Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
June 2024, vol. 29, no. 1, pp. 78–84
https://doi.org/10.2478/acss-2024-0010
https://content.sciendo.com

Heterogeneous Map Fusion from Occupancy Grid Histograms for Mobile Robots

Aleksandrs Sisojevs[1*], Aleksandrs Korsunovs[2], Martins Banis[3], Vilnis Turkovs[4], Reinis Cimurs[5]
[1]*Riga Technical University, Riga, Latvia*
[2–5]*Robotic Solutions Ltd., Riga, Latvia*

***Abstract* – With the increase in the capabilities of robotic devices, there is a growing need for accurate and relevant environment maps. Current robotic devices can map their surrounding environment using a multitude of sensors as mapping sources. The challenge lies in combining these heterogeneous maps into a single, informative map to enhance the robustness of subsequent robot control algorithms. In this paper, we propose to perform map fusion as a post-processing step based on the alignment of the window of interest (WOI) from occupancy grid histograms. Initially, histograms are obtained from map pixels to determine the relevant WOI. Subsequently, they are transformed to align with a selected base image using the Manhattan distance of histogram values and the rotation angle from WOI line regression. We demonstrate that this method enables the combination of maps from multiple sources without the need for sensor calibration.**

***Keywords* – Heterogeneous map fusion, occupancy grid histograms, mapping for mobile robots.**

I. Introduction

For robots to operate freely in the environment, they need to gather information about their surroundings. Through sensor readings, it is not only possible to understand the position of the robot relative to other objects but also to record this information and map the environment. This recorded map is crucial for generating motion plans. Therefore, it should be as precise and up-to-date as possible to optimise the robot motion and avoid safety-critical situations [1].

A variety of methods exist that record and map the environment by employing different kinds of sensors. Robot motion estimation can use dead reckoning from the robot's wheels to gain the travelled distance, or IMUs can be used to estimate the scale of the motion [2]. In static locations, key point markers can help triangulate the exact location, or a satellite signal can be used in GPS-enabled environments [3]. This information is then combined with various distance detection sensors, such as depth cameras, radar, LiDAR, sonar, or similar, and used for simultaneous mapping and localisation (SLAM) of the environment and the positioning of the robot in it [4]. This has proven to work well in various applications and has allowed the use of SLAM systems not only to gain an initial estimation of the robot's surroundings but also to update it throughout its lifecycle [5].

However, using an unreliable source of distance information for mapping tasks can result in inaccurate maps, leading to subpar planning of robot motion. For instance, sonar and radar data recordings have to deal with noisy distance estimations. Stereo cameras are susceptible to lighting conditions and the number of features in the scene. Laser-enabled depth cameras have difficulty registering glass structures and have a limited field of view. 2D laser sensors can only record a single plane, and 3D sensors still suffer from not being able to detect light-absorbent and transparent obstacles. Therefore, redundancy in mapping tasks plays a crucial role in enabling the robots to execute their motion in a safe and efficient manner. Sensor fusion can be employed to obtain such redundancy at runtime but requires prior knowledge about the sensor types and calibration to ensure proper functionality [6]. This can be cumbersome for systems that may include multiple sensors of different types. Each sensor combination requires different calibration before the start of the program. Alternatively, redundancy can be obtained by merging produced maps from different data sources. Elements that are not recorded from one data source can appear in maps of other types without the need for calibration and a priori knowledge of sensor interactions. Using post-processing of already obtained maps allows for the use of a dynamic number of sensor sources with a simple setup process for SLAM systems. Therefore, in this paper, we propose a method of map merging from heterogeneous sources for mobile robot navigation.

II. Related Work

The task of automating the creation and merging of maps of the environment obtained from different heterogeneous sources is relevant in various fields of robotics. A system, described in [7], enables a mobile robot to acquire knowledge about its surroundings and accomplish tasks by understanding the environment. The mobile robot gathers information from camera sensors about the environment. Through a machine learning module, the robot can identify objects within the surroundings and determine its semantic positioning in the

[*] Corresponding author's e-mail: Aleksandrs.Sisojevs@rtu.lv

Article received 2023-12-18; accepted 2024-07-22

environment. However, map retention and reusability are questionable in such situations due to possible changes in the environment. In [8], a map of an environment is obtained using mapping data generated by an independent cleaning robot during its initial cleaning operation. The method describes the transferability of mapping and behaviour information to an independent cleaning robot to execute a behaviour linked to the label during a subsequent cleaning task; however, it does not address subsequent map updates from new sources. The MRSLAM (multi-robot SLAM) technique, described in [9], primarily includes a resilient map merging algorithm and a decision-making algorithm that governs field agents. Here, a single map is obtained in real time from multiple robots in the same environment but requires similarity in mapping sensor sources.

