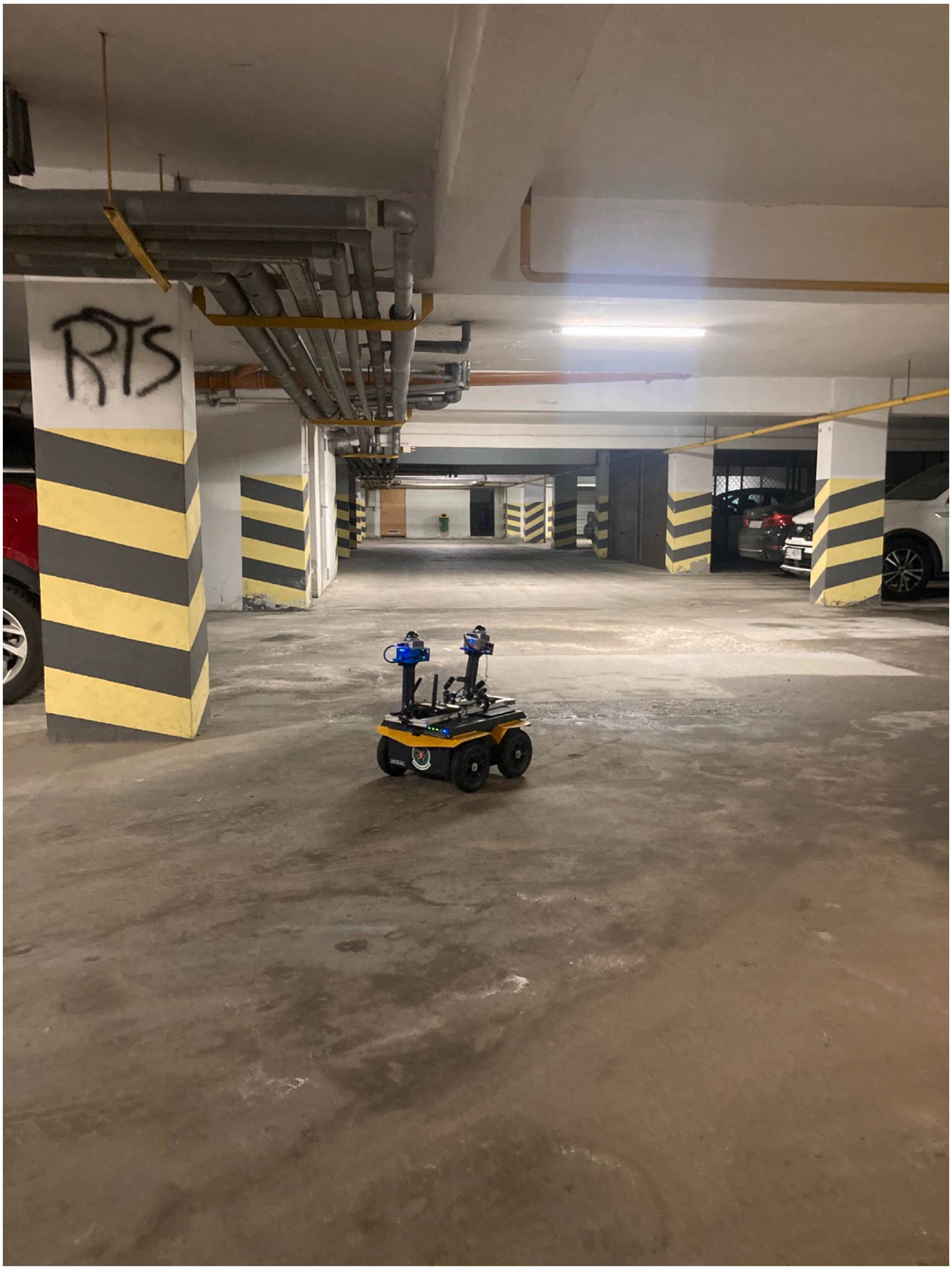


Fig. 13. Mobile robot during experiment in underground parking.

