Dynamic map update of non-static facility logistics environment with a multi-robot system

Nayabrasul Shaik[1], Thomas Liebig[1], Christopher Kirsch[2], and Heinrich Müller[1]

[1] TU Dortmund University, Dortmund, Germany
{nayabrasul.shaik, thomas.liebig, heinrich.mueller}@tu-dortmund.de
[2] Fraunhofer Institute for Material Flow and Logistics, Dortmund, Germany
Christopher.Kirsch@iml.fraunhofer.de

Abstract. Autonomous robots need to perceive and represent their environments and act accordingly. Using simultaneous localization and mapping (SLAM) methods, robots can build maps of the environment which are efficient for localization and path planning as long as the environment remains unchanged. However, facility logistics environments are not static because pallets and other obstacles are stored temporarily. This paper proposes a novel solution for updating maps of changing environments (i.e. environments with low-dynamic or semi-static objects) in real-time with multiple robots. Each robot is equipped with a laser range sensor and runs localization to estimate its position. Each robot senses the change in the environment with respect to a current map, initially built with a SLAM method, and constructs a temporary map which will be merged into the current map using localization information and line features of the map. This procedure enables the creation of long-term mapping robot systems for facility logistics.

1 Introduction

For autonomous navigation, robots need a representation of the operating environment. Maps of static environments can be built using simultaneous localization and mapping (SLAM) methods. Maps built with SLAM work well for localization and path panning as long as the environment remains static [9]. But most of the environments are not static due to changes during day-to-day operations. These changes can be due to *high-dynamic objects* or *low-dynamic objects*. Objects whose location change can be observed in the robot's field of view, e.g. humans or other moving vehicles are high-dynamic objects. Objects like pallets and stationary vehicles, which are stationary in the robot's field of view, are low-dynamic objects [26,31]. Low-dynamic objects are also termed *semi-static objects* [25]. Considering changes in the environment due to the low-dynamic objects can improve the localization capabilities of the robot system [25] and improve path planning of large robot teams [17] as well as the coordination of a multi-robot system. If the up-to-date map could be combined with a coordination algorithm (e.g. Multi-Agent-System), fixed routes could be changed and optimized for changes in the environment. A very crucial part is the reduction of "reactive behaviours".

If the robots of the multi-robot system can share the information about dynamic or semi-static objects, the efforts of obstacle avoidance could be reduced. This could save costs and time in logistic environments because the robots can always drive the faster path which is coordinated with all robots and planned according to the latest environment information. Hence coordinated path planning with updated environment informtion could reduce waiting time and guarantees the achievement of transports [22,23].

The mapping of dynamic obstacles is therefore a major step towards life-long robot navigation. The scope of this work is (1) to detect the low-dynamic objects and (2) to update the representation of the environment in long-term operation of multi-robot system.

The following terminology is used throughout the paper:

- *Static Map*: Map built initially by a standard SLAM algorithm. It contains static features of the environment like walls and fixed machinery which never changes. Its line features will be used by the approach for alignment.
- *Temporary Map*: Maps built by each robot upon detecting changes.
- *Current Map*: Map updated so far by merging temporary maps. With each sensor update every robot checks for changes in the environment in comparison to the current map. Initially, the current map is the same as the static map.

The paper is organized as follows. In the next section related work is presented. The succeeding section describes the approach and explains the method for calculating divergence and line-based map merging. Real-world results are presented in Section 5 followed by conclusions and future work.

2 Related Work

In case of building a static map of an unknown environment with a single robot system, many SLAM methods are described in literature. Most of those methods are based on the Extended Kalman Filter (EKF) [12] and the Rao-Blackwellised particle filter [27]. Cooperative Simultaneous Localization and Mapping (C-SLAM) methods are used in case of multi-robot systems [13]. Particle filters are also extended to handle multi-robot SLAM [14].

