Towards Life-Long Autonomy of Mobile Robots Through Feature-Based Change Detection

Erik Derner[1,2,3], Clara Gomez[3], Alejandra C. Hernandez[3], Ramon Barber[3], and Robert Babuška[1,4]

***Abstract*— Autonomous mobile robots are becoming increasingly important in many industrial and domestic environments. Dealing with unforeseen situations is a difficult problem that must be tackled in order to move closer to the ultimate goal of life-long autonomy. In computer vision-based methods employed on mobile robots, such as localization or navigation, one of the major issues is the dynamics of the scenes. The autonomous operation of the robot may become unreliable if the changes that are common in dynamic environments are not detected and managed. Moving chairs, opening and closing doors or windows, replacing objects on the desks and other changes make many conventional methods fail. To deal with that, we present a novel method for change detection based on the similarity of local visual features. The core idea of the algorithm is to distinguish important stable regions of the scene from the regions that are changing. To evaluate the change detection algorithm, we have designed a simple visual localization framework based on feature matching and we have performed a series of real-world localization experiments. The results have shown that the change detection method substantially improves the accuracy of the robot localization, compared to using the baseline localization method without change detection.**

***Index Terms*— Life-long autonomy, change detection, mobile robots, localization, place detection, computer vision in robotics.**

I. Introduction

Mobile robots have become a key component for many tasks in the robotics domain, such as object manipulation and transportation, human-robot collaboration, or surveillance. Deployment of autonomous mobile robots in industrial and domestic environments poses a difficult challenge due to the dynamics of the environments. These challenges give a rise to the development of advanced methods that will be able to deal with changes occurring in the environment to perform tasks such as robot localization and navigation precisely and reliably. Novel approaches and suitable environment representations are being sought to allow for life-long autonomy of mobile robots in highly dynamic environments.

In this work, we present a novel approach for change detection based on local feature descriptors. The robot continuously monitors its environment and detects changes that have occurred. Upon detection of a change, the robot updates its representation of the environment to incorporate the information about the change. The key point consists in automatically learning the persistent regions of each scene, which remain unchanged over a long period of time.

The type of changes that we mainly consider in our work comprise moving chairs and items on tables, altering the picture on computer or TV screens, changing the contents of whiteboards and notice boards, opening or closing doors, adjusting blinds in the windows, etc. These changes occur every day in various industrial, domestic and office environments. Fig. 1 shows examples of such changes.

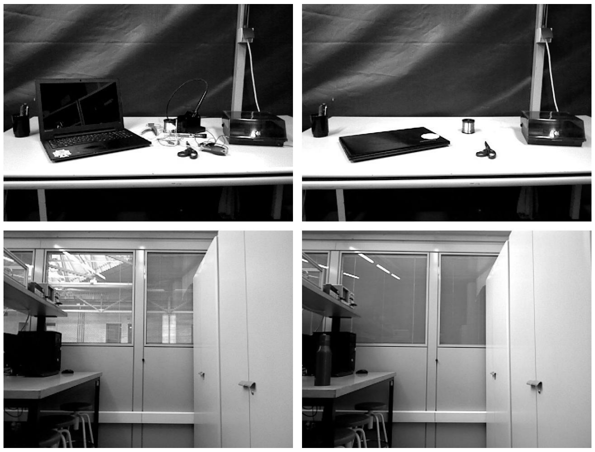


Fig. 1. Examples of changes that can be detected by the proposed algorithm. Such changes may confuse methods assuming static environments.

The concept of change detection introduced in our method can be used for robot localization and navigation, place detection, etc. It allows the robot to recognize its surroundings more reliably and therefore perform these tasks more precisely.

The paper is organized as follows. The related research in the field of change detection is presented in Section II. A baseline visual localization framework is introduced in Section III and the proposed change detection method is presented in Section IV together with its incorporation to the localization framework. The experimental evaluation is described in Section V and Section VI concludes the paper.

