# Localization of map changes by exploiting SLAM residuals

Zoltan Rozsa[1,2][0000−0002−3699−6669], Marcell Golarits[1][0000−0001−9652−4148], and Tamas Sziranyi[1,2][0000−0003−2989−0214]

[1] Machine Perception Research Laboratory of Institute for Computer Science and Control (SZTAKI), H-1111 Budapest, Kende u. 13-17., Hungary
sziranyi.tamas@sztaki.mta.hu
[2] Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics (BME KJK), H-1111 Budapest, Műegyetem rkp. 3., Hungary
zoltan.rozsa@logisztika.bme.hu

**Abstract.** Simultaneous Localization and Mapping is widespread in both robotics and autonomous driving. This paper proposes a novel method to identify changes in maps constructed by SLAM algorithms without feature-to-feature comparison. We use ICP-like algorithms to match frames and pose graph optimization to solve the SLAM problem. Finally, we analyze the residuals to localize possible alterations of the map. The concept was tested with 2D LIDAR SLAM problems in simulated and real-life cases.

**Keywords:** SLAM · change detection · autonomous driving.

## 1 Introduction

Simultaneous Localization and Mapping (SLAM) is the basis of the navigation of today's mobile robots and possibly of the future's autonomous vehicles. Mobile robots traverse a path more than once, this offers the possibility of loop closing. Especially true this statement for automated guided vehicles (AGVs) and the connected vehicles of the future. One autonomous car may travel on a road just once, but others will navigate on this road continuously and can utilize the information acquired earlier. From time-to-time, changing of the environment is inevitable. However, one of the biggest problems of the present SLAM algorithms is the robustness against moving objects (short term) and environment change (long term). Using robust features (and RANSAC like algorithms [19]) to estimate the relative transformation can partly eliminate this problem. However, the detected features can be on moving objects as well and this would cause incorrect relative motion estimation. In general, motion estimation is enhanced by increasing the number of matched features, but it will certainly result in decreased performance if there is some change in the scenes where the match is done. Trying to find these changes to eliminate these problems, update the map, or construct the correct 4D reconstruction [20] can be an exhausting search. We propose a solution to automatically detect these changes.

### 1.1 Contributions

The paper contributes to the following:

- A new methodology is proposed to localize possible changes of SLAM maps instead of brute force search.
- The method does not require any additional step, just the basic SLAM need to be executed and error change need to be analyzed.
- Numerous 2D graph SLAM tests have been tested and evaluated to generate the proof of concept.

### 1.2 Outline of the Paper

The paper is organized as follows: Section 2 surveys the literature about the related topics. Section 3 briefly explains the theoretical backgrounds and Section 4 describes the proposed method and the concept of change localization in detail. Sections 5 shows our test results. Finally, Section 6 draws some conclusions.

## 2 Related Works

Today, real-time scan matching and SLAM algorithms [7] available in industrial and market products based on only 2D laser scanner data. However, SLAM can be realized with many sensor types part of the autonomous driving kit. One of the most effective algorithms is the ORB-SLAM which is applicable for mono, stereo, and RGB-D cameras as well [12], but there are particular SLAM algorithms for other 2.5D depth sensors like multi-layer LIDARs [3]. SLAM algorithms can be categorized in many ways besides the sensors for which is applicable. We can distinguish feature-based like the ORB-SLAM semi-direct [5] and direct SLAMs [13], minimizing reprojection or photometric error. There can be 2D and 3D methods based on the motion assumption. In the case of 2D the optimization problem we differentiate filtering-based (e.g. extended Kalman filter - EKF) SLAMs and graph SLAM. In this paper, we deal with direct matching (in case of relative motion and loop closure estimation as well) and graph optimization based SLAMs, like [10]. The motion can be arbitrary, but it will be assumed to be two-dimensional because of ground vehicles and simplicity.