Handling changes in the environment for life-long navigation of robots is currently a major research topic [4]. Meyer-Delius et al. [25] used temporary maps for localisation in a semi-static environment. Their approach maintains temporary maps in a KD-tree and uses the corresponding map when observations are not consistent with the static map. Temporary maps are created when the fraction of range measurements in the current observation, which is not consistent with the current map (called *outlier ratio*), exceeds a predefined threshold, and when no existing temporary map explains the current observations. Temporary maps are discarded when the average outlier ratio is high. Jensen et al. [15] employed the shape information of objects and visibility criteria to update changes in a semi-static environment. Both [25] and [15] are for the case of a single robot system. In [10], each robot maintains a global map and senses changes in the environment based on divergence of short term and long term likelihoods. Upon detecting a change, a temporary map is built. Temporary maps are merged into the global map using rigid transformation. The calculation of the transformation bases on the Hough spectrum [11]. The resulting map is dispatched to the other robots, and each robot updates its map based on this information. Kleiner et al. [17] used occupancy grid maps with Hidden Markov Models (HMM) to detect changes with a large team of robots in real-time to compute an optimal road map.

Various direct and indirect map merging algorithms [20] find transformations between maps (that are built by individual robots) using relative positions, and common areas. Carpin et al. [11] find the transformation between maps based on Hough transform, X-spectrum and Y-spectrum. Lakaemper et al. [19] used shape similarity to merge maps with polygonal curves.

Many autonomous guided vehicle systems are present for intra-logistics in warehouses for material flow and order fulfilment. Kiva systems use the mobile robot drivepod shown in Figure 1(a) [3] which uses cameras for navigation to read bar codes placed on the floor. The KARIS system, shown in Figure 1(b) [30], uses grid map based Monte Carlo localization for autonomous navigation. Grenzebach's G-Pro vehicles, see Figure 1(c) [1], use induction loops in the floor for navigation. Fraunhofer IML's Cellular Transport System [16] replaces conveyor systems by a swarm of Cellular Transport Vehicles, shown in Figure 1(d), with transport capabilities for material handling.

(a) Kiva System

(b) Karis System

(c) Granzbach G-Pro AGV

(d) Cellular Transport Vehicle

Fig. 1: Robots in warehouse logistics

3 Approach

In the approach proposed in this paper each robot detects changes in the environment and builds a temporary map. The temporary maps are merged into the current map. An initial map of the environment is built with a standard SLAM algorithm which contains only static parts of the environment (i.e walls and fixed installations). This map will be called *static map* in the following.

Each robot is equipped with a laser range sensor and runs the localization to estimate its position. Sensor observations and the estimated position are used to detect changes in the environment with respect to the given *current map* (at first, the initial current map equals the static map). Upon detecting a change in the environment, the robot can start and stop building a *temporary map* which will be merged into the current map. Figure 2 shows an outline of the approach. In this context a robot can either be in *free state* in which it did not detect any change, or a robot can be in *building temporary map* state in which the robot detected a change and is building a temporary map. Afterwards, merging of the temporary map and the current map is done using line features from the static parts of the environment. The updated current map will be used for further detection of changes.

3.1 Assumptions

The approach makes the following assumptions for reliable map merging:

- The localization uncertainty is not very high. Otherwise matching of corresponding lines will be difficult.
- The environment has enough line features. This is mostly common in indoor environments.
- Enough static line features of the environment are present in a temporary map. If a temporary map contains entirely new information, it will not be possible to do line matching.

The three main blocks of the approach, *Detecting/sensing change*, *Building temporary maps*, *Map merging*, are described in the following subsections.

3.2 Detecting/Sensing Change

Detecting/sensing change in the environment is done using weighted recency averaging of the likelihood and utilizes the method of [9]. The main idea is to find the divergence of short-term and long-term measurement likelihood for a given tuple of an estimated pose x_t, a laser scan Z_t and a map m:

$$W_{avg}(t) = p(z_t|x_t, m), \tag{1}$$

$$W_{slow}(t+1) = W_{slow}(t) + \alpha_{slow} * (W_{avg}(t) - w_{slow}(t)), \tag{2}$$

$$W_{fast}(t+1) = W_{fast}(t) + \alpha_{fast} * (W_{avg}(t) - W_{fast}(t)), \tag{3}$$