This work was supported by the European Regional Development Fund under the project Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000470), by the Grant Agency of the Czech Technical University in Prague, grant no. SGS19/174/OHK3/3T/13, and by HEROITEA: Heterogeneous Intelligent Multi-Robot Team for Assistance of Elderly People (RTI2018-095599-B-C21), funded by Spanish Ministerio de Economía y Competitividad.

[1] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Czech Republic.

[2] Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.

[3] Robotics Lab, Department of Systems Engineering and Automation, Carlos III University of Madrid, Spain.

[4] Cognitive Robotics, Faculty of 3mE, Delft University of Technology, The Netherlands.

Corresponding author: Erik Derner, erik.derner@cvut.cz.

978-1-7281-3605-9/19/$31.00 ©2019 IEEE

II. Related Work

Dynamic environments and changes in the environment have been perceived as a challenge in most robotic navigation contexts. In order to perform stable localization and path-planning, robots must take into account these changes [1]. Change detection has attracted the interest of many authors recently and different approaches have been proposed.

Most change detection algorithms are based on object detection and tracking in long-term operation [2], [3], [4], [5]. In [2] a robot patrols an indoor environment and detects movable objects by change detection and temporal reasoning. Their objective is to determine how many movable objects are there in the environment and track their position. The Rao-Blackwellized particle filter and the expectation-maximization algorithm are used to track the objects and learn the parameters of environment dynamics. In [3] a service robot is deployed in different indoor environments and a hierarchical map of the environment is maintained that takes into account the changes in object positions by comparing current object detections to mapped ones. In [5] the change detection problem is treated through reasoning about observations. Observations are classified considering long-term features, short-term features, and dynamic features, which correspond to mapped static objects, unmapped static objects, and unmapped dynamic objects respectively. Short-term features produce local adjustments to the belief about the trajectory of the robot, while long-term features generate global adjustments.

Other works directly detect changes and correspondences between robot views or images [6], [7], [8], [9], [10], [11], [12]. Full RGB-D views are used in [6] to build a map of the robot world. Changes between successive views are computed to discover the objects (moved areas) and learn them. Similarly, in [8], a Truncated Signed Distance Function (TSDF) grid and 3D reconstructions of the environment are maintained. New observations are aligned with previous ones and included in the new reconstruction. The new reconstruction is compared to the previous one in order to identify dynamic clusters between both reconstructions. Image views are used in [9] to detect changes using Gaussian Mixture Models (GMMs). As GMMs have long computational times, Vertical Surface Normal Histograms provide main plane areas which are discarded in the search of changes. Change detection is accomplished as the difference in the Gaussians generated for two images. Pointclouds from a LiDAR are compared to an octree-based occupancy map in [10] to obtain a set of changes. Change candidates are computed with the Mahalanobis distance and filtered to eliminate outliers. Authors in [11] proposed a 2D LiDAR-based framework for long-term indoor localization on prior floor plans. The system combines graph-based mapping techniques and Bayes filtering to detect significant changes in the environment. They use an ICP-based scan matching to determine the probability that a LiDAR scan related to a trajectory pose corresponds to the currently observable environment. This probability is used to improve the trajectory estimation through the update of the previous nodes. In [12] a method for life-long visual localization using binary sequences from images is proposed. It is assumed the idea of using sequences of images instead of single images for recognizing places. Features are extracted using global LDP descriptors to obtain the binary codes of each image. These binary descriptors are efficiently matched by computing the Hamming distance.

Change detection has also been broadly studied for outdoor environments [13], [14], [15]. Structural change detection from street view images is performed in [13]. Multisensor fusion SLAM, deep deconvolution networks and fast 3D reconstruction are used to determine the changing regions between pairs of images. In [14], a Bayesian filter is proposed to model feature persistence of road and traffic elements. Single-feature and neighbouring-feature information are used to detect changes in feature-based maps and estimate feature persistence. Big effort has been also made to overcome seasonal changes for outdoor environment navigation [16], [17], [18], [19]. In [16], HOG features and deep convolutional networks are used to compare and match the new acquired image with a database of images independently of the weather and seasonal conditions. The approach presented in [17] compares different variants of SIFT and SURF feature detectors in the frame of an appearance-based topological localization on panoramic images capturing seasonal changes.