In [10], a method is introduced for constructing a 3D point cloud map using laser-measured data as the robot navigates its surroundings. To balance the trade-off between data size and terrain accuracy, a terrain-adaptive density mapping technique is employed, utilising local curvatures as criteria for simplification. This adaptive density mapping technique is then integrated into the map merging framework, enhancing both matching speed and accuracy.

Paper [11] introduces a novel algorithm for map fusion in a multi-robot cooperative SLAM framework. This algorithm is designed to create a map of an unfamiliar environment by utilising multiple agile exploration robots. Two major challenges include the accumulation of redundant information in the evolving map over time and the occurrence of mapping errors. In [12], a comprehensive overview is provided of innovative strategies aimed at addressing these challenges by eliminating redundant information from the map and correcting mapping errors.

In [13], real-time sensor fusion for SLAM is proposed through trajectory matching. Although the algorithm is lightweight, it does not consider the integration of maps obtained from different robots at different times. Direct sensor fusion from LiDAR and monocular camera sources is introduced in [14]. 3D LiDAR and sensor fusion is depicted in [15], [16]. Other methods involve sensor fusion between LiDAR, RADAR, ultra-wideband, monocular RGB, RGB-D, and other sensors [17], [18], [6]. While these methods perform well during the runtime and real-time collection of mapping data, they do not address the challenge of matching mapping data taken at different times and possibly from different robot devices. Such map fusion needs to account for synchronisation in both time and space across various sensors [19], [20].

In [21], post-processing map fusion is presented, involving multiple mapping sources, one of which is ground truth, and a prior map of the environment. An entirely robot-detected map merging approach is proposed in [22], where LiDAR fusion with monocular RGB SLAM is achieved through keypoint matching. In this method, the same keypoints need to be extracted in all map sources to align the mapping data. In [23], a method is proposed for merging top-down views of the environment with 3D LiDAR data, which is rarely available for indoor robot-enabled locations. In [24], a method of merging heterogeneous maps in a post-processing step is proposed by executing a set of motions by different robots in the same map, information that is not always available or easy to reconstruct. Direct pixel matching and subsequent clustering are proposed in [25].

In this article, we propose an approach for merging 2D maps of the environment based on the creation and use of statistical histograms, which, in turn, rely on occupancy grid pixels. The outlined approach is iterative, involving transformations on this data. As such, the method is generally sensor, recording time, and robot trajectory agnostic, allowing it to be updated whenever a new map of the environment becomes available.

III. Proposed Approach

The proposed approach utilises occupancy grid maps from multiple sources to generate a single, final map of the environment. Mapping is conducted for the entire area with each data source, and the recorded occupancy grids are employed in post-processing to derive the final map. In general, the map is indifferent to the sensors used in its collection and the robots with which it was recorded, as long as the final result is an occupancy grid of the same scale. The objective of the proposed approach is to merge the provided maps in a manner that maximises the alignment of map elements.

We assume that the recorded maps represent the same environment and that their mapping is complete, i.e., they have captured the entirety of the area under consideration. We designate any map from an arbitrary source as the base map, denoted as M_b. For optimal performance, this map should exclude elements beyond the specified area, such as those recorded through windows or artefacts resulting from mirrors and glass doors. An example of such a map recorded from visual SLAM is shown in Fig. 1.

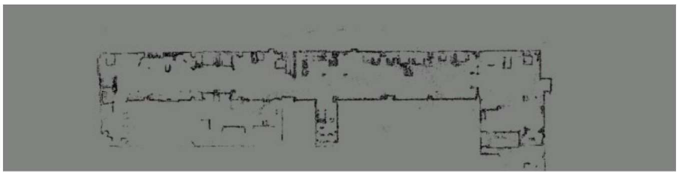

Fig. 1. Visualisation of the base map M_b from visual SLAM source.

Other maps allow for the presence of artefacts related to scanning open spaces behind windows and doors. Any other map from an arbitrary source is designated as M_s, where $s \in S$ as long as $s \neq b$. Here, S is all the available map sources. An example of such a map recorded from a 2D LiDAR source is shown in Fig. 2.

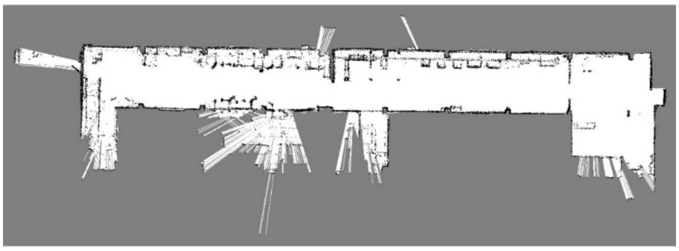

Fig. 2. The visualisation of map M_s from a 2D LiDAR source.