### 2.1 2D Pose Graph SLAM

This sub-problem alone and the related optimization is quite complex and has been widely researched. There are different approaches to simplify the solution of this nonlinear least squares optimization problem. One possibility can be the linear approximation [4], others try to separate the problem to linear and nonlinear parts [8] (if the orientation is known, position estimation is linear). Besides, dimension reduction is a research direction too. H. Wang et al. proved a few theories for graph optimization problems in the case of spherical covariance matrices [23]. They proved that six-dimensional least square optimization problem of the trivial pose-graph (3 nodes, 3 edges) can be reduced to the optimization of one variable and for another trivial problem (two anchor nodes) they provided solutions in closed form. Later, they showed that one step point feature SLAM can be also reduced to a one variable optimization [22]. Also, the general point feature SLAM of $3m + 2n$ variables to an $m$ variables optimization problem ($m$ is the number of poses and $n$ is the number of features). It can be explained by its separable structure and with the fact that features can be considered as poses in the graph. These are the basis of our research.

### 2.2 SLAM and Change Detection

Change detection is important for many reasons as surveillance, statical monitoring of buildings, traffic forecasting or path planning of vehicles. It can be realized with different sensors like mobile-laser scanners [24] or cameras [15] . It can be done with different data structures like point-cloud generated with depth sensors [21] or mono cameras [15] by Structure from Motion (SfM) or on 2D image pairs with conventional methods [16] or deep networks [1]. The common point and disadvantage of these processes are that we have to do the comparison from frame to frame with a previous image or a submap.

Object graphs can be applied for visual place recognition [14] or solution of the whole SLAM problem [11]. Some semantic SLAM algorithms are capable of detecting objects which are inconsistent with previous measurements [17]. However, they require high-level interpretation and limited to objects constructing the pose graph. Other methods just ignore the inconsistent objects. Making robust SLAM algorithms in dynamic environments and analyzing their behavior is an actual topic [25], a survey about them can be found in [18], but these researches deal with moving objects instead of long term changes.

## 3 The Problem Formulation

In the following, a brief introduction will be given to the problem of 2D graph SLAMs. A detailed explanation can be found in [6]. We assume to have a heterogeneous graph with just robot poses or known data association (feature identification is correctly done by the front-end). The quantity and effect of false loop closures (perceptual aliasing [9]) is negligible, because of the vehicle, robot (e.g. patrolling ones) making rounds, we have a good position estimation.

$X = (p_1, \cdots, p_n)^T$ is a vector of position parameters, where $p_i = (x_i, y_i, \theta_i)$ describes the pose of node $i$. Let $z_{i,j}$ and $\Omega_{i,j}$ be the mean and the information matrix of a measurement between the node $i$ and the node $j$. In case of a given conguration of the nodes $p_i$ and $p_j$, $\tilde{z}_{i,j}(p_i, p_j)$ is the prediction of a measurement representing the relative transformation between two nodes. The function of computing the difference between the expected observation and the real observation is:

$$e(p_i, p_j, z_{i,j}) = z_{i,j} - \tilde{z}_{i,j}(p_i, p_j) \tag{1}$$

The error has assumed to have normal distribution with zero mean. The goal of a maximum likelihood approach is to find the conguration of the nodes $p^*$ that minimizes the negative log-likelihood $F(p)$ of all observations:

$$F(p) = \sum_{i,j} e_{i,j}^T \Omega_{i,j} e_{i,j} \tag{2}$$

thus, we aim to solve the following equation:

$$p^* = argmin_p F(p) \tag{3}$$

# 4 Proposed Method

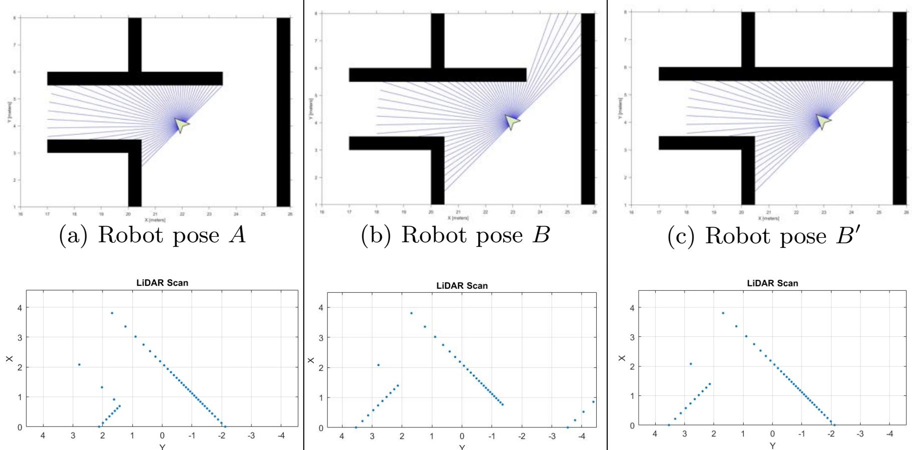
In the following, we assume in our change localization that the relative transformation estimation between two nodes is done by some frame matching algorithm [7] using all the points of the frames. This means relatively small change affects on the estimated transformation too. Usually, these algorithms produce some kind of score of the match, but these strongly depend on the frames we match (the scene and position influences). They cannot be directly compared, used for direct change localization (an illustrative example is shown in Fig. 1 ). However, the environment changes will produce an extra error in the pose estimation, which will appear in the residual error of the pose graph optimization.

## 4.1 Assumptions

We will investigate in the following the case when a new loop closure edge is detected and added to the graph. First, we will make some theoretical assumptions and practical simplifications:

- Loop closures mainly have local effects. This assumption is necessary to identify the neighborhood of the loop closing edge as the location of the change. In the investigated cases: we circle on a small graph (eg. AGVs on material handling system); or traveling and loop closing on a 'detachable' sub-graph of a large graph (road network).
- The uncertainty of positions will not increase with the new loop closure edge. Let $F_n(p)$ and $F_{n+1}(p)$ the minimized log-likelihood after the $n^{th}$ and $(n+1)^{th}$ graph optimization and so loop closure ($n$ is much higher than number of rounds). $F$-s are assumed to be close to the optimal solution in the current configuration and also to the global optimum in case of high $n$ and round numbers. Then $\det(\Omega^n) > \det(\Omega^{n+1})$. It is true that the uncertainty of each new position estimates is increasing, but it will be decreasing with every loop closure edges. This holds either in case of uncertainty derived from the sensor model (as [2] proved) or using identity information matrix as frequently applied in SLAM optimization back-ends.
- Approximately identical variance $\sigma_e$ for all $e_{i,j}$. Because of the locally investigated environment, we assume uniformly distributed measurements and consequently loop closures.

| Position $A$ | Position $B$ | Position $B'$ (Postion $B$ in changed map) |
|---|---|---|




(a) Robot pose $A$ (b) Robot pose $B$ (c) Robot pose $B'$

(d) LIDAR scan in $A$ (e) LIDAR scan in $B$ (f) LIDAR scan in $B'$

| | $A \to B$: | $A \to B'$: |
|---|---|---|
| | match score $= 23.7995$ | match score $= 20.9996$ |
| | $\Delta x_e = 0.0$ | $\Delta x_e = -0.75$ |
| | $\Delta y_e = -0.05$ | $\Delta y_e = -0.7$ |
| | $\Delta\theta_e = 0.0063$ | $\Delta\theta_e = 0.0063$ |

Fig. 1: Example about changing environment can cause wrong relative motion estimation and the uncertainty of match score (bad relative motion estimation with a higher score). Ground truth relative motion: $\Delta x_{gt} = -\frac{\sqrt{2}}{2}, \Delta y_{gt} = -\frac{\sqrt{2}}{2}, \Delta\theta_{gt} = 0.0$