Many algorithms need computationally demanding learning processes or the maintenance of heavy map reconstructions to perform change detection. On the contrary, the approach proposed in this paper relies on local feature detection and matching, which allows to run in real time on low-cost hardware platforms. While a lot of related methods deal with seasonal changes, we focus on changes that typically occur in indoor environments.

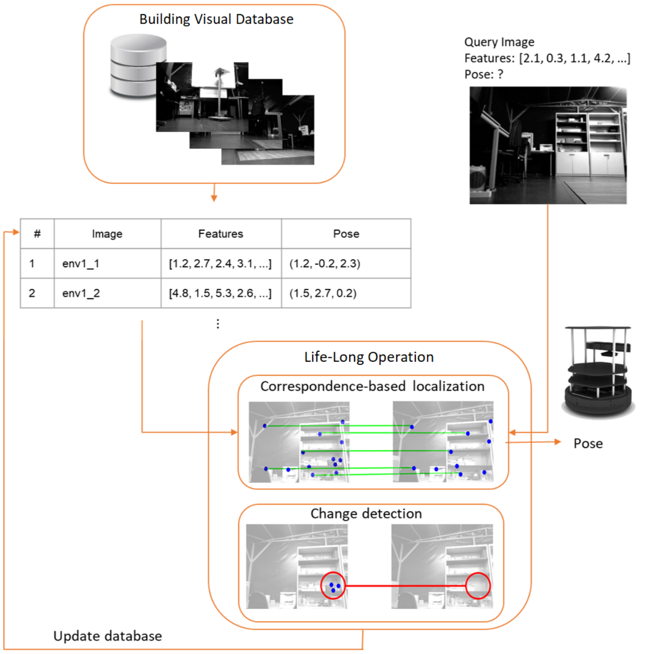
III. Visual Localization Framework

The change detection method proposed in this paper can be used for localization or place detection with various algorithms based on local features. In this section, we present a simple visual localization framework that will serve as a baseline and that will be later extended with the proposed change detection method.

An overview of the method is presented in Fig. 2. For now, we assume that the change detection module is not present and we introduce the baseline localization framework.

At first, a robot equipped with a camera builds a discrete representation of the environment in the form of a *visual database*, where images are stored together with their location coordinates. Note that a robot featuring a self-localization capability in a static environment needs to be employed to build the map (visual database). Standard methods based on wheel encoders and laser rangefinder readings can be applied to perform this task.

In the life-long deployment, the robot continuously localizes itself in the environment using feature matching against the previously built visual database. The matching procedure will be described in detail in Section III-B.



#	Image	Features	Pose
1	env1_1	[1.2, 2.7, 2.4, 3.1, ...]	(1.2, -0.2, 2.3)
2	env1_2	[4.8, 1.5, 5.3, 2.6, ...]	(1.5, 2.7, 0.2)

⋮

Fig. 2. Overview of the visual localization framework. The long-term localization uses a previously built visual database. Query images from the robot are matched against the visual database and the closest match determines the position of the robot. The change detection module monitors the changes in the matched images and updates the visual database when it detects a difference.

A. Building the Visual Database

First, we need to create the visual database that will serve as a reference for localization. A mobile robot equipped with a camera and using a standard localization method moves around the environment and records images and their locations. The visual database consists of individual records r_i, indexed by $i \in \{1, \ldots, N\}$, which have the following structure:

- a grayscale image I_i, captured by the camera mounted on the robot,
- a set of descriptors D_i of the points of interest P_i detected in image I_i, where $P_i = \{p_i^1, p_i^2, \ldots, p_i^{m_i}\}$ and $D_i = \{d_i^1, d_i^2, ..., d_i^{m_i}\}$,
- the coordinates $c_i = (x_i, y_i, \varphi_i)$ representing the location (pose) of the robot.