$$d(t) = max(0, 1 - \frac{W_{fast}(t)}{W_{slow}(t)}). \tag{4}$$

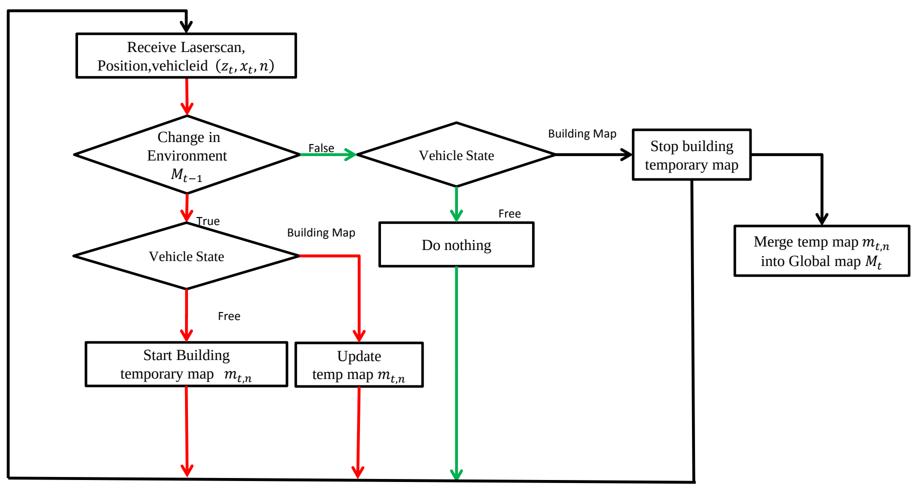


Fig. 2: Outline of the approach

- If $d(t) > 0$ start building a temporary map.
- If $d(t) \leq 0$ stop building the temporary map and merge it with the current map.

$\alpha_{slow}, \alpha_{fast}$ are decay parameters such that $0 \leq \alpha_{slow} << \alpha_{fast}$ and equation (1) represents a laser sensor model based on beam range finder model [29] to find the probability p of the observation Z_t being at the location x_t given the map m. Figure 3 shows the evolution of the divergence $d(t)$, α_{slow}, α_{fast}, α_{avg} for a period of about five minutes navigation through a changing environment. Besides changes in the environment also a rotation of the robot can cause a change in the divergence which may trigger the creation of a temporary map even if there is no change in the environment. This is mitigated by not updating the divergence during rotation of the robot which can be seen in the plot as constant values.

3.3 Building Temporary Maps

Temporary maps were built using the Hector SLAM [2] package available in *Robot Operating System* (ROS) [7]. Hector SLAM builds the occupancy grid based on scan matching by aligning the end points of current laser scan beams with the map learned so far using a Gauss-Newton approach. A multi-resolution map