Elements of the base map M_b will remain fixed in the proposed approach as we will use it to align maps from other sources with it. The proposed approach consists of the following steps:

Step 1. To establish the bounds of the relevant map data, we utilise pixel histogram information. Based on the data from M_b, histograms of pixel distribution are created along O_x and O_y coordinate axes. As map data can be noisy, and single-pixel values might not accurately represent the actual position of the recorded pixel, we enhance the granularity of the histogram to encompass p consecutive pixels within a single bin. The bin intervals of pixel count values along the O_x and O_y coordinates are collected as follows:

$$bin_x(i) = \sum_{v=i\cdot p}^{v=i\cdot p+p} x_v ; \tag{1}$$

$$bin_y(j) = \sum_{v=j\cdot p}^{v=j\cdot p+p} y_v , \tag{2}$$

where $bin_x(i)$ – i-th interval along O_x, $bin_y(j)$ – j-th interval along O_y. i is the number of bins along x axis and j is the number of bins along y axis. x_v and y_v represent the pixel value at the coordinate, where the value is 1 if the pixel is occupied and 0 if not.

Based on the histograms in bin_x and bin_y, a window of interest (WOI) with dimensions W×H is identified, within which the map is situated. Each bin value in bin_x and bin_y is iteratively examined, and the first and last values that exceed the thresholds $\delta_{\text{hist}(x)}$ and $\delta_{\text{hist}(y)}$, respectively, are chosen as the starting and ending values for the WOI for each axis. An example of histograms and the identified WOI, indicated by a blue frame, is shown in Fig. 3.

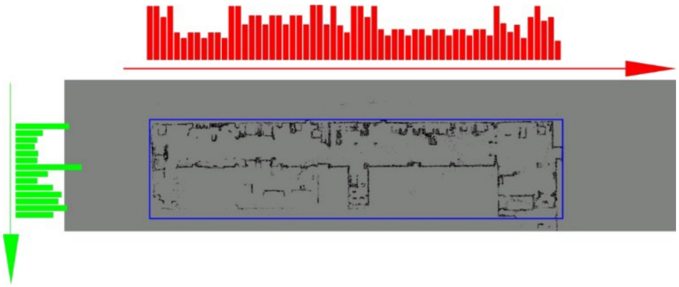

Fig. 3. The visualisation of M_b with histograms and window of interest. Window of interest is visualised by the blue rectangle. Red histogram shows bins with pixel values along x axis. Green histogram shows bins with pixel values along y axis.

The final WOI is recorded as a rectangle formed of minimal and maximal x and y coordinates as follows:

$$WOI_{\min} = \left[p\cdot i_{\min}, p\cdot j_{\min} \right]; \tag{3}$$

$$WOI_{\max} = \left[p\cdot i_{\max} + p, p\cdot j_{\max} + p \right], \tag{4}$$

where $i_{\min}$ and $j_{\min}$ are the indices of the first valid bin that exceed $\delta_{\text{hist}(x)}$ and $\delta_{\text{hist}(y)}$, respectively. $i_{\max}$ and $j_{\max}$ are the indices of the last valid bin that exceed $\delta_{\text{hist}(x)}$ and $\delta_{\text{hist}(y)}$, respectively.

Step 2. For any map from an arbitrary source M_s, the same histogram calculation is performed as in Step 1. The histograms obtained from M_s using (1) and (2) are compared to the histograms from the base map M_b by calculating the Manhattan distances:

$$d_x = \sum_i \left| bin_x^b(i) - bin_x^s(i) \right| ; \tag{5}$$

$$d_y = \sum_j \left| bin_y^b(j) - bin_y^s(j) \right| , \tag{6}$$

where d_x is the Manhattan distance between histograms along O_x, d_y is the Manhattan distance between histograms along O_y. $bin_x^b(i)$ and $bin_y^b(j)$ represent the i-th and j-th histogram bins in base map, $bin_x^s(i)$ and $bin_y^s(j)$ represent the i-th and j-th histogram bins in source map.

The position at which the sum of distances $d_x + d_y$ is minimised is considered the optimal position for the WOI in the source map. This allows for the translation of the source map to align it with the base map.

Step 3. To estimate the rotation angle between WOI of M_b and M_s, we compute linear regression functions for the data for each map, respectively. For the WOI of M_b, the coefficients of the linear function are expressed as:

$$y = \alpha_b x + \beta_b , \tag{7}$$

and are calculated as follows:

$$\begin{bmatrix} \alpha_b \\ \beta_b \end{bmatrix} = \begin{bmatrix} \sum_k x_k^2 & \sum_k x_k \\ \sum_k x_k & k \end{bmatrix} \cdot \begin{bmatrix} \sum_k x_k \cdot y_k \\ \sum_k y_k \end{bmatrix} . \tag{8}$$

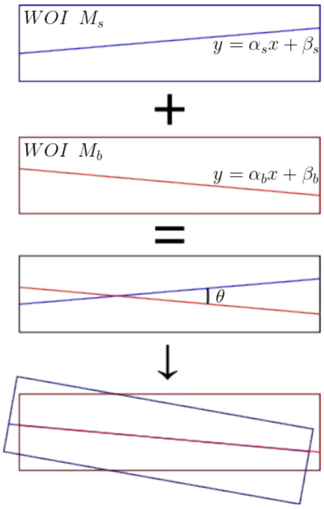

Fig. 4. The visualisation of the rotation procedure of the window of interest of M_s relative to the window of interest of M_b.

