A Hybrid Map with Permanent 3D Wireframes and Temporal Line Segments toward Long-Term Visual Localization

Tomoya Kaneko *, Junji Takahashi **, Seiya Ito *, and Yoshito Tobe *

Abstract: This paper deals with map-construction problems for visual localization. Basically, the map is an aggregation of visual landmarks, and it is desirable that such landmarks exist densely and permanently for long-term localization. However, there are no landmarks with these attributes at the same time. In order to solve this problem, we propose a hybrid map with permanent landmarks and temporal landmarks. As a permanent landmark, we employ 3D wireframe which can be easily obtained from architectural CAD. For a temporal landmark, we use line segments which are visually detected in images captured by a camera. To handle these two types of landmarks on the same map, we develop two algorithms. One is to extract temporal line segments from images containing two mixed landmarks, and the other is to reconstruct them into the 3D wireframe map. We experimentally demonstrated that the proposed hybrid map outperformed the 3D wireframe map in terms of localization accuracy.

Key Words: visual localization, 3D pre-constructed map, hybrid landmarks, line segments.

1. Introduction

Localization technology is indispensable for robot navigation and internet of things (IoT) services [1]. However, there does not exist any critical localization method with low computational resources and low maintenance cost, that would provide an accurate position for mobile devices and robots for a long period of time.

Algorithms of simultaneous localization and mapping (SLAM) can simultaneously perform self-localization and environmental map construction using sensors, such as monocular cameras and red-green-blue-depth (RGB-D) cameras, and they have been extensively studied [2]–[5]. One advantage of SLAM algorithms is that a robot can be aware of its own location even in unknown environments. Meanwhile, because both localization process and map construction process depend on each other, the errors continue to increase. Furthermore, its calculation cost is enormous. Smart house approaches, that utilize radio frequency identifier (RFID) [6] or WiFi [7] signals from devices embedded in the environment to measure mobile object or robot, are not so accurate in localization. In addition, these approaches involve significant costs for installation and maintenance. If we limit a target environment to an indoor situation, we can easily construct 3D pre-maps using architectural drawings for localization. The localization using pre-maps leads to reduction in computational cost and provides precise localization [8]. Therefore, we tackle the problems of localization using a pre-map.

The most important question of map-based localization is what is the best representation of the map. Since a map is an aggregation of landmarks, this question is equal to what kind of feature is best for landmarks of the map. One of the important attributes of landmarks for a pre-map is time-invariance. Since there is basically a large time gap between the time when a pre-map is constructed and the time when it is used for localization operation, landmarks in the pre-map must not change nor disappear to be detected by a camera as always. On the other hand, for effective localization operation, it is also important for landmarks to be densely detected everywhere. In most cases, the number of detected landmarks in one sensing (scan or shot) directly affects the accuracy of localization operation. For instance, almost public buildings have periodic structure so that the results of localization tend to be confused among
similar appearance scene only by less type of landmarks. However, if rich types of landmarks are utilized for the localization process, such a confusing case can be avoided. The more densely landmarks are detected, the more robustly localization algorithm becomes as against the periodic structure, observing errors and visual occlusions in the practical scene. However, there is no landmark that possesses these two attributes simultaneously. For example, line segments caused by boundaries among a floor, a wall and a ceiling can be regarded as permanent landmarks, but they are not densely detected. The visual point features such as SIFT, SURF, AKAZE, etc., can be densely detected in a messy situation, but the life-span as a landmark is not enough long; they cannot survive for more than weeks.

In order to realize an effective and long-life localization system, we propose a novel hybrid map that consists of two types of landmark (Fig. 1). The one is a 3D wireframe from architectural CAD as a permanent landmark. The other is line segment feature visually detected by an RGB camera. This hybrid map has great advantages. Firstly, the data size is very small and the compatibility is nice because the landmarks can be recorded as a text-based file. Secondly, these two types of landmarks can be treated in the same way in the localization process. Especially they can be used by our previously proposed image-retrieval-based localization algorithms [9], [10]. Thirdly, multiple types of landmark are robust for periodic structure. Even if one type of landmarks is detected periodically, another type of landmarks, which is rarely detected in the same period, prevents confusion.

The rest parts are organized as follows. In Section 2, related work of visual localization and the difference of this paper against them are described. In Section 3, we give a brief description of our proposing cloud-based localization infrastructure system named Universal Map and explain the position of this paper in it. In Section 4, the method to merge detected line segments into a pre-constructed 3D wireframe map is described. This method includes the classification process of line segment types and 3D reconstruction process. Section 5 describes experiments and results. In Section 6, we summarize this paper and mention our future work.