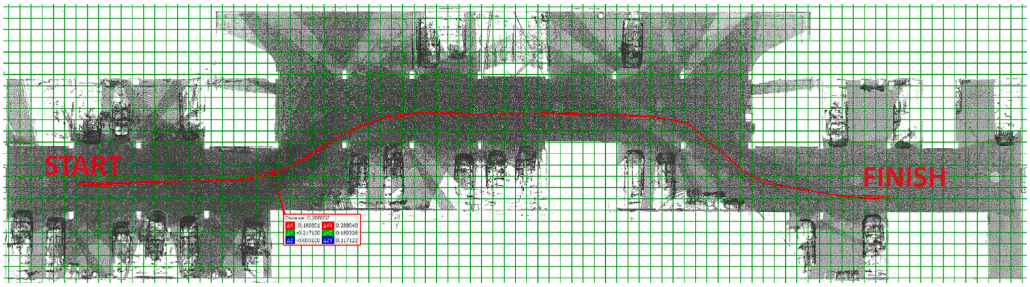


Fig. 14. Result of navigation performance measures. Mobile robot was navigating ten times repetitively START → END and backward. The maximum deviation between forward–backward trajectories is around 0.3 m. Gray-scale: map of the garage, red: navigation trajectories, green: grid 1 × 1 [m].

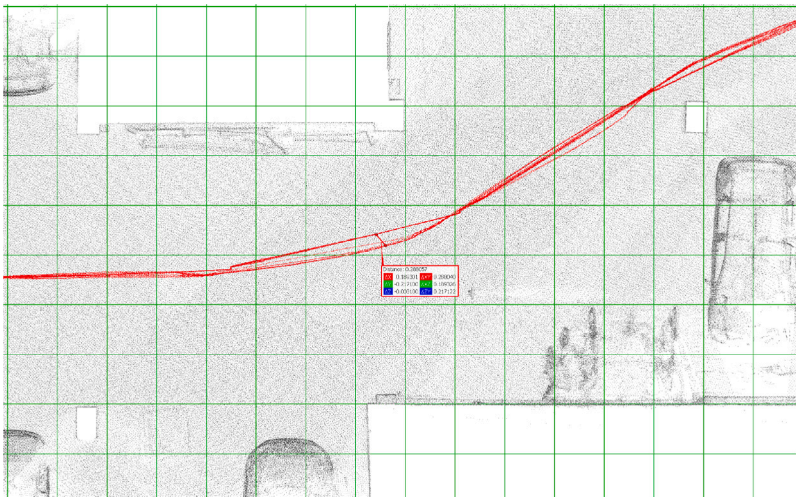


Fig. 15. Zoomed maximum deviation between forward–backward trajectories from Fig. 14.

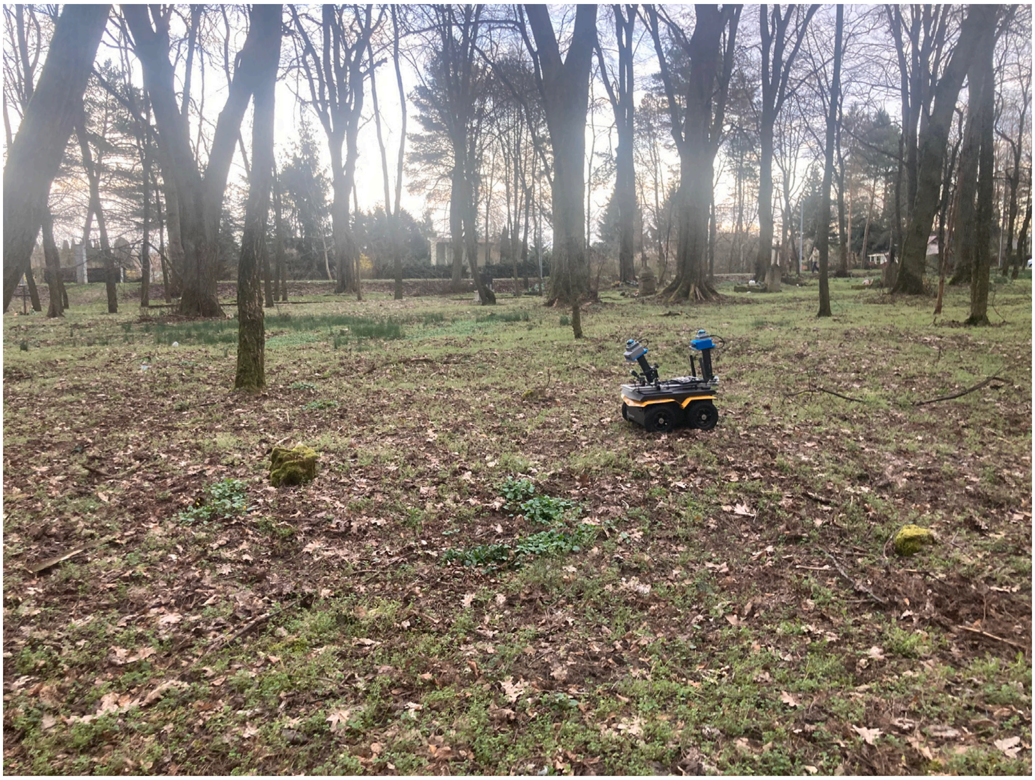


Fig. 16. Mobile robot during experiment in forest.