For the WOI of M_s, the coefficients of the linear function are expressed as:

$$y = \alpha_s x + \beta_s , \tag{9}$$

and are calculated analogously as for M_b with (8).

The angle between the linear functions (7) and (9) represents the rotation angle of M_b relative to M_s and is computed as follows:

$$\theta = \arctan\left(\alpha_s\right) - \arctan\left(\alpha_b\right), \tag{10}$$

where θ is the rotation angle. Figure 4 illustrates the operation of the map rotation procedure using the WOI of each respective map.

Step 4. After performing the translation and rotation transformations, the source map M_s is updated and becomes M_s'. We evaluate if the translation and rotation are sufficient to fulfil the map fusion criteria. For this purpose, the equation results from (5), (6), and (10) must fall under a given threshold for each respective value:

$$\begin{cases} d_x < \varepsilon_x \\ d_y < \varepsilon_y \, , \\ \theta < \varepsilon_\theta \end{cases} \tag{11}$$

where ε_x and ε_y are threshold values for the translation coefficients and ε_θ is the threshold value for the rotation angle.

If the condition is not met, Steps 1 to 3 are repeated for the new values of M_s'. Execution of the method is completed once the criteria in (11) are met or when the iteration count becomes larger than a pre-set value:

$$iter_N > iter_{\max}\, , \tag{12}$$

where $iter_N$ is the N-th iteration of the method, and $iter_{\max}$ is the maximum number of iterations.

Full method implementation for the map M_s with pre-computed values for M_b is described in Algorithm 1.

Algorithm 1. Map Fusion Algorithm

```
$iter_N = 0$                                   ▷Set iteration count
while $iter_N > iter_{\max}$ do
    Step 1:
    Calculate histograms with (1) and (2)
    Obtain WOI with (3) and (4)
    Step 2:
    Calculate Manhattan distance with (5) and (6)
    Translate WOI by $d_x$ and $d_y$
    Step 3:
    Calculate linear function coefficients with (8)
    Calculate rotation angle with (10)
    Rotate WOI by $\theta$
    Step 4:
    if (11) is True then
        Map fusion complete
        Break
    else
        $iter_N = iter_N + 1$
    end if
end while
```

IV. Experiments

To validate the proposed method, we conducted experiments by fusing maps recorded by a mobile robot SLAM system from multiple sources. The base map utilises a visual SLAM recorded map, as shown in Fig. 1. The map fused with the base map is obtained from a 2D LiDAR SLAM and is visualised in Fig. 2. During the experiment, two iterations were carried out to move and rotate data from M_s relative to M_b. The result of the map merging can be seen in Fig. 5. As observed, the data from M_s coincide with the data from M_b where possible.

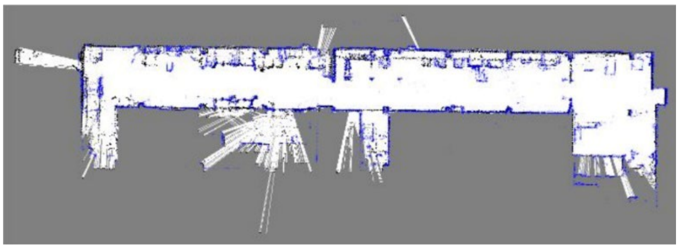

Fig. 5. The visualisation of map fusion. The M_b is shown in blue colour pixels overlaid over the M_s map.

A. Validation on SLAM Maps

To further evaluate the validity of the approach, a mapping of real-world environments was carried out in two distinct industrial environments. We refer to these as *exp*1 and *exp*2, respectively. For both of these experiments, the base map was recorded using a visual SLAM and the source map with LiDAR SLAM.

For *exp1* the M_b is visualised in Fig. 6. The M_s is depicted in Fig. 7. The final merged map, after executing the algorithm, is shown in Fig. 8. Here, the overlapping pixels are depicted in blue colour. The pixels from the M_b are shown with red, and pixels from M_s are shown in green pixels. The final map is a combination of both map information that adds additional information from both SLAM sources. We also record the number of pixels generated in the final map. Only the pixel values that represent occupied grids are used for this purpose, as they are the basis for map fusion. Additionally, we show the added data in pixel count that appears from visual and LiDAR SLAM separately. The data is given in Table I. We can see that in *exp*1 we gain an overlap of map data of 92.2 %. The visual SLAM data gives an additional 2.3 % amount of data that would not be recorded using just LiDAR SLAM. LiDAR SLAM, however, accounts for 5.5 % of additional data that would not be obtained by using only visual SLAM.