A robust feature detector and descriptor needs to be employed to detect points of interest and calculate their compact representation. We have chosen Speeded-Up Robust Features (SURF) [20], however, other transformation-invariant features could be used too.

B. Correspondence-Based Localization

Using the previously created visual database, the robot can be deployed and localize itself using the real-time feed of images from its camera that we refer to as the *query images*. The following steps are performed to localize the robot:

1) Capture a grayscale image I_q – the query image.
2) Run the SURF detector and descriptor on the query image. The set of descriptors on the query image I_q is denoted as $D_\mathrm{q} = \{d_\mathrm{q}^1, d_\mathrm{q}^2, \ldots, d_\mathrm{q}^{m_\mathrm{q}}\}$.
3) Match the set of descriptors D_q found in the query image against the sets of descriptors D_i stored with the database records r_i.
4) Report the location of the robot as the location c_{i^*} stored with the database record r_{i^*}, which achieved the highest ratio of correspondences $p_{\mathrm{q},i^*}/m_{i^*}$ among all records r_i in the database.

The index i^* of the database record r_{i^*} with the highest ratio of correspondences $p_{\mathrm{q},i^*}/m_{i^*}$ is determined by the following equation:

$$i^* = \operatorname*{argmax}_i \left(\frac{p_{\mathrm{q},i}}{m_i} \right), \tag{1}$$

where $p_{\mathrm{q},i}$ is the number of tentative correspondences between the query image and the database image I_i found by the matching algorithm and $m_i = |D_i|$, i.e., m_i is the number of descriptors stored with the database record r_i. It means that we are searching for a database record which will have the largest portion of its descriptors matched with the descriptors found in the query image.

IV. Change Detection Method

We propose a method for change detection that improves the life-long autonomy of mobile robots through maintaining an accurate and up-to-date representation of the environment. The essence of the method consists in learning the scene regions that are stable, distinguishing them from areas that are changing. This task is performed through change detection and results in a representation robust to changes in the environment. In the following text, we present the change detection method and we show how it is incorporated in the baseline localization framework.

A. Detecting Changes

The change detection algorithm is based on comparison of feature descriptors. We define a similarity measure between two descriptors d and d' as their Euclidean distance:

$$s(d, d') = \left|\left|d - d'\right|\right|_2 . \tag{2}$$

Note that the lower is the similarity measure, the more similar are the two features.

The outline of the change detection algorithm is as follows:

1) Based on the pairs of tentative correspondences found by the matching algorithm, use MSAC [21]to estimate the transformation between the query image I_q and the best-match database image I_{i^*}.
2) Transform the positions of the points of interest P_{i^*} in the best-match database image I_{i^*} to the coordinate frame of the query image I_q, yielding a set of transformed points of interest $\bar{P}_{i^*}$.
3) Calculate the SURF descriptors $\bar{D}_{i^*} = \{\bar{d}_{i^*}^1, \bar{d}_{i^*}^2, \ldots, \bar{d}_{i^*}^{m_{i^*}}\}$ of the transformed points of interest $\bar{P}_{i^*}$ in the query image I_q.

4) Calculate the similarity measure s_j between the SURF descriptors $d_{i^*}^j$ corresponding to the points of interest $p_{i^*}^j$ in the database image I_{i^*} and the SURF descriptors $\bar{d}_{i^*}^j$ of their projections $\bar{p}_{i^*}^j$ in the query image I_{q}.
5) For all $j \in \{1,\ldots,m_{i^*}\}$, if the similarity measure s_j is larger than a given threshold θ, the descriptor $d_{i^*}^j$ and the corresponding point of interest $p_{i^*}^j$ is removed from the database record.