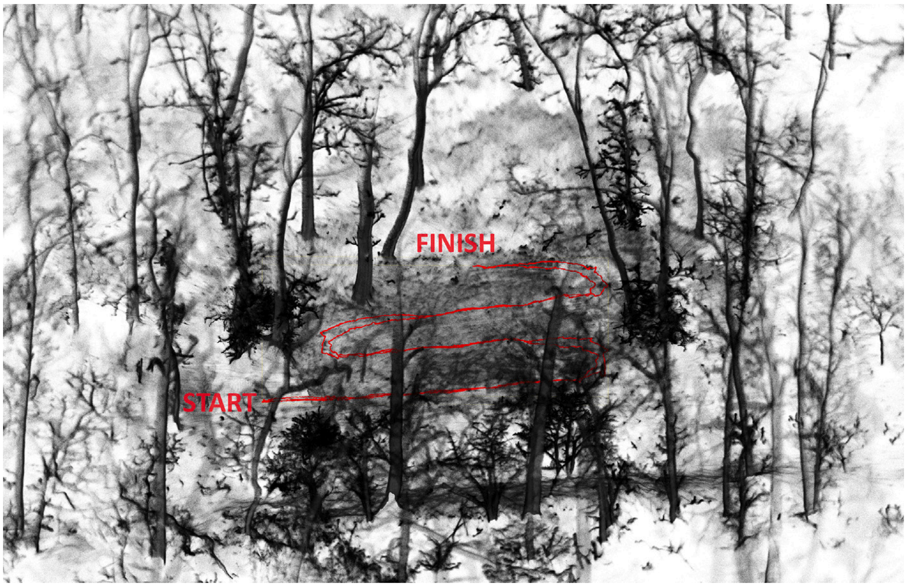


Fig. 17. Forest experiment. Mobile robot from Fig. 16 was deployed in the forest. It was navigating ten times repetitively from START → FINISH and backward. This experiment proofs the robustness of the system in the forest environment. Black: point cloud data, red: robot trajectory, both obtained with mobile mapping system attached to the robot using methodology elaborated in [20].

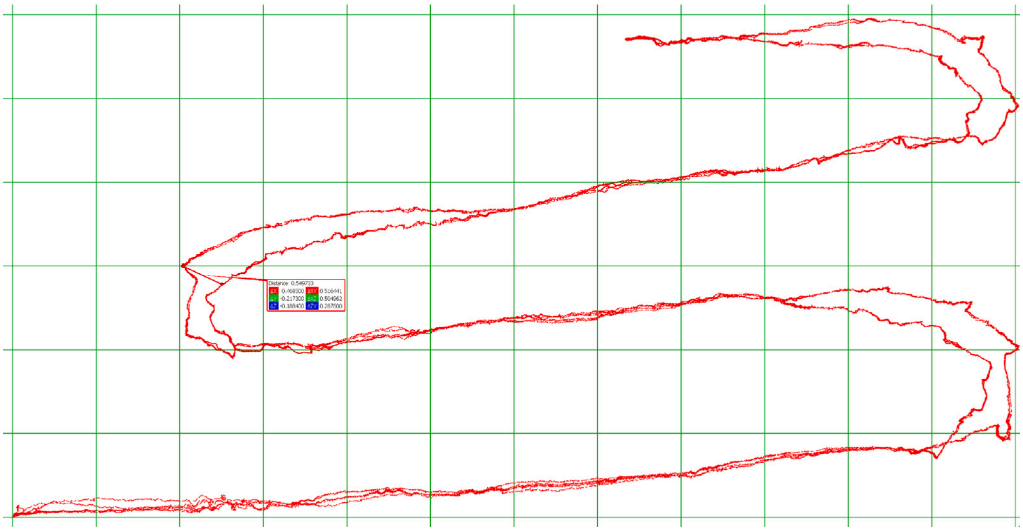
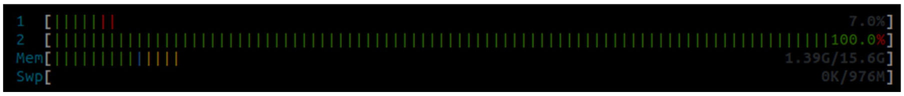