– Residual error is supposed to have a Gaussian distribution. Let $\Omega_{i,j}$ be the identity matrix. After $n$ (high enough) loop closure $F(p) = \sum e_{i,j}^T e_{i,j}$, and the sum of error squares can be approximated as a normal distribution (instead of $\chi^2$) with the expected value:

$$\mu_{\Sigma e^2} = n\sigma_e^2 \tag{4}$$

and variance square:

$$\sigma_{\Sigma e^2}^2 = n2\sigma_e^4 \tag{5}$$

where $\sigma_e$ is the variance of $e_{i,j}$ and the index $\Sigma e^2$ refer to the sum of error squares. The equations above can be deduced from the central limit theorem. This simple form is true for independent and identically distributed variables as we assumed earlier. If that is not the case, $\sigma_{e_{i,j}}$-s significantly differ, $\sigma_e^2$ and $\sigma_e^4$ still can be substituted with one number, the mean of $\sigma_{e_{i,j}}^2$ and $\sigma_{e_{i,j}}^4$. Despite the fact the problem is not linear and $p^*$ node configuration is continuously changing, we found the distribution above to be a good approximation for $F(p^*)$ as the proportion $\frac{\text{number of loop closure edges}}{\text{number of not loop closure graph edges}} \approx 1$ (or higher), as it can be, and was in the cases we investigated.

Using the assumption above, the expected value of the squared error of the next loop closure is approximated as $\mu_{e^2} = \sigma_e^2$ and its variance $\sigma_{e^2} = \sqrt{2}\sigma_e^2$. The threshold for the new error term to decide whether it fits the current approximation of the distribution or not (it indicates a change in the map with $3\sigma$ rule certainty):

$$th = \sigma_e^2 + 3\sqrt{2}\sigma_e^2 \tag{6}$$

It is one of the main ideas of the proposed change localization. It is important to determine a number $n$ where we can start our inspection (previously we are far from the optimum, the residuals are unstable and most importantly the distribution assumption does not hold). It depends on the scale and number of the laps (it has to be minimum 2), the unknown variance of the errors (determined by the environment). We monitor the change of the specific function $f(p^*) = \frac{F(p^*)}{n}$ in order to decide this $n$. When the value of this function numeric derivative approaches 0 (becoming smaller than $3 \cdot 10^{-6}$ in our tests) we consider $n$ large enough. Then, we assume that the difference of residual terms will not frequently change and the distribution of the squares can be approximated as a Gaussian one. The reason for that $\frac{d}{dn}\frac{n\sigma_e^2}{n} = \frac{d}{dn}\sigma_e^2 = 0$, the variance of the errors assumed to be approximately identical, so it does not depend on the number of loop closures.

### 4.2 Processing Steps

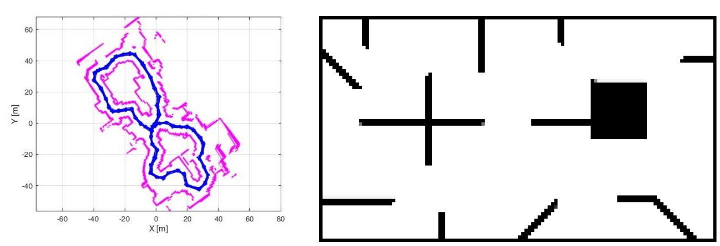
The steps of the proposed method:

1. Compute the earlier defined specific function $f(n)_{|p^*} = \frac{F(n)_{|p^*}}{n}$ and its derivative $\frac{d}{dn}f(n)_{|p^*}$ to examine limit.
2. Define the threshold for change detection at $n$ (number of loop closures) $\frac{d}{dn}[G * f(n)_{|p^*}]$ approaches 0. It will be a higher $n$ value than $\frac{d}{dn}f(n)_{|p^*}$ would result, so we will be closer to the optimum solution of the sub-graph.
3. We have to examine peaks in $\frac{d}{dn}[G * F(n)_{|p^*}]$. Gaussian smoothing is proposed for avoiding single extreme values indicating temporarily local minimums in the graph optimization. Maximums indicating changes should be present in more than one loop closures (depending on the distance between them), because a change in the map can be visible from more than one viewpoint. Also, near extreme values should be checked, because periodic maximums after the change with decreasing values are reasonable (explained later). However, a new maximum with a higher value can mean a new change.
4. In case of an extreme value (possible change candidate) the original value $\frac{d}{dn}F(n)_{|p^*}$ must be compared to the threshold value defined as $\sigma_e^2(1 + 3\sqrt{2})$ (Eq. 6) to decide whether it is salient value (possible change) or not.