The M_b recorded with a visual SLAM for *exp*2 is visualised in Fig. 9. The M_s recorded with a LiDAR SLAM is visualised in Fig. 10. The final, combined map is visualised in Fig. 11. Blue pixels represent the overlapping pixels, red represents the pixels from M_b, and green from M_s. For *exp*2, the pixel coverage data is given in Table II. Here, even more data can be obtained in the final map by fusing multiple sources. 9 % of unique pixel values account for coming entirely from the visual SLAM source and 21 % uniquely from the LiDAR SLAM source.

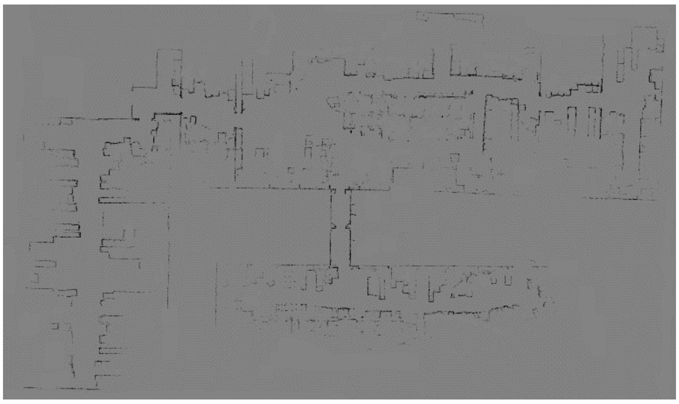

Fig. 6. Recorded map from visual SLAM for *exp*1.

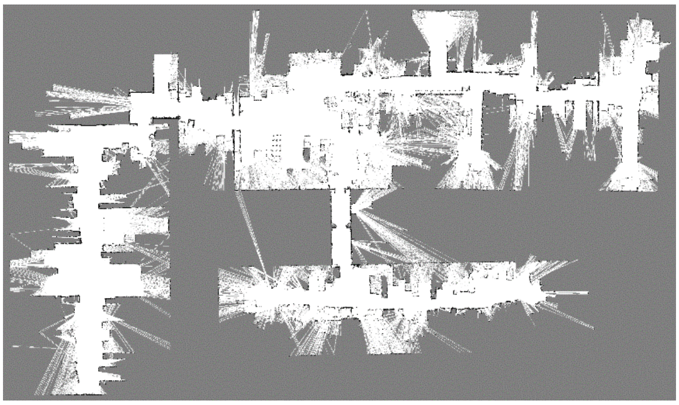

Fig. 7. Recorded map from LiDAR SLAM for *exp*1.

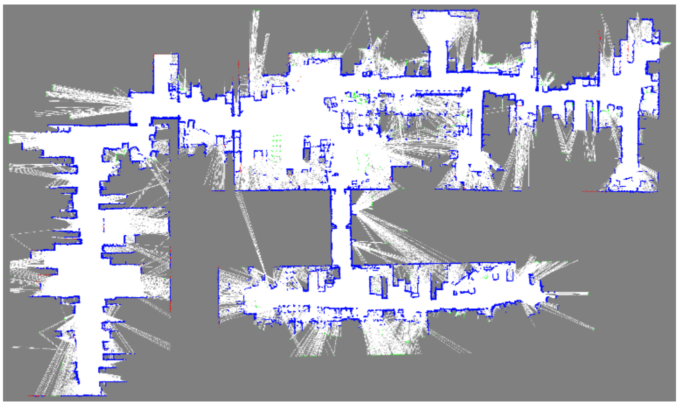

Fig. 8. Final result of fused map for *exp*1. Blue pixels represent an overlap of visual and LiDAR SLAM maps. Red pixels represent the visual SLAM pixels, and green – LiDAR SLAM pixels.