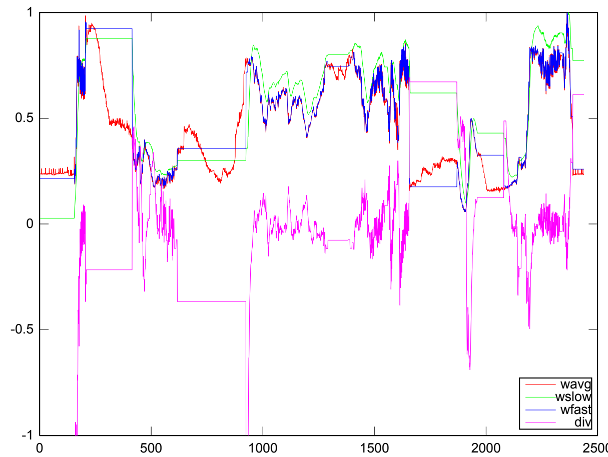


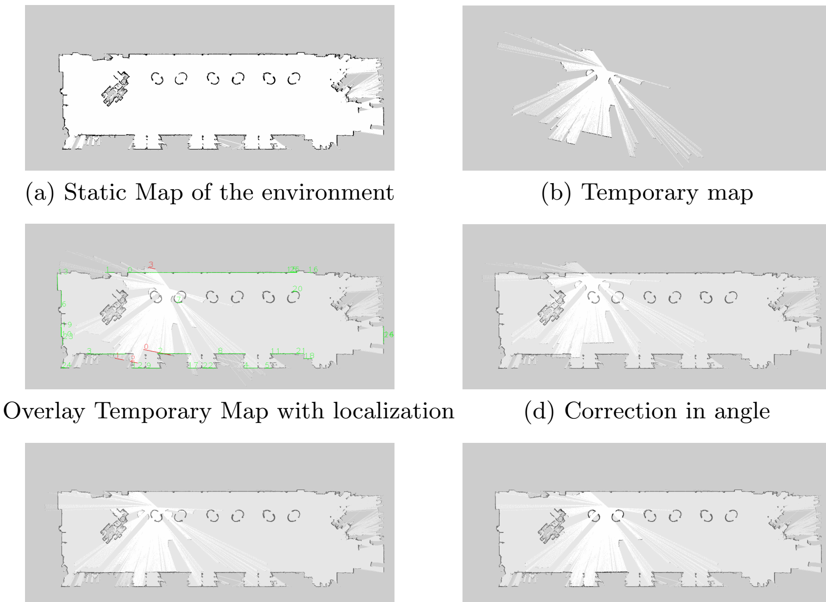
Fig. 3: Evaluation of divergence, $w_{avg}(t)$, $w_{fast}(t)$, $w_{slow}(t)$

representation is used to address the problem of local minima [18]. Examples of temporary maps can be seen in Figure 4(b) and Figure 5(c).

3.4 Fusion / Merging Temporary Map

The goal of the merging process is to align and merge the temporary maps with the current map. In this step an obstacle which is temporarily mapped will be added to the map or subtracted by its removal. To align the maps, localization information and line features are used. Initially, the temporary map is transformed into a static map coordinate system, utilizing the first robot location from which the construction of the temporary map started. Depending on the localization method, the accuracy of the estimated position varies. The effect of an uncertainty in the localization position can be seen in Figure 4(c). Ideally, if the estimated position, obtained by localization, is accurate, the temporary map should align with the current map perfectly. To adjust misalignments of these maps, occurring due to the uncertainty in estimated position, line features from the static environment are used. The correcting transformation $T\{dx, dy, d\theta\}$ is calculated sequentially, i.e. initially a correction in angle $d\theta$ is found. Afterwards, the vertical displacement dy and the horizontal displacement dx are calculated [28].

Line segments from the static map and the temporary map are extracted using the Hough transform [6]. Due to the width of the edges and noise in the sensor readings the same edge can provide various lines. This set of line segments is preprocessed so that each edge is represented by a single line segment. Extracted lines can be seen in Figure 4(c), green and red line segments correspond to static and temporary maps, respectively. Afterwards, matching line segments in pairs are determined. Each line segment is represented by Hesse Normal Form which contains the normal distance from the origin r and the angle between the normal and the horizontal axis θ. A pair of lines is defined by the normal distance



(a) Static Map of the environment

(b) Temporary map

(c) Overlay Temporary Map with localization

(d) Correction in angle

(e) Vertical Correction

(d) Horizontal Correction

Fig. 4: Exemplification of the proposed map merging steps using line segments.

$(r_m - r_n)$ and angular difference $(\theta_m - \theta_n)$. For a given pair of line segments in the temporary map $\{l_{tm}, l_{tn}\}$, all the matching pairs of line segments in the static map are found. All the lines in the static map are candidates for matching line segments. For both segments in the temporary pair, a normal is drawn from the mid point. In turn, the nearest line segment from the candidate line segments of the matched static pair is found. The nearest lines are matching lines if the nearest line segment pair is matched with temporary pair.

Correction in angle $d\theta$ is the mean value of the angle differences between the matched line segments weighted by the length of the temporary line. The vertical displacement dy is the normal distance between the matching line segments after rotating temporary line segments with correction angle $d\theta$.

Before calculating the horizontal displacement dx, the temporary map is transformed with the previously calculated correction in angle $d\theta$, see Figure 4(d), and the vertical displacement dy (Figure 4(e)). The horizontal displacement is calculated by finding maximum matching horizontal displacement i.e. the displacement which corresponds to the maximum matching of grid cells in the static map and transformed temporary map.