2. Related Work

Research on localization for a pre-map has been studied for a long time. The pre-map proposed by early researchers was a simple database of pre-captured view images [11]. The localization process was treated as an image retrieval problem that finds the closest image to the query in the database. On the bounty of keypoint technique and the bag-of-words technique, the image-retrieval-based localization approach can cover a large area with high estimation accuracy [12]. In [13], they estimated the position using a database of large amounts of RGB-D images. However, the above techniques are costly to develop a pre-map and not robust against the environment changes.

The structure from motion (SfM) [14] approach is useful for constructing a pre-map and a localizing a query image at the same time. Multiple images and a query image are localized by solving perspective-n-point (PnP) problem of 3D map. For landmarks, point features [15], [16] are common, but the line segment feature [17] has also been studied. Micusik et al. focused on the fact that line segments were easily detected at the indoor environment and constructed a large-scale 3D line segments map with line-based SfM [18]. Then they developed a localization algorithm using line segments map [19]. However, these methods of constructing a 3D map of line-segments cannot cope with the change when a new landmark is added to the real environment.

The approaches to update the map to deal with changes in the real environment and perform long-term localization of the robot have been proposed in recent years [20]. Dymczyk et al. proposed scoring functions for measuring landmark utility so that the summarized map would be available for long-term localization [21]. In [22], the map was represented as 2D spatio-temporal occupancy grid, and they tried to construct a map that corresponds to the change in the real environment. The temporal map was constructed for each robot of a multi-robot system, and the map was updated by merging the map [23]. In [24], they tried to realize long-term mapping by adding new landmarks to the map using a laser sensor. Konolige et al. [25] proposed a system to add new landmarks to the 2D map using a stereo camera. However, with these map update methods, expensive sensors must be used, and maps are limited to 2D.

3. Universal Map

3.1 Concept and Advantages

The conceptual figure of our proposed Universal Map system is drawn in Fig. 2. This system consists of three subsystems: a central server which maintains the map composed of universal landmarks, clients who access the server so that they obtain positions of themselves, and agents who detect and report environment changes to the server. The client usually uploads its current sensing data; then the server localizes the sensing data in the map; finally, the localization result will be downloaded to the client.

This universal map system has many advantages. Since the map is maintained on the server and the localization calculation is conducted on the server side, the client can conserve the computational resources as well as energy resources. The client can avoid the maintenance for keeping the map new. In the position of the supervisor of this universal map infrastructure system, being unnecessary of physical maintenance other than a workstation machine and communication infrastructure is a great benefit. By the contribution of the agents, the map on the central server is always kept new so that this localization infrastructure system stays alive almost permanently.

The work described in this paper becomes an important part of several roles of the agent in the scenario of the universal
map. An agent engages the universal map with two different modes depending on its equipment. When an agent equipped precise localization technology other than the universal map, it reports newly observed landmarks with simply associating the position information given by that technology. This is named as the well-armed agent mode. On the other hand, when an agent does not have any localization method other than the universal map, the agent has to conduct two tasks: self-localization and report. The latter, which is named as the light-armed mode, is difficult to work well, but this mode is very effective because any types of the light-armed client can take a role as an agent in the light-armed mode. In preparation for the case of a light-armed client, we explain 2D camera image localization in the following subsection.

### 3.2 2D Camera Image Localization in 3D Wireframe Map

We explain our developed 2D-3D line-segments matching algorithm. The algorithm is to estimate the global position from one shot image at the known indoor environment.

As preprocessing, various position and direction of perspective 2D images are generated from the 3D wireframe map (Fig. 3). This image generating process utilizes open graphics library (OpenGL) drawing functions and gets executed on the server machine. Each generated image has the same angle of view and the same image aspect ratio as the camera of the server machine. Each generated image has the same angle of view and the same image aspect ratio as the camera of the server machine.

Then, n-bit grayscale distance transform images are generated from the 2D images. Let \( \text{dist} \) denote the value at the same pixel position of the distance transform image, \( r \) denote the searching range for non-zero pixels. The pixel value after grayscale distance transform is given as follows:

\[
p = \left\lfloor (2^n - 1) \left(1 - \frac{\min(r, \text{dist})}{r}\right) \right\rfloor.
\]

Note that the pixel value of \( p \) must be an integer, so the floor function is applied to truncate the decimal point. We name the set of these 2D distance transform images as a database (DB).