TABLE I

MAP PIXEL COVERAGE IN EXP1

	Visual SLAM	LiDAR SLAM	Overlap	Total
Nr. of Pixels	4250	10 166	169 804	184 220
% of Pixels	2.3 %	5.5 %	92.2 %	-

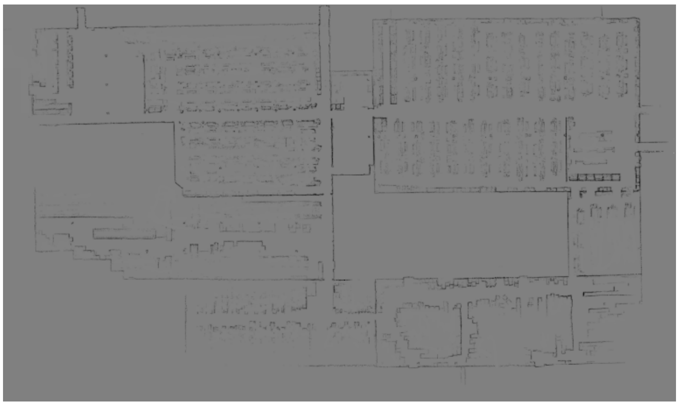

Fig. 9. Recorded map from visual SLAM for *exp*2.

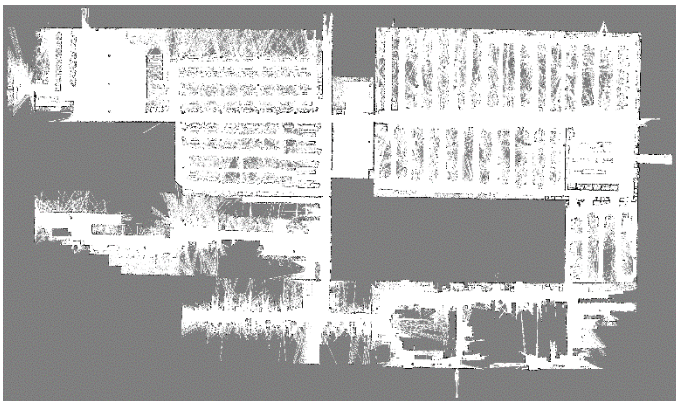

Fig. 10. Recorded map from LiDAR SLAM for *exp*2.

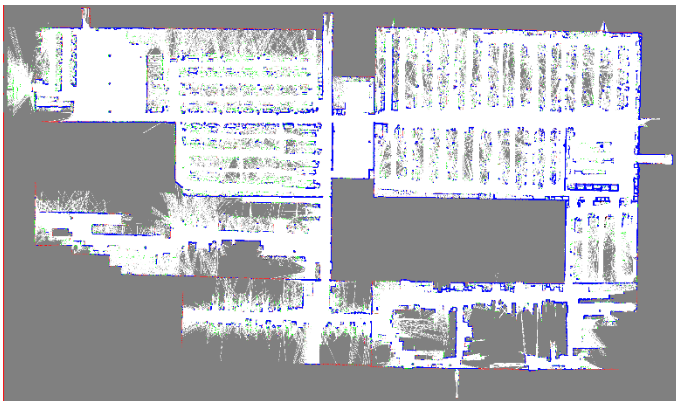

Fig. 11. Final result of fused map for *exp*2. Blue pixels represent an overlap of visual and LiDAR SLAM maps. Red pixels represent the visual SLAM pixels, and green – LiDAR SLAM pixels.

TABLE II

MAP PIXEL COVERAGE IN EXP2

	Visual SLAM	LiDAR SLAM	Overlap	Total
Nr. of Pixels	40 640	94 241	314 396	449 277
% of Pixels	9 %	21 %	70 %	-

V. Conclusion

The experimental results demonstrate the effectiveness of the proposed approach on real examples of maps obtained from robot sensors. It is possible to iteratively update the map translation and rotation to fuse maps generated from different sources. The proposed method is agnostic to the map generation method as long as recorded information is preserved in the form of an occupancy grid. Additionally, it is possible to extend the method to use multiple maps by independently fusing them with the base map information. This means that the proposed method does not require intrinsic or extrinsic calibration between sensors and allows the fusion of maps obtained from different robots as well as at different times. Using histograms as the basis for map region estimation provides a lightweight solution for estimating map bounds and fusing such information.

However, the proposed approach relies on several assumptions, one of which is that the recording pixel size is equal across multiple SLAM tasks. This allows for the use of the same scale assumption in the current implementation of the proposed method. Another assumption is that the mapped environment is complete, and the fused environment is fully mapped to the best possible extent. However, it might be possible that only a part of the map is available for some sources, making it difficult to obtain a proper WOI in such cases.

Here, more intelligent methods for estimating relevant WOI patches would need to be implemented, such as estimation by pixel overlap or employing Hough Transforms to estimate key line overlaps. Another issue to consider is merging maps from lifelong mapping systems in dynamic or changing environments. In this case, the map age could be taken into account, and occupancy grid pixels corresponding to newer maps could be weighted. This would enable the use of older maps in the environment to estimate missing elements in subsequent mapping steps and update the possible base map information through lifelong mapping, even if different sensor sources are used. These ideas will be explored in our future research.

Acknowledgement

This research has been funded by the European Regional Development Fund within the project “Clean 4.0 – Accelerating the Adoption of Autonomous Technologies across the European Cleaning Ecosystem”, grant number 1.1.1.1/20/A/186.