4 Implementation

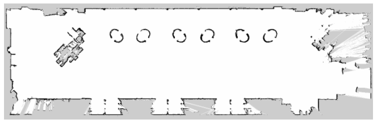
The approach has been implemented on Cellular Transport System (CTS) vehicles (Figure 1(d)) [16]. Each vehicle is equipped with a *Sick* safety laser sensor and runs landmark-based Monte Carlo localization to estimate its position. The server runs Ubuntu 14.04 with ROS Indigo. Each vehicle sends its estimated position (x_t) and the corresponding laser reading (Z_t) to the server through ZeroMQ [8]. On the server the divergence is calculated and temporary maps are built. Merging of the temporary maps with the static map is done on server.

5 Experiments and Results

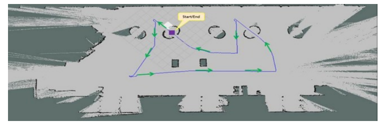
The approach has been tested in the LivingLab for Cellular Transport Systems at the Fraunhofer Institute for Material Flow and Logistics [5]. The dimension of the testing area is about $60\,\mathrm{m} \times 18\,\mathrm{m}$. Initially a complete map of the environment with static parts is built using Hector SLAM which can be seen in Figure 5(a). This static map contains walls, picking stations (seen as ellipses) and other fixed installations (one on the right and another one far to the left). Later, two pallets were placed in the environment and few other stationary robots were also present along the horizontal wall at the bottom during the experiment. A test robot was made to navigate along the path shown in Figure 5(b) at speed of 0.5 m/s which takes about 5 minutes. In this experiment, the robot detected changes at four locations and corresponding maps were built.

One of the temporary maps is shown in Figure 5(c). The temporary maps are transformed and merged into the current map. Figure 5(d) shows correction of temporary map overlaid onto current map. The two pallets were successfully detected and updated in the map after merging the temporary maps, cf. Figure 5(e). In addition, added stationary robots can be seen along the horizontal wall at the bottom.

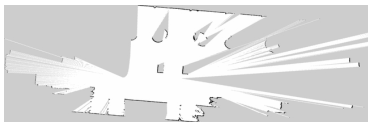
Next, the performance of map matching has been investigated in detail. Two cases are distinguished: (1) addition of obstacles and (2) removal. The robot speed is equal in both cases. For each case three temporary maps are presented and the edge-wise matching errors are highlighted. The resulting maps for case one (addition of obstacle) are shown in Figure 7.