Figure 4 shows an example of DB images.

We acquire a query image uploaded by a client equipped with an RGB camera. From the query image, the line-segment-image is generated by a line segment detector (LSD) [26]. The similarity-based localization procedure is as follows. The line segment image pixel values \( \{0,1\} \) performing a logical operation is converted into two types of pixel values of \( \{0,2^n - 1\} \) in advance. A logical-conjunction-image between the line-segment-image and a DB image is calculated one by one. The line segment image (1-bit) is converted into \( n \)-bits grayscale image before the calculation. Let \( A_{db}(j,k) \) denote the number of non-zero pixels of the \( i \)-th DB image, \( \text{line}(j,k) \) denote an \( n \)-bit line-segments-image, and \( db(i,j,k) \) denote the \( i \)-th DB image. Then the matching ratio \( m_i \) is given as follows:

\[
m_i = \frac{\sum_k \text{line}(j,k) \land db(i,j,k)}{(2^n - 1)A_{db}},
\]

where \( (j,k) \) represents pixel coordinate in an image.

Next, the process searches the logical-conjunction-image which has maximum \( m_i \) and then registers its DB image as a best-matched DB image. The position of the best-matched DB image is the resulted client position. The policy of best-matched DB image selection is

\[
i_{best} = \arg \max_i m_i.
\]

Figure 5 shows an example of each image used for matching. In Fig. 5 (d), the green lines are derived from the line-segment-image (generated from query image), the red lines are derived from the DB image, and the white line represents the result of their logical product.

### 4. Proposed System

#### 4.1 System Overview

This section describes the system of adding new landmarks to the 3D wireframe map. We define two landmark categories: a structure edge which is a category of time-invariant landmarks such as borders among floor, wall, and ceiling; a color edge which is a category of temporal landmarks such as borders of poster or pattern on floor. The structure edge includes casing lines of window and door that are drawn on architectural CAD data. The process flow of adding color edges is drawn in Fig. 6. The query image is localized by the matching algorithm described in Section 3.2. By using the best-matched DB image, the line segments from query image are classified into structure edges or color edges or noise. The classified color edges are converted into 3D coordinate to be reconstructed and be merged into the 3D wireframe map. The detail of classification and 3D reconstruction are described in the following subsections.
4.2 Classification

The classification algorithm utilizes the best-matched DB image. The structure edges and color edges are obtained by masking processes between the line-segment-image form query and the best-matched DB image. The remaining pixels of the resulting logical conjunction image are the candidate of the structure edge. The line segments composed of remaining pixels are compared with the line list, which is made on ahead during the line detection process, and are determined as structure edges if similar lines exist in the list. On the other hand, the candidates of color edges are obtained as relative complement pixels of the best-matched DB image in the line-segment-image from the query. The candidates are compared in the list and determined by the existence of a similar line. This comparing process performs as filtering out minute noises. The information of structure edges and color edges are managed in the MySQL. The graphical explanation is given in Fig. 7.

4.3 3D Reconstruction

In 3D reconstruction, line segments, which are classified as a color edge, are converted to 3D coordination and added to the 3D wireframe map. Let \( A_c = (x_c, y_c, z_c)^T \) denote a point on the camera view coordinate. The view conversion matrix \( M \) is defined by the viewing location \( e = (x_v, y_v, z_v)^T \), the point of gaze \( g = (x_g, y_g, z_g)^T \), and an upper direction vector \( u = (x_u, y_u, z_u)^T \). The projection transformation matrix \( P \) is defined by the angle of view, the aspect ratio of the window, the depth of near plane of projection and far plane of projection. The viewport matrix \( U \) is defined by the width and the height of the image window. We define \( S = (S_x, S_y, S_z) \) as the coordinate of the starting point from which the projection line starts. We define \( V = (V_x, V_y, V_z) \) as the unit matrix representing the direction of the projection line. We define \( t \) as the parameter of the projection line. The projection line with no depth information from the \( A_c \) with no information on \( z_c \) as follows:

\[
 tV + S = (U \cdot P \cdot M)^{-1} \begin{bmatrix} x_c \\ y_c \\ z_c \\ \end{bmatrix}.
\]  

(4)