References

[1] H. Bavle, J. L. Sanchez-Lopez, C. Cimarelli, A. Tourani, and H. Voos, “From slam to situational awareness: Challenges and survey,” *Sensors*, vol. 23, no.10, May 2023, Art. no. 4849. https://doi.org/10.3390/s23104849

[2] L. P. Nalla Perumal and A. S. Arockia Doss, “Sensor fusion for automotive dead reckoning using GPS and IMU for accurate position and velocity estimation,” *Trends in Mechanical and Biomedical Design: Select Proceedings of ICMechD 2019*, pp. 83–95, 2021. https://doi.org/10.1007/978-981-15-4488-0_8

[3] N. El-Sheimy and Y. Li, “Indoor navigation: State of the art and future trends,” *Satellite Navigation*, vol. 2, no. 1, May 2021, Art. no. 7. https://doi.org/10.1186/s43020-021-00041-3

[4] I. A. Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, “A survey of state-of-the-art on visual slam,” *Expert Systems with Applications*, vol. 205, Nov. 2022, Art. no. 117734. https://doi.org/10.1016/j.eswa.2022.117734

[5] M. Bujanca, X. Shi, M. Spear, P. Zhao, B. Lennox, and M. Luján, “Robust slam systems: Are we there yet?” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, Sep. 2021, pp. 5320–5327. https://doi.org/10.1109/IROS51168.2021.9636814

[6] W. Chen, C. Zhou, G. Shang, X. Wang, Z. Li, C. Xu, and K. Hu, “Slam overview: From single sensor to heterogeneous fusion,” *Remote Sensing*, vol. 14, no. 23, Nov. 2022, Art. no. 6033. https://doi.org/10.3390/rs14236033

[7] C. V. Jones, G. E. Hall, S. J. Baron, B. Hild, S. Zickler, and J. Sinnigen, “Mobile cleaning robot artificial intelligence for situational awareness,” U.S. Patent 10 878 294, Dec. 29, 2020.

[8] M. Munich, A. Kolling, M. Narayana, and P. Fong, “Mapping for autonomous mobile robots,” U.S. Patent 11 249 482, Feb. 15, 2022.

[9] C. A. Velásquez Hernández and F. A. Prieto Ortiz, “A real-time map merging strategy for robust collaborative reconstruction of unknown environments,” *Expert Systems with Applications*, vol. 145, May 2020, Art. no. 113109. https://doi.org/10.1016/j.eswa.2019.113109

[10] Y. Xie, Y. Tang, R. Zhou, Y. Guo, and H. Shi, “Map merging with terrain-adaptive density using mobile 3D laser scanner,” *Robotics and Autonomous Systems*, vol. 134, Dec. 2020, Art. no. 103649. https://doi.org/10.1016/j.robot.2020.103649

[11] Z. Li, B. Xu, D. Wu, K. Zhao, S. Chen, M. Lu, and J. Cong, “A YOLO-GGCNN based grasping framework for mobile robots in unknown environments,” *Expert Systems with Applications*, vol. 225, Sep. 2023, Art. no. 119993. https://doi.org/10.1016/j.eswa.2023.119993

[12] N. Banerjee *et al.*, “Lifelong mapping in the wild: Novel strategies for ensuring map stability and accuracy over time evaluated on thousands of robots,” *Robotics and Autonomous Systems*, vol. 164, Jun. 2023, Art. no. 104403. https://doi.org/10.1016/j.robot.2023.104403

[13] S.-H. Chan, P.-T. Wu, and L.-C. Fu, “Robust 2D indoor localization through laser slam and visual slam fusion,” in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan, Oct. 2018, pp. 1263–1268. https://doi.org/10.1109/SMC.2018.00221

[14] Y. Xu, Y. Ou, and T. Xu, “Slam of robot based on the fusion of vision and lidar,” in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, Shenzhen, China, Oct. 2018, pp. 121–126. https://doi.org/10.1109/CBS.2018.8612212

[15] X. Xu, L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, and M. Luo, “A review of multi-sensor fusion slam systems based on 3D lidar,” *Remote Sensing*, vol. 14, no. 12, Jun. 2022, Art. no. 2835. https://doi.org/10.3390/rs14122835

[16] M. F. Ahmed, K. Masood, V. Fremont, and I. Fantoni, “Active slam: A review on last decade,” *Sensors*, vol. 23, no. 19, Sep. 2023, Art. no. 8097. https://doi.org/10.3390/s23198097

[17] H. Zhou, Z. Yao, and M. Lu, “Lidar/UWB fusion based SLAM with anti-degeneration capability,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 820–830, Dec. 2020. https://doi.org/10.1109/TVT.2020.3045767