### 4.3 Illustration of the Process

In the following, we would like to introduce the change localization process through a test representing typical data. The parameters of the test case are the following. Map: no. 2 (illustrated in Fig. 2), noise in robot position: 0.5 m, number of changes in the map: 5, rounds without change - with change: 10 - 5. Detailed explanation about the parameters and tests can be found in Section 5.

Fig. 3 shows that in this test, in the case of a unchanged environment the residual error approximately linearly increasing with the number of loop closure edges. This fact supports our earlier assumption that the expected value of the matching error in a local environment is approximately the same in the case of approximately equidistant measurements from different positions. The first change can be detected, when there is a loop closing edge that includes a node where the change is perceivable. This will result in an error that does not fit into its previous distribution. This higher error introduced by the map change will be appearing as a step in the error term. However, when the map change will not be visible by the sensor, the approximately linear residual increase will set again, because the consequent frame can be well-matched in the changed environment too. Only relative motion estimation between the old and the new map (loop closure) will result in outstanding error terms. In every round, when the new (displaced or disappeared) objects of the map will be perceivable, a new high error term is added to the current residual values. However, this should decrease every time as we make the rounds. Now in the new state of the map, the edges representing this new state will dominate (also because over time loop closures will be linked to the changed map). Finally, the varying of the residual error should set back to the linear increasing state, but all the error term resulted by the change will not be eliminated. This offset represents the contradictory edges in the graph. The edges and nodes (now outliers) can be filtered out and the map should be updated.




(a) Reconstructed map and path by the SLAM algorithm

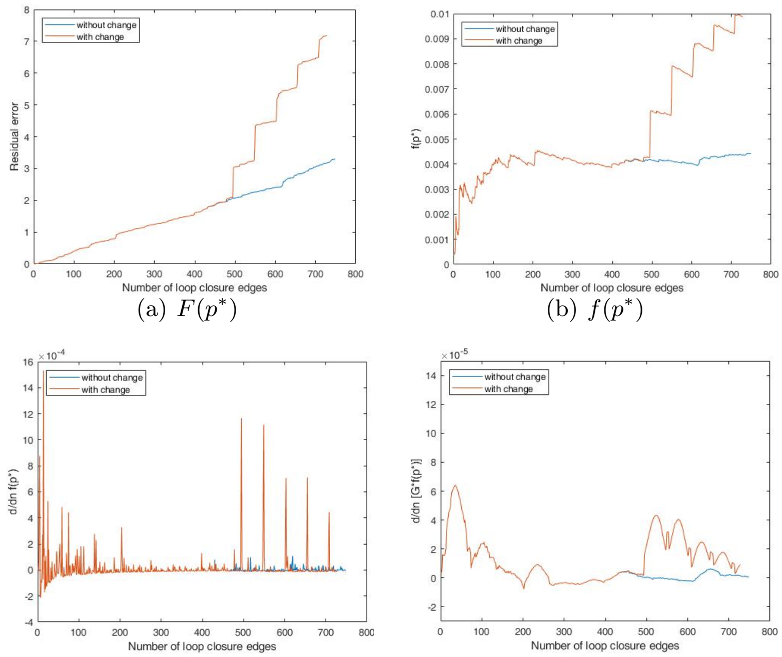
(b) Original map

Fig. 2: Example map no. 2

We illustrated the specific function $f(p^*)$ in a basic case Fig. 4 where 0 m noise added and 0 object changed in the environment (e.g. a tram circles on a fixed path) for comparison to the examined case with changes Fig. 3. In this case, a decreasing tendency can be observed in the measured range.

## 5 Test Results

Our concept and method for change localization have been evaluated in many test cases in a virtual environment and in real-life as well. In this section, the circumstances of the tests and the achieved results are presented.