Fig. 18. Zoomed trajectory from forest experiment (Fig. 17). The maximum deviation between forward–backward trajectories is $\Delta XY \approx 52$ [cm]. Green: grid 1 × 1 [m].



```
  1  [||||||||                                                                      7.0%]
  2  [|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||100.0%]
  Mem[||||||||||||||                                                           1.39G/15.6G]
  Swp[                                                                             0K/976M]
```

Fig. 19. CPU and RAM usage during forest experiment. It can be seen 7% usage of first core, 100% usage of second core and 1.4G of RAM.

estimation. It was tested both (i) during EnRicH 2023 — the European robotics hackathon (MSAS team), (ii) laboratory conditions. The system is capable providing map on centimeter level. It can navigate in this map also on centimeter level what was evaluated in laboratory and realistic conditions. Future work will be a loop closure and a multi robot system. This work is dedicated for researchers that would test this approach. Thus, open source project is provided at https://github.com/JanuszBedkowski/msas_enrich_2023.

Disclaimer

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

CRediT authorship contribution statement

Janusz Będkowski: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Kümmerle Rainer, Grisetti Giorgio, Strasdat Hauke, Konolige Kurt, Burgard Wolfram. G2o: A general framework for graph optimization. In: 2011 IEEE international conference on robotics and automation. 2011, p. 3607–13.

[2] Dellaert Frank, Contributors GTSAM. borglab/gtsam. 2022.

[3] Sola Joan, Deray Jeremie, Atchuthan Dinesh. A micro Lie theory for state estimation in robotics. 2021.

[4] Agarwal Sameer, Mierle Keir, Team The Ceres Solver. Ceres Solver. 2022.

[5] Bedkowski Janusz. Large-scale simultaneous localization and mapping. In: Cognitive intelligence and robotics, Springer; 2022.

[6] Meurer Aaron, Smith Christopher P, Paprocki Mateusz, Čertík Ondřej, Kirpichev Sergey B, Rocklin Matthew, et al. SymPy: symbolic computing in python. PeerJ Comput Sci 2017;3:e103.

[7] Guennebaud Gaël, Jacob Benoît, et al. Eigen v3. 2010, http://eigen.tuxfamily.org.

[8] Xu Wei, Cai Yixi, He Dongjiao, Lin Jiarong, Zhang Fu. FAST-LIO2: Fast direct LiDAR-inertial odometry. 2021, CoRR, abs/2107.06829.

[9] Bai Chunge, Xiao Tao, Chen Yajie, Wang Haoqian, Zhang Fang, Gao Xiang. Faster-LIO: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels. IEEE Robot Autom Lett 2022;7(2):4861–8.

[10] Dellaert Frank, Kaess Michael. Factor graphs for robot perception, vol. 6, Foundations and Trends in Robotics; 2017.

[11] Madgwick Sebastian. An efficient orientation filter for inertial and inertial / magnetic sensor arrays. 2010.

[12] Besl PJ, McKay Neil D. A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell 1992;14(2):239–56.

[13] Sunderhauf N, Protzel P. Switchable constraints for robust pose graph SLAM. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. 2012, p. 1879–84.

[14] Magnusson Martin, Lilienthal Achim J, Duckett Tom. Scan registration for autonomous mining vehicles using 3D-NDT. J Field Robot 2007;24(10):803–27.

[15] Kuemmerle R, Grisetti G, Strasdat H, Konolige K, Burgard W. g2o: A General Framework for Graph Optimization. In: Proc. of the IEEE int. conf. on robotics and automation. 2011.

[16] Karakuş Sıddıka, Gündoğan Halit. C. 2010, p. 367–77.

[17] Horn Berthold KP. Closed-form solution of absolute orientation using unit quaternions. J Opt Soc Amer A 1987;4(4):629–42.

[18] Horn Berthold KP, Hilden Hugh M, Negahdaripour Shahriar. Closed-form solution of absolute orientation using orthonormal matrices. J Opt Soc Amer A 1988;5(7):1127–35.

[19] Rusu RB, Cousins S. 3D is here: Point cloud library (PCL). In: Robotics and automation (ICRA), 2011 IEEE international conference on. 2011, p. 1–4.

[20] Będkowski Janusz. Open source, open hardware hand-held mobile mapping system for large scale surveys. SoftwareX 2024;25:101618.