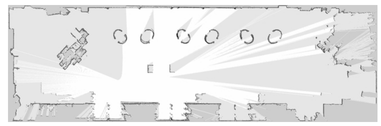
(a) Static Map



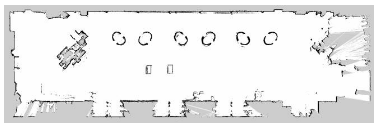
(b) Path of robot during experiment



(c) Exemplified Temporary Map

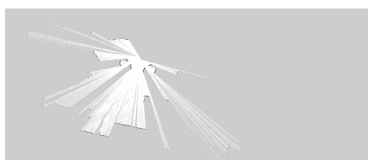


(d) Overlaying Temporary Map

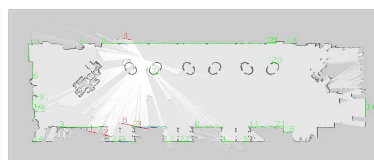


(e) Final map after Merging

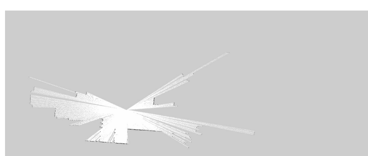
Fig. 5: Results of map updating



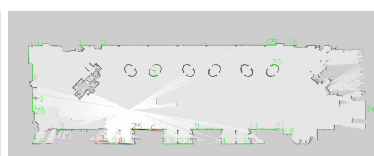
(a) Temporary Map1



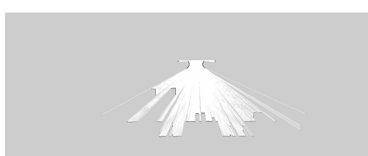
(b) Matching of Temporary Map1



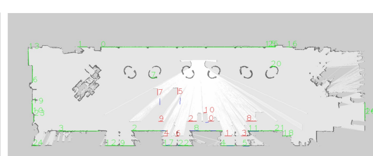
(c) Temporary Map2



(d) Matching of Temporary Map2



(c) Temporary Map3



(d) Matching of Temporary Map3

Fig. 6: Evaluation of line matching for temporary maps in case of obstacle addition. Lines from both the static and the temporary map are numbered and shown in green, and red, respectively. Additionally lines from the temporary map after using corrected transformation are shown in blue. Best viewed in color.

Table 1: Performance of edge-wise matching in case of obstacle addition.

Temporary Map	temporary line ID	global line ID	distance difference in px	angle difference in degrees
Map1	0	2	0.707107	0
Map1	1	12	0.707107	0
Map1	2	9	0.707107	0
Map1	3	2	0.689639	-1.20035
Map1	4	0	1.6094	2.24936
Map2	0	2	2.54951	0
Map2	2	2	0.695382	-0.925865
Map2	3	12	0.707107	0
Map2	4	9	0.707107	1.20035
Map2	5	2	1.58114	0
Map3	0	8	2.54951	0
Map3	1	4	0.695382	0
Map3	2	8	0.707107	0
Map3	3	5	0.707107	0
Map3	6	17	1.58114	3.08052
Map3	8	11	0.707107	0.605741
Map3	9	2	1.58114	0

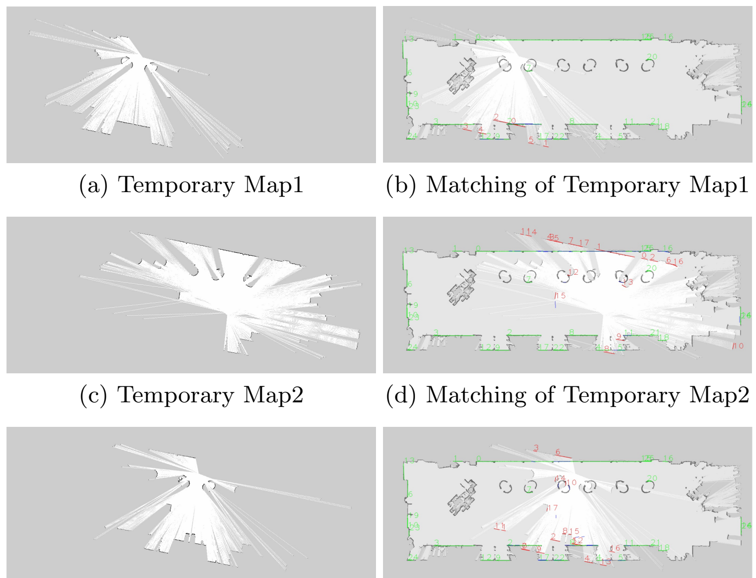
Table 2: Performance of edge-wise matching in case of obstacle removal.

Temporary Map	temporary line ID	global line ID	distance difference in px	angle difference in degrees
Map1	0	2	0.707107	0
Map1	1	22	1.58114	0
Map1	2	2	1.58378	−0.605741
Map1	3	12	1.58114	0
Map1	4	9	0.707107	0
Map1	5	22	1.58114	0
Map2	0	0	0.707107	0
Map2	1	0	0.707107	0
Map2	2	0	0.707107	0
Map2	3	0	0.707107	0
Map2	4	0	1.57597	1.21538
Map2	5	0	0.707107	−1.83647
Map2	7	0	0.707107	2.24936
Map2	8	4	0.707107	0
Map2	9	11	1.4301	−1.25872
Map2	11	0	1.58114	0
Map2	14	0	1.58114	0
Map2	17	0	0.707107	0
Map3	1	2	0.707107	0
Map3	2	8	0.707107	0
Map3	3	0	1.63474	4.13334
Map3	4	4	0.707107	0
Map3	5	8	0.707107	0
Map3	6	0	0.695467	−1.03391
Map3	7	2	52.5024	0
Map3	9	22	1.59199	−1.93029
Map3	11	2	0.707107	2.81821
Map3	12	8	1.58114	0
Map3	13	5	1.59479	−2.3415

Results from the two datasets above show that the approach has produced consistent occupancy grid maps which represent the map changes. Also from the evaluation calculations, the distance difference between matched lines in almost all the cases was less than 1 or 2 pixels. This error can occur due to the noise from the laser measurements or due to the approximation of lines from Hough transformation. Also the angle difference between matched lines was zero in most cases and maximum difference obtained was less than 5 degrees.