(a) $F(p^*)$ (b) $f(p^*)$

(c) $\frac{d}{dn} f(p^*)$ (d) $\frac{d}{dn}[G * f(p^*)]$

Fig. 3: The characteristic indicator functions of the test case in unchanged (blue) and changed (orange) environment

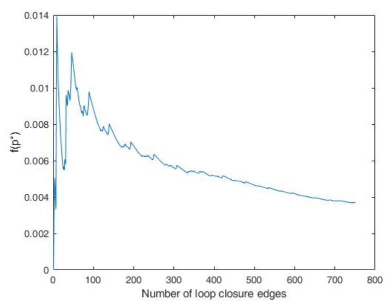



Fig. 4: $f(p^*)$ in the basic case (0 m noise and 0 object changed in the env.)

### 5.1 Quantitative Evaluation

In our synthetic tests, we used MATLAB environment and its robotics toolbox for the generation of test cases, simulate the robot environment perception and for the LIDAR SLAM [7]. First, we created some maps manually called *baseMaps*, then we placed in these *baseMaps* objects with random shape and random position. We used two maps with these random objects in each simulation to test our theorem. One of the maps was used as long term unchanged environment. To approximate real-world experiments, even more, we ran our tests with noise added to the robot path. We used three different *baseMaps*; three different amount of noise in robot position: $\rho = 0m, 0.5m, 1m$ (robot positions in its paths are randomly distributed within $\rho$ radius of its original (first round) path); number of changes in the map: between 0 and 100, rounds without change - with change: both varying between 2 and 10; and we also varied the loop closure threshold of the algorithm from very low to very high. Loop closure threshold is proposed to set a medium value (default Matlab value is appropriate). If it is too low, obviously, too much error will be gathered, but if it is too high loop closure edges of changes will be dropped. (Decreasing loop closure number per round can indicate map change too, but it will not tell its location without further ado.) High errors in robot position, or too much change resulted in incorrect maps (or dropped loop closures for high matching thresholds). These will be left out of the following evaluation because in our earlier assumption we know, the robot circles, so too much deviation from the original path can be filtered. Also, too much change (more than 100 % of original object area appeared - about 35-40 objects) cases indicated by a significant drop of loop closure edges per round are left out. In the final evaluation there were about 20.000 loop closures in 330 rounds.

As we have mentioned previously we evaluated the change of the residual error of the pose graph and its normalized with the number of loop closures. Table 1 shows the detection results based on different thresholds, $xc$, and $xuc$ indices mean the function value at the position of change and unchanged, F-rate$_{uc}$ refer to the detection of "unchangingness".

We found that, when the areas of the new objects less than 3% of the areas of the original objects, the change does not influence much the matching and thus the residual error respect to loop closure number. These tests with little changes on the map cause the recall values not equal to 1.0 in Table 1.

Another conclusion we can draw from our test is that increasing the offset to the original path can lead to variation in residual error trend because of the completely new route. While the proposed threshold gave satisfactory results for the 'same-path' cases (first row of Table 1), for the 'noisy-path' cases not so (third row). Here, we increased this threshold value to deal with the possible higher deviation (fourth row). This can be done, because the traversed path is known, but the effect of the new route planned to be considered in a more appropriate way in the future.

### 5.2 Real-life Experiments

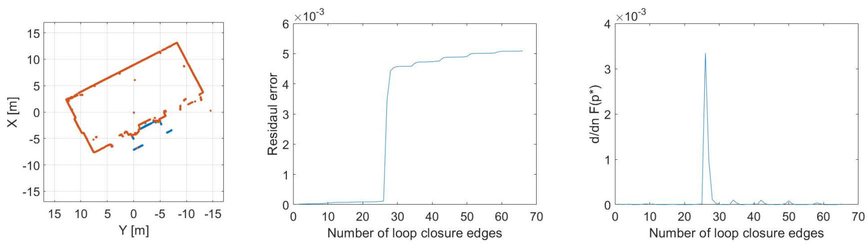
We did real-life tests with an automatized forklift equipped with Sick NAV350 sensor for navigation purposes. The equipment was provided and the test was conducted in the VVRL [3] The 2D point clouds acquired by the laser scanner was used for SLAM and the environment change localization. The AGV has done 5-5 rounds in the original and the changed environment (just as in some simulated

[3] Vehicle Vision Research Laboratory of the Faculty of Transportation Engineering and Vehicle Engineering's Department of Material Handling and Logistics Systems of Budapest University of Technology and Economics

test cases). The measurements in the different laps were executed approximately with the same position and orientation (as the vehicle circles on the same route in the production line). The results are presented in Fig. 5.