[18] X. Dang, Z. Rong, and X. Liang, “Sensor fusion-based approach to eliminating moving objects for SLAM in dynamic environments,” *Sensors*, vol. 21, no. 1, Jan. 2021, Art. no. 230. https://doi.org/10.3390/s21010230

[19] I. Andersone, “Heterogeneous map merging: State of the art,” *Robotics*, vol. 8, no. 3, Aug. 2019, Art. no. 74. https://doi.org/10.3390/robotics8030074

[20] C. Debeunne and D. Vivet, “A review of visual-lidar fusion based simultaneous localization and mapping,” *Sensors*, vol. 20, no. 7, Apr. 2020, Art. no. 2068. https://doi.org/10.3390/s20072068

[21] F. Chanier, P. Checchin, C. Blanc, and L. Trassoudaine, “Map fusion based on a multi-map SLAM framework,” in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Korea (South), Aug. 2008, pp. 533–538. https://doi.org/10.1109/MFI.2008.4648050

[22] Y. Lu, J. Lee, S.-H. Yeh, H.-M. Cheng, B. Chen, and D. Song, “Sharing heterogeneous spatial knowledge: Map fusion between asynchronous monocular vision and lidar or other prior inputs,” in *Robotics Research: The 18th International Symposium ISRR*, Nov. 2020, pp. 727–741. https://doi.org/10.1007/978-3-030-28619-4_51

[23] Y. Megahed, A. Shaker, and W. Y. Yan, "Fusion of airborne lidar point clouds and aerial images for heterogeneous land-use urban mapping," *Remote Sensing*, vol. 13, no. 4, Feb. 2021, Art. no. 814. https://doi.org/10.3390/rs13040814

[24] B. Zhang, J. Liu, and H. Chen, "AMCL based map fusion for multi-robot SLAM with heterogenous sensors," in *2013 IEEE International Conference on Information and Automation (ICIA)*, Yinchuan, China, Aug. 2013, pp. 822–827. https://doi.org/10.1109/ICInfA.2013.6720407

[25] Z.-g. Liu, L. Zhang, G. Li, and Y. He, "Change detection in heterogeneous remote sensing images based on the fusion of pixel transformation," in *2017 20th International Conference on Information Fusion (Fusion)*, Xi'an, China, Jul. 2017, pp. 1–6. https://doi.org/10.23919/ICIF.2017.8009656

Aleksandrs Sisojevs was born in 1980. He obtained the degree of Bachelor of Engineering in 2001 and Engineer qualification in 2003 from Riga Technical University and Master of Computer Science in 2006 from the University of Latvia. In 2011, he received the Degree of Doctor of Engineering in Information Technology, Systems Analysis, Modelling and Design. He received the position of an Associate Professor in 2018 and a Senior Researcher in 2019 of the Faculty of Computer Science, Information Technology and Energy (former Faculty of Computer Science and Information Technology) at Riga Technical University. His research interests include computer graphics, geometrical modelling & processing, image & signal processing, pattern recognition and computer vision.
E-mail: alexiv@inbox.lv
ORCID iD: https://orcid.org/0000-0002-2267-4220

Vilnis Juris Turkovs received BSc in Intelligent Robotic Systems from Riga Technical University, Latvia, 2018. He is currently working as a Developer at MikroTik. He has industrial and research experience in the field of autonomous mobile robots. His research interests include robot controls, SLAM and embedded systems.
E-mail: vilnisturkovs@gmail.com

Mārtiņš Banis was born in 1987. He earned a professional Bachelor degree in Computer Science and a qualification of Programming Engineer from EKA University of Applied Sciences. He has a diverse range of technical and non-technical qualifications, including certifications in Deep Learning and Artificial Intelligence. His work involves managing R&D projects, developing and deploying robotic solutions, and overseeing product creation from design to market launch. In 2023, he was appointed as the Chief Product Officer at Robotic Solutions.
E-mail: martins.banis@gmail.com

Aleksandrs Koršunovs received MSc in Intelligent Robotic Systems from Riga Technical University, Latvia, 2019. He has industrial and research experience in the field of autonomous mobile robots. His research interests include robot navigation and navigation algorithm development.
E-mail: aleksandrs.korsunovs@roboticsolutions.lv

Reinis Cimurs received B.S. degree in Automation and Computer Control from Riga Technical University in 2011 and M.S. degree in Automation and Computer Control from Riga Technical University in 2014. He received his PhD in Intelligent Robot Engineering in 2021 from Hanyang University, Seoul, the Republic of Korea where he also conducted his post-doctoral studies in the area of autonomous exploration with mobile robots. His research interests include path planning, behaviour modelling, robot navigation and deep learning.
E-mail: reinis.cimurs@roboticsolutions.lv
ORCID iD: https://orcid.org/0000-0002-9256-6655