6 Conclusion and Future Work

We proposed a method for updating the environment map for long term operation of a multi-robot system. Therefore, we utilized line features to merge grid maps. Our experiments demonstrate practicability of the approach. The method can



(a) Temporary Map1 (b) Matching of Temporary Map1

(c) Temporary Map2 (d) Matching of Temporary Map2

(c) Temporary Map3 (d) Matching of Temporary Map3

Fig. 7: Evaluation of line matching for temporary maps in case of obstacle removal. Lines from both static and temporary map are numbered and shown in green, and red, respectively. Additionally lines from temporary map after using corrected transformation are shown in blue. Best viewed in color.

be applied easily in a multi-robot environment. Next steps will be, besides improvement of the approach, to perform qualitative evaluations in more complex and realistic scenarios for longer durations. The mapping of dynamic obstacles is a major step towards life-long navigation. In future, related path planning, collision avoidance and congestion avoidance problems will also be studied. The fusion with stationary sensors (e.g. Bluetooth [21,24]) directs to autonomous vehicle problems, where Car/to/Infrastructure communication is an active reserach topic.

Acknowledgements. The authors were partially funded by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876, project B2 (the study was also performed in collaboration with project B4) and the European Union Horizon 2020 Programme (Horizon2020/2014-2020), under grant agreement number 688380 “VaVeL: Variety, Veracity, VaLue: Handling the Multiplicity of Urban Sensors”.

References

1. Grenzebach maschinenbau gmbh, hamlar, germany. `http://www.grenzebach.com/index.php/eng/technology/logistic_solutions/agv_solutions/g_pro`, [Online; accessed 03-May-2015]

2. Hector slam. http://wiki.ros.org/hector_slam, [Online; accessed 07-April-2015]
3. Kivasystems. http://www.kivasystems.com/solutions/system-overview/, [Online; accessed 03-May-2015]
4. Lifenav – reliable lifelong navigation for mobile robots. http://lifenav.informatik.uni-freiburg.de/index.html, [Online; accessed 03-May-2015]
5. Livinglab zellulare transportsysteme. http://www.iml.fraunhofer.de/en/researchhallslaboratories/zft-halle.html, [Online; accessed 10-April-2015]
6. Opencv houghlinesp. http://docs.opencv.org/modules/imgproc/doc/feature_detection.html?highlight=houghlinesp#houghlinesp, [Online; accessed 12-April-2015]
7. Robot operating system. http://www.ros.org/about-ros/, [Online; accessed 07-April-2015]
8. zeromq. http://www.zeromq.org, [Online; accessed 07-April-2015]
9. Abrate, F., Bona, B., Indri, M., Rosa, S., Tibaldi, F.: Map updating in dynamic environments. In: ISR/ROBOTIK 2010,. pp. 1–8 (2010)
10. Abrate, F., Bona, B., Indri, M., Rosa, S., Tibaldi, F.: Multi-robot map updating in dynamic environments. In: Martinoli, A., Mondada, F., Correll, N., Mermoud, G., Egerstedt, M., Hsieh, M.A., Parker, L.E., Støy, K. (eds.) Distributed Autonomous Robotic Systems - The 10th International Symposium, DARS 2010, Lausanne, Switzerland, November 1-3, 2010. Springer Tracts in Advanced Robotics, vol. 83, pp. 147–160. Springer (2010), https://doi.org/10.1007/978-3-642-32723-0
11. Carpin, S.: Fast and accurate map merging for multi-robot systems. Auton. Robots 25(3), 305–316 (2008)
12. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part i. Robotics Automation Magazine, IEEE 13(2), 99–110 (June 2006)
13. Fenwick, J.W., Newman, P.M., Leonard, J.J.: Cooperative concurrent mapping and localization. pp. 1810–1817 (2002)
14. Howard, A.: Multi-robot simultaneous localization and mapping using particle filters. Int. J. Rob. Res. 25(12), 1243–1256 (Dec 2006)
15. Jensen, B., Ramel, G., Siegwart, R.: Detecting semi-static objects with a laser scanner. In: Dillmann, R., Wörn, H., Gockel, T. (eds.) Autonome Mobile Systeme 2003, pp. 21–31. Informatik aktuell, Springer Berlin Heidelberg (2003)
16. Kamagaew, A., Stenzel, J., Nettstrater, A., ten Hompel, M.: Concept of cellular transport systems in facility logistics. In: Automation, Robotics and Applications (ICARA), 2011 5th International Conference on. pp. 40–45 (Dec 2011)
17. Kleiner, A., Sun, D., Meyer-Delius, D.: Armo: Adaptive road map optimization for large robot teams. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. pp. 3276–3282 (Sept 2011)
18. Kohlbrecher, S., Meyer, J., von Stryk, O., Klingauf, U.: A flexible and scalable slam system with full 3d motion estimation. In: Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). IEEE (November 2011)
19. Lakaemper, R., Latecki, L., Wolter, D.: Incremental multi-robot mapping. In: Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on. pp. 3846–3851 (Aug 2005)
20. Lee, H.C., Lee, S.H., Lee, T.S., Kim, D.J., Lee, B.H.: A survey of map merging techniques for cooperative-slam. In: Ubiquitous Robots and Ambient Intelligence (URAI), 2012 9th International Conference on. pp. 285–287 (Nov 2012)
21. Liebig, T., Kemloh Wagoum, A.U.: Modelling microscopic pedestrian mobility using bluetooth. In: ICAART. pp. 270–275. SciTePress (2012)