In Eq. (4), the obtained calculation result is not a point but a line on the 3D map coordination. Actually, the starting point \( S \) is an arbitrary point on the projection line, and the direction vector \( V \) is the difference between \( S \) and another point on the projection line. Against the problem of not knowing the depth, we use the fact that the color edges are always on the wall. So as to complement the lost depth information, we introduce the assumption that the 3D coordinate of the end point of the color edge is located on the wall plane of the 3D map that the projection line intersects firstly (Fig. 8). Let \( P = (x_p, y_p, z_p) \) denote the position vector of the intersection point, \( P_0 = (x_{p0}, y_{p0}, z_{p0}) \) denote the position vector of an arbitrary point of the wall plane, \( N = (x_n, y_n, z_n)^T \) denote the normal vector of the wall plane. The wall plane can be defined as follows:

\[
 (P - P_0) \cdot N = 0.
\]  

(5)

Using Eqs. (4) and (5), the parameter \( t \) is given as follows:

\[
 t = \frac{-(S \cdot N - P_0 \cdot N)}{(V \cdot N)}.
\]  

(6)

By substituting Eq. (6) into Eq. (4), the 3D coordinate of the end point of the line segment which is the point of intersection of the projection line and the wall plane is calculated if it exists.

However, on the wall plane on which the end point of the line segment is projected, it is necessary to select one wall plane from all number of wall planes in the 3D wireframe map. For example, in the case of the experiment in Section 5, the number of wall planes is 1634. Therefore, for each wall plane, the projection plane of the endpoint is selected by performing the following filtering processing. First of all, the processing verifies whether it is a wall that intersects the projection line. It then verifies that the point of intersection is on the front side of the camera and the coordinate of the point of intersection is within the wall plane. Then the Euclidean distances from the camera position to the point of intersection on every plane that is successfully verified by the former processing are calculated. After finishing the processing of all the wall planes, the wall plane that has the smallest distance from the camera is selected, and the intersection between the projection line and the wall is set as the endpoint of the line segment.
5. Experiments

We evaluated our proposed system by comparing the localization accuracy in the case of only using 3D wireframe map with that in the case of using the hybrid map which consists of structure edges and color edges. In the experiments, we also investigated the effects of different camera angles and effects of different agent modes (well-armed mode versus light-armed mode).

5.1 Conditions

For the verification of the proposed system, we prepared a dataset including 51 picture images taken at the actual environment with the actual position information as the ground truth. In this experiment, we targeted the corridor of the 5th floor of O-building in Aoyama Gakuin University (Fig. 9). The images in the dataset were taken by a smartphone (Xperia, Sony Mobile Communications Inc.) to include a poster (0.9m × 0.6m) on the wall as a sample of temporal landmarks. The height of the camera was fixed at 1.2 m from the floor and the tilt angle was fixed to 0° (horizontal direction). We prepared 3 directions in crossing angle to the wall, such as 30°, 45°, 60° as shown in Fig. 10. Seventeen pictures in each angle, that is, 51 pictures in total were taken for the dataset.

The DB images were generated in the range of 7.1m in x-direction, 0.5m in y-direction, and 0.2m in z-direction. The image generated interval is 0.1m in all directions. The target area is indicated as a bold rectangle in Fig. 9. In each position, three types of angles between the wall and camera direction, 30°, 45°, 60° were considered as shown in Fig. 10. Additionally, −2° to +2° direction images were generated in each angle for covering sensing error. Basically, 15 directions of images were generated in each position. The total number of DB images were 10,650. The parameter of range for searching non-zero pixels to make 2D distance transform images was set \( r = 16 \).

5.2 Procedures

For each angle, we conducted cross-validation. First, eight randomly selected images were utilized like training data for extracting color edges and for merging them into 3D wireframe map to construct a 3D hybrid map. Next, the new DB images were generated from the new 3D hybrid map. Finally, all of the remaining 43 images were used for localization test.

In the process of updating the map, we assumed two agent modes, such as the well-armed mode and the light-armed mode. At the well-armed mode, the agent knows its own position, that is, the ground truth position data is available for reconstructing color edges. On the other hand, at the light-armed mode, the ground truth position data is not available, and the agent has to estimate its position with using universal map system before extracting and reconstructing color edges. By this comparison, we can discuss the importance of agent mode more deeply.