The change detection is therefore based on computing the similarity measure between the descriptors $d_{i^*}^j$ calculated on the best-match database image I_{i^*} and their transformed counterparts $\bar{d}_{i^*}^j$ calculated on the query image I_{q}. Note that this is different from using the points of interest P_{q} and their descriptors D_{q} detected in the query image for comparison. As the transformed points of interest from the best-match database image and their descriptors are used instead, the algorithm becomes more robust to the precision and repeatability shortcomings of the feature detector.

The similarity measure (2) is adapted to the following form:

$$s_j = \left|\left|d_{i^*}^j - \bar{d}_{i^*}^j\right|\right|_2, \text{where } j \in \{1,\ldots,m_{i^*}\}. \tag{3}$$

The set of descriptors D_{i^*} and the respective set of points of interest P_{i^*} is then updated by removing the elements with the similarity measure above the threshold θ:

$$D'_{i^*} = D_{i^*} \setminus \left\{d_{i^*}^j : s_j > \theta \quad \forall j \in \{1,\ldots,m_{i^*}\}\right\}, \tag{4}$$

$$P'_{i^*} = P_{i^*} \setminus \left\{p_{i^*}^j : s_j > \theta \quad \forall j \in \{1,\ldots,m_{i^*}\}\right\}. \tag{5}$$

The updated sets D'_{i^*} and P'_{i^*} replace the original sets D_{i^*} and P_{i^*} in the database record r_{i^*}.

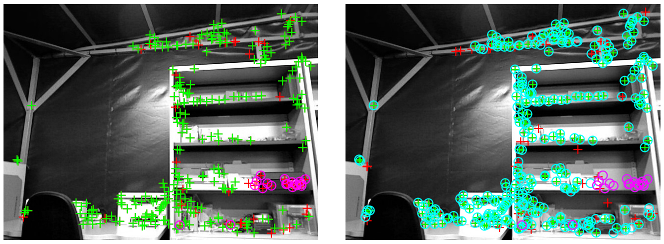


Fig. 3. A database image (a) and a query image with one object missing on the third shelf from the top (b). The crosses represent points of interest in both images. Tentative correspondences found by the matching algorithm are shown in green. Cyan circles show the transformations of the points of interest from the database image to the query image. Magenta circles show points of interest that were identified as a change.

Fig. 3 shows an example of a scene on which we can illustrate the principle of the change detection algorithm. An item, e.g. a toolbox, has been removed from one of the shelves after building the visual database. The change detection algorithm transforms the points of interest from the database image to the query image and calculates their SURF descriptors. They are then compared to the corresponding SURF descriptors in the database image. Since the descriptors in the region of the toolbox have the similarity measure above the threshold, they are removed, together with the respective points of interest, from the respective sets linked to the database image. Note that as a by-product, some unstable features may be removed as well.

B. Life-Long Operation

The localization framework presented in Section III can be now extended by the change detection module. An overview of the life-long localization is shown in Fig. 2. In the life-long operation, the robot continuously localizes itself in the environment and maintains its visual database up to date by incorporating changes detected in the environment.

An important condition that makes the change detection method efficient is that wrong matches (localization failures) need to be avoided as much as possible, because incorporating false positive changes in wrong matches decreases the quality of the visual database. To avoid this, we introduce two conditions that serve as a *confidence criterion*: a spatial and a temporal condition.

The spatial condition calculates the mean Euclidean distance m_s (in the (x,y)-space of the robot) between the best match and the n_s successive most similar matches of the query image with the images in the visual database (starting with the second-most similar match) and compares it with a reference mean of distances m_r. The reference mean m_r is calculated as the mean Euclidean distance of the best match and n_r closest database records in the (x,y)-space, where the best match itself is excluded from this set. Typically, the number of closest reference points n_r is set to be several times larger (e.g. 5×) than n_s. If the mean distance of the most similar matches m_s is larger than the reference mean distance m_r, the confidence criterion is not met. Note that the reference means m_r can be pre-calculated offline for all database records for efficiency.

The temporal condition checks if the distance between the current location and the previous location is smaller than a given threshold δ. It takes into account the physical limitations of the robot and discards unreliable matches in cases when the robot appears to have moved further than it possibly could. The temporal condition is tested only if the spatial condition was met both for the current and for the previous image, which allows for recovery after a localization failure.