Table 1: Results based on different thresholds

| Noise - threshold values | Precision | Recall | F-rate | Average $\frac{\frac{d}{dn}F(p^*)_{xc}}{th}$ | Maximum $\frac{\frac{d}{dn}F(p^*)_{xuc}}{th}$ | F-rate$_{uc}$ |
|---|---|---|---|---|---|---|
| 0 m - $th$ | 1.0 | 0.83 | 0.91 | 22.14 | 0.57 | 0.99 |
| 0 m - 2.5 $th$ | 1.0 | 0.5 | 0.66 | 0.23 | 0.3325 | 0.99 |
| 0.5 m - $th$ | 0.46 | 1.0 | 0.63 | 14.29 | 2.22 | 0.96 |
| 0.5 m - 2.5 $th$ | 1.0 | 0.83 | 0.91 | 5.71 | 0.88 | 0.99 |




(a) Unchanged (blue) and changed (orange) env. (b) $F(p^*)$ (c) $\frac{d}{dn}F(p^*)$

Fig. 5: Real-life experiment

The predicted behaviour in the residual error function can be well-observed in both diagrams of Fig. 5. The extremum in the differential where the change in the environment perceived first, easily can be located as it exceeds the calculated threshold for this 'same-path' case. After the change has been localized, if one would like to find the exact change, it is enough to examine the corresponding loop closure edge and the two scans of its nodes.

## 6 Conclusion

In the paper, a methodology is proposed to automatically localize changes in SLAM generated maps and so avoid the brute force based feature matching. This is extremely useful in cases where mobile robots frequently traverse approximately the same path (e.g. AGVs in a production line, vehicles with a fixed path as a railway) or in case of more vehicles with an information network. Analysis of the optimization residual can be useful in many cases. The success of the proposed method was proven with many 2D test cases and real-life experiments. The method can be easily extended to 3D or adapt to other sensors. Some proof of concept tests of the extension was already made, in the future, we would like to develop this extension and consider the effect of completely different paths.

## Acknowledgment

The publication was supported by the European Commission through the Centre of Excellence in Production Informatics and Control (EPIC), grant No. 739592. The research was further supported by the Hungarian Scientific Research Fund No. OTKA/NKFIH K_120499.

## References

1. Alcantarilla, P.F., Stent, S., Ros, G., Arroyo, R., Gherardi, R.: Street-view change detection with deconvolutional networks. Autonomous Robots **42**(7), 1301–1322 (Oct 2018). https://doi.org/10.1007/s10514-018-9734-5
2. Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation **17**(3), 229–241 (June 2001). https://doi.org/10.1109/70.938381
3. Droeschel, D., Behnke, S.: Efficient continuous-time SLAM for 3D lidar-based online mapping. 2018 IEEE International Conference on Robotics and Automation (ICRA) pp. 1–9 (2018). https://doi.org/https://doi.org/10.1109/icra.2018.8461000
4. Durrant-Whyte, H., Roy, N., Abbeel, P.: A Linear Approximation for Graph-Based Simultaneous Localization and Mapping. MITP (2012). https://doi.org/https://doi.org/10.7551/mitpress/9481.003.0011
5. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 834–849. Springer International Publishing, Cham (2014). https://doi.org/https://doi.org/10.1007/978-3-319-10605-2-54
6. Grisetti, G., Kümmerle, R., Stachniss, C., Burgard, W.: A tutorial on graph-based SLAM. IEEE Intelligent Transportation Systems Magazine **2**, 31–43 (2010). https://doi.org/https://doi.org/10.1109/mits.2010.939925
7. Hess, W., Kohler, D., Rapp, H., Andor, D.: Real-time loop closure in 2D LIDAR SLAM. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 1271–1278 (May 2016). https://doi.org/10.1109/ICRA.2016.7487258
8. Khosoussi, K., Huang, S., Dissanayake, G.: Exploiting the separable structure of SLAM. In: Robotics: Science and Systems (2015). https://doi.org/https://doi.org/10.15607/rss.2015.xi.023
9. Lajoie, P.Y., Hu, S., Beltrame, G., Carlone, L.: Modeling perceptual aliasing in SLAM via discrete-continuous graphical models. IEEE Robotics and Automation Letters **PP**, 1–1 (01 2019). https://doi.org/10.1109/LRA.2019.2894852
10. Mendes, E., Koch, P., Lacroix, S.: ICP-based pose-graph SLAM. In: 2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). pp. 195–200 (Oct 2016). https://doi.org/10.1109/SSRR.2016.7784298
11. Mu, B., Liu, S.Y., Paull, L., Leonard, J.J., How, J.P.: SLAM with objects using a nonparametric pose graph. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 4602–4609 (2016). https://doi.org/https://doi.org/10.1109/iros.2016.7759677