22. Liebig, T., Piatkowski, N., Bockermann, C., Morik, K.: Dynamic route planning with real-time traffic predictions. Information Systems 64, 258–265 (2017), http://www.sciencedirect.com/science/article/pii/S0306437916000181
23. Liebig, T., Sotzny, M.: On avoiding traffic jams with dynamic self-organizing trip planning. In: Clementini, E., Donnelly, M., Yuan, M., Kray, C., Fogliaroni, P., Ballatore, A. (eds.) Spatial Information Theory - 13th International Conference, COSIT 2017, L'Aquila, Italy, September 4-8, 2017, Proceedings, p. (accepted). L'Aquila, Italy (2017)
24. Liebig, T., Xu, Z., May, M.: Incorporating mobility patterns in pedestrian quantity estimation and sensor placement. In: Citizen in Sensor Networks, pp. 67–80. Springer Berlin Heidelberg (2013)
25. Meyer-Delius, D., Hess, J.M., Grisetti, G., Burgard, W.: Temporary maps for robust localization in semi-static environments. In: IROS. pp. 5750–5755. IEEE (2010)
26. Mitsou, N., Tzafestas, C.: Temporal occupancy grid for mobile robot dynamic environment mapping. In: Control Automation, 2007. MED '07. Mediterranean Conference on. pp. 1–8 (June 2007)
27. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: Fastslam: A factored solution to the simultaneous localization and mapping problem. In: In Proceedings of the AAAI National Conference on Artificial Intelligence. pp. 593–598. AAAI (2002)
28. Nguyen, V., Harati, A., Tomatis, N., Martinelli, A., Siegwart, R.: Orthogonal SLAM: a Step toward Lightweight Indoor Autonomous Navigation. In: None (2006)
29. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press (2005)
30. Trenkle, A., Seibold, Z., Stoll, T.: Safety requirements and safety functions for decentralized controlled autonomous systems. In: Information, Communication and Automation Technologies (ICAT), 2013 XXIV International Symposium on. pp. 1–6 (Oct 2013)
31. Walcott, A.N.: Long-term Robot Mapping in Dynamic Environments. Ph.D. thesis, Cambridge, MA, USA (2011), aAI0823852