Only if both conditions are met, the change detection module is run and the visual database is updated. This way, the chance of incorporating false positive changes on wrongly matched images is minimized.

The presented approach allows for localization in unstructured dynamic environments even in cases when standard localization methods based on a static map would fail. Note that the amount of changes between the moment of building the visual database and the current state of the environment can be large. The method can deal with such changes thanks to the adaptation to gradually emerging changes by continuously updating the visual database.

V. Experiments

We have chosen the popular robotic platform TurtleBot 2 to validate our method. The robot is equipped with a camera Asus Xtion PRO LIVE, which provides RGB and depth images. However, only grayscale images are used in our method. We have used an extension attached to the top of the TurtleBot in order to fix the camera at a higher position, as can be seen in Fig. 4. The position of the robot is captured through odometry based on wheel encoders.

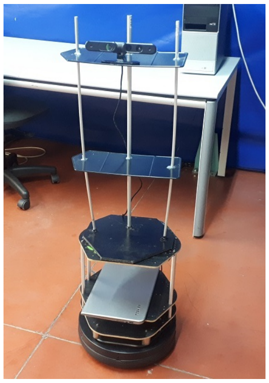


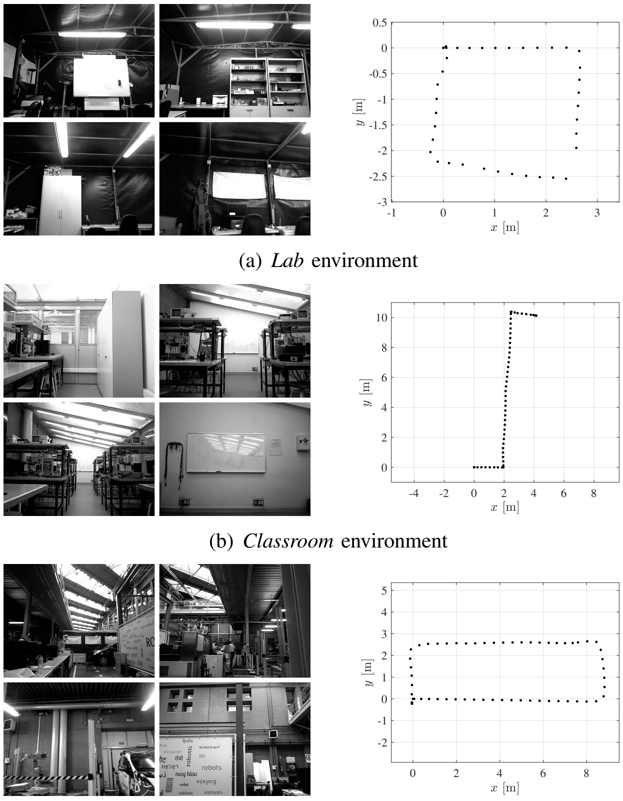
Fig. 4. Mobile robot TurtleBot 2 equipped with a camera used in the experiments.

A. Environments

Due to the lack of publicly available data sets capturing dynamic environments, which would be well suited for the type of changes that we focus on in this paper, we have created three data sets in different environments at the Carlos III University in Madrid. The data sets in each of the environments – *Lab*, *Classroom*, and *Hall* – consist of multiple sequences. The sequences have been recorded on different days and at different times of the day, capturing various changes in the environment (moving chairs and items on the desks, changing the picture on computer screens, opening and closing window blinds, etc.). The trajectories and examples of images for each environment are shown in Fig. 5. The number of images and the length of the trajectories for each of the sequences are given in Table I.

TABLE I
PROPERTIES OF THE IMAGE SEQUENCES USED IN THE EXPERIMENTS.