12. Mur-Artal, R., Tards, J.D.: ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (Oct 2017). https://doi.org/10.1109/TRO.2017.2705103
13. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: Dense tracking and mapping in real-time. In: 2011 International Conference on Computer Vision. pp. 2320–2327 (Nov 2011). https://doi.org/10.1109/ICCV.2011.6126513
14. Oh, J.H., Jeon, J.D., Lee, B.H.: Place recognition for visual loop-closures using similarities of object graphs. Electronics Letters **51**(1), 44–46 (2015). https://doi.org/10.1049/el.2014.3996
15. Sakurada, K., Okatani, T., Deguchi, K.: Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 137–144 (June 2013). https://doi.org/10.1109/CVPR.2013.25
16. Sakurada, K., Okatani, T.: Change detection from a street image pair using CNN features and superpixel segmentation. In: BMVC (2015). https://doi.org/https://doi.org/10.5244/c.29.61
17. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H.J., Davison, A.J.: SLAM++: Simultaneous localisation and mapping at the level of objects. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1352–1359 (June 2013). https://doi.org/10.1109/CVPR.2013.178
18. Saputra, M.R.U., Markham, A., Trigoni, N.: Visual SLAM and structure from motion in dynamic environments: A survey. ACM Comput. Surv. **51**(2), 37:1–37:36 (Feb 2018). https://doi.org/10.1145/3177853
19. Torr, P.H.S., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding **78**, 138–156 (2000). https://doi.org/https://doi.org/10.1006/cviu.1999.0832
20. Ulusoy, A.O., Mundy, J.L.: Image-based 4-d reconstruction using 3-d change detection. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 31–45. Springer International Publishing, Cham (2014)
21. Underwood, J.P., Gillsj, D., Bailey, T., Vlaskine, V.: Explicit 3D change detection using ray-tracing in spherical coordinates. In: 2013 IEEE International Conference on Robotics and Automation. pp. 4735–4741 (May 2013). https://doi.org/10.1109/ICRA.2013.6631251
22. Wang, H., Huang, S., Frese, U., Dissanayake, G.: The nonlinearity structure of point feature SLAM problems with spherical covariance matrices. Automatica **49**(10), 3112 – 3119 (2013). https://doi.org/https://doi.org/10.1016/j.automatica.2013.07.025
23. Wang, H., Huang, S., Khosoussi, K., Frese, U., Dissanayake, G., Liu, B.: Dimensionality reduction for point feature slam problems with spherical covariance matrices. Automatica **51**, 149 – 157 (2015). https://doi.org/https://doi.org/10.1016/j.automatica.2014.10.114
24. Xiao, W., Vallet, B., Schindler, K., Paparoditis, N.: Street-side vehicle detection, classification and change detection using mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing **114**, 166 – 178 (2016). https://doi.org/https://doi.org/10.1016/j.isprsjprs.2016.02.007
25. Yu, C., Liu, Z., Liu, X.J., Xie, F., Yang, Y., Wei, Q., Qiao, F.: DS-SLAM: A semantic visual SLAM towards dynamic environments. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 1168–1174 (2018). https://doi.org/https://doi.org/10.1109/iros.2018.8593691