5.3 Results and Discussion

The examples of constructed hybrid maps by the well-armed agent mode are shown in Fig. 11. At a glance, there are not large differences in every angle case in reconstruction accuracy. All angle cases have similar reconstruction error. Figure 12 shows an example of the constructed hybrid map by the light-armed mode. There is obviously a large reconstruction error, which is marked in the circle. This is thought to be due to a location estimation error in the localization process. This misguided reconstruction result may mislead the localization process when this hybrid map is used for localization as a universal map. Incidentally, the landmarks detected on the ceiling were
Table 1 The average error of localization result.

| angle (°) | Normal map (3D wireframe) m (S.D.) | Hybrid map (well-armed) m (S.D.) | Hybrid map (light-armed) m (S.D.) |
|----------|-----------------------------------|----------------------------------|----------------------------------|
| 30       | 0.48 (0.84)                       | 0.21 (0.46)                      | 0.28 (0.63)                      |
| 45       | 0.57 (0.91)                       | 0.27 (0.59)                      | 0.35 (0.72)                      |
| 60       | 0.60 (0.92)                       | 0.20 (0.45)                      | 0.28 (0.58)                      |

The resulted localization errors on every map case and angle are described in Table 1. Obviously, it is confirmed that the localization error becomes smaller in the case of the hybrid map than that of normal 3D wireframe map. In comparison between the well-armed mode and the light-armed mode, the well-armed mode is better in any angle cases. The reason why the light-armed mode increases the error is thought that the misguided reconstruction due to location estimation error.

We consider the error that camera angle affects. From Table 1, it can be seen that the accuracy of the hybrid map improves from any angle of 30° to 60° than that of the normal map, and it shows that there is not much influence on accuracy by angle. From Figs. 13, 14, and 15, the ratio of the estimation error exceeding 5.0 m decreased from the normal map by the hybrid map (well-armed) is 11% at 30°, 11% at 45°, 15% at 60°. In the case of light-armed mode, it was 7% at 30°, 7% at 45°, 13% at 60°. Therefore, it was confirmed that the precision of map updating was not greatly decreased depending on wall and camera angle. In other words, the reliability of the update data when the angle with the wall is 30° to 150° can be considered to be uniform. However, if the angle is even smaller, it is expected that the accuracy will decrease. It is one of the future tasks to determine from which angle the accuracy drops.

In the update process of the map information, the delete process of non-existing landmarks is the same importance as the add process of existing landmarks. The difficulty of the delete process is how to detect or measure the existence of temporal landmarks. We are considering two approaches for identifying landmarks that are no longer existed. The first approach is to introduce an excellent deletion agent with rich sensors. The second approach is to introduce an evaluation function to calculate how much the color edge was used in client localization. This issue is one of the important future studies.

6. Conclusion

We proposed the system which updates the pre-constructed 3D map by adding new landmarks from the images taken by the cameras. We developed two algorithms which are to extract temporal landmarks from images and to reconstruct their 3D positions for updating the pre-constructed 3D map. In our experiments, the proposed hybrid map improved the localization accuracy of the original 3D map. In the future, we will develop a map addition method considering the angle and aim to be a system that can add the map more accurately. Furthermore, we will also develop a delete process of the landmarks that have not been used on the map.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP16K21339 and SECOM grant.

References

[1] S. Thrun, W. Burgard, and D. Fox: Probabilistic Robotics, MIT Press, 2005.
[2] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse: MonoSLAM: Real-time single camera SLAM, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 29, No. 6, pp. 1052–1067, 2007.
[3] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardós: ORB-SLAM: A versatile and accurate monocular SLAM system, IEEE Trans. Robot., Vol. 31, No. 5, pp. 1147–1163, 2015.
[4] J. Engel, T. Schöps, and D. Cremers: LSD-SLAM: Large-scale direct monocular SLAM, Proc. European Conference on Computer Vision, pp. 834–849, 2014.
[5] K. Tateno, F. Tombari, I. Laina, and N. Navab: CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction, Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 6243–6252, 2017.
[6] M.Y. Ahmad and A.S. Mohan: Novel bridge-loop reader for positioning with HF RFID under sparse tag grid, IEEE Trans. Industrial Electronics, Vol. 61, No. 1, pp. 555–566, 2014.
[7] C. Chen, Y. Chen, Y. Han, H. Lai, and K.J.R. Liu: Achieving
centimeter-accuracy indoor localization on WiFi platforms: A frequency hopping approach, *IEEE Internet of Things Journal*, Vol. 4, No. 1, pp. 111–121, 2017.

[8] O. Pink: Visual map matching and localization using a global feature map, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008, CVPRW’08*, pp. 1–7, 2008.

[9] J. Takahashi: A concept and recent results of universal map: Cloud-based positioning infrastructure system, *Proc. Third International Workshop on Smart Sensing Systems, S2-2*, 2018.

[10] S. Ito, N. Kaneko, J. Takahashi, and K. Sumi: Global localization from a single image in known indoor environments, *7th