Data set	Sequence	Images	Length
Lab	L-DB (database)	41	10.0 m
	L-Q1 (query)	89	10.5 m
	L-Q2 (query)	85	10.2 m
	L-Q3 (query)	84	9.7 m
	L-Q4 (query)	103	10.0 m
Classroom	C-DB (database)	57	14.1 m
	C-Q1 (query)	113	14.2 m
	C-Q2 (query)	106	13.9 m
	C-Q3 (query)	103	14.0 m
Hall	H-DB (database)	58	22.5 m
	H-Q1 (query)	58	7.1 m
	H-Q2 (query)	57	6.9 m
	H-Q3 (query)	59	6.9 m



(a) *Lab* environment

(b) *Classroom* environment

(c) *Hall* environment

Fig. 5. Examples of images (left) and trajectories (right) from the environments used in the experiments.

B. Experimental Setup

At first, we have constructed visual databases L-DB, C-DB and H-DB for the environments *Lab*, *Classroom*, and *Hall*, respectively. The playback of the recorded query sequences served as a stream of query images in real time. Each query image was matched with the most similar image in the visual database and the coordinates associated with the most similar database image were returned as the pose of the robot. If the conditions for confidence (as described in Section IV-B) were met, the record of the best match in the visual database was updated with the detected changes.

In all experiments, we have used the following default configuration. The similarity threshold for change detection was set to $\theta = 0.5$. With regards to the confidence criterion, the number of closest samples n_s was set to 2 and the number of reference samples $n_r = 10$. The temporal difference tolerance was set to $\delta = 0.5\,\mathrm{m}$. We have empirically evaluated that the default values work well for all tested scenarios. However, they may be adjusted for improved performance on data sets with substantially different properties.

C. Results

We have executed a cascade of subsequent runs of the lifelong localization algorithm on query sequences from all three environments. We have compared the root-mean-squared (RMS) localization errors on query sequences matched against the original visual databases and against visual databases that have been updated by executing localization with change detection on one of the query sequences. The results are summarized in Table II, where '–' in the 'Updated on' column means that the original visual database was used, i.e., without any updates made based on the changes observed in the environment.

The RMS localization errors are shown in Table II. The results show that the change detection algorithm leads to an improved localization accuracy. Employing the change detection yields an average improvement of 38 % on the query sequences from the *Lab* environment, 24 % for the *Classroom* environment and 28 % for the *Hall* environment.

TABLE II
LOCALIZATION RMS ERRORS ON DIFFERENT QUERY SEQUENCES. SEQUENCES EVALUATED ON A DATABASE UPDATED WITH CHANGES DETECTED IN A PREVIOUS SEQUENCE ARE DISPLAYED IN BOLD.

Visual database	Updated on	Query sequence	Localization RMS error
L-DB	–	L-Q2	0.56 m
L-DB	**L-Q1**	**L-Q2**	**0.35 m**
L-DB	–	L-Q3	0.52 m
L-DB	**L-Q1**	**L-Q3**	**0.26 m**
L-DB	–	L-Q4	0.66 m
L-DB	**L-Q1**	**L-Q4**	**0.49 m**
C-DB	–	C-Q3	0.63 m
C-DB	**C-Q1**	**C-Q3**	**0.54 m**
C-DB	**C-Q2**	**C-Q3**	**0.42 m**
H-DB	–	H-Q2	0.64 m
H-DB	**H-Q1**	**H-Q2**	**0.54 m**
H-DB	–	H-Q3	0.91 m
H-DB	**H-Q1**	**H-Q3**	**0.54 m**

The processing time of a single query image is 200–300 ms on a standard laptop[1] for all experiments.

VI. CONCLUSIONS

We have proposed a method for change detection based on comparison of local visual features and we have shown how the change detection method can be incorporated into a simple localization framework. We have introduced two conditions that evaluate the confidence of a correct localization. This way, we avoid decreasing the quality of the visual database by introducing changes from wrongly matched images. The experimental evaluation has shown that updating the representation of the environment with the information about the changes leads to a considerably more accurate localization.

In the future work, we plan to extend the method to make it more robust, e.g. by adding features for objects newly present in the scene and introducing feature weights to distinguish between short-term and long-term changes. We will also compare our method with alternative state-of-the-art approaches used for localization.

A possible line of future research would be to incorporate the semantic information into the change detection algorithm, e.g. by applying an object detection method to determine the changes in the scenes based on object occurence.

[1] CPU Intel Core i7-4610M (2 cores @ 3.0 GHz), 16 GB RAM

REFERENCES

[1] R. Alterovitz, S. Koenig, and M. Likhachev, "Robot planning in the real world: research challenges and opportunities," *Ai Magazine*, vol. 37, no. 2, pp. 76–84, 2016.

[2] N. Bore, P. Jensfelt, and J. Folkesson, "Multiple object detection, tracking and long-term dynamics learning in large 3d maps," *arXiv preprint arXiv:1801.09292*, 2018.

[3] L. Kunze, H. Karaoguz, J. Young, F. Jovan, J. Folkesson, P. Jensfelt, and N. Hawes, "Soma: A framework for understanding change in everyday environments using semantic object maps." AAAI, 2018.

[4] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, J. Wyatt, D. Hebesberger, T. Kortner, *et al.*, "The strands project: Long-term autonomy in everyday environments," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 146–156, 2017.

[5] J. Biswas and M. Veloso, "Episodic non-markov localization: Reasoning about short-term and long-term features," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3969–3974.

[6] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard, "Toward lifelong object segmentation from change detection in dense rgb-d maps," in *2013 European Conference on Mobile Robots*. IEEE, 2013, pp. 178–185.

[7] B. Bescós, J. M. Fácil, J. Civera, and J. Neira, "Dynslam: Tracking, mapping and inpainting in dynamic scenes," *arXiv preprint arXiv:1806.05620*, 2018.

[8] M. Fehr, F. Furrer, I. Dryanovski, J. Sturm, I. Gilitschenski, R. Siegwart, and C. Cadena, "Tsdf-based change detection for consistent long-term dense reconstruction and dynamic object discovery," in *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE, 2017, pp. 5237–5244.

[9] P. Drews, L. J. Manso, S. da Silva Filho, and P. Núñez, "Improving change detection using vertical surface normal histograms and gaussian mixture models in structured environments," in *2013 16th International Conference on Advanced Robotics (ICAR)*. IEEE, 2013, pp. 1–7.

[10] L. Wellhausen, R. Dubé, A. Gawel, R. Siegwart, and C. Cadena, "Reliable real-time change detection and mapping for 3d lidars," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, 2017, pp. 81–87.

[11] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, "A pose graph-based localization system for long-term navigation in cad floor plans," *Robotics and Autonomous Systems*, vol. 112, pp. 84–97, 2019.

[12] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Towards life-long visual localization using an efficient matching of binary sequences from images," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6328–6335.

[13] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, "Street-view change detection with deconvolutional networks," *Autonomous Robots*, vol. 42, no. 7, pp. 1301–1322, 2018.

[14] F. Nobre, C. Heckman, P. Ozog, R. W. Wolcott, and J. M. Walls, "Online probabilistic change detection in feature-based maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[15] B. Neuman, B. Sofman, A. Stentz, and J. A. Bagnell, "Segmentation-based online change detection for mobile robots," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 5427–5434.

[16] T. Naseer, W. Burgard, and C. Stachniss, "Robust visual localization across seasons," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 289–302, 2018.

[17] C. Valgren and A. J. Lilienthal, "Sift, surf and seasons: Long-term outdoor localization using local features," in *3rd European conference on mobile robots, ECMR'07*, 2007, pp. 253–258.

[18] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett, "Recurrent-octomap: Learning state-based map refinement for long-term semantic mapping with 3-d-lidar data," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3749–3756, 2018.

[19] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli, "Learning context flexible attention model for long-term visual place recognition," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4015–4022, 2018.

[20] "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[21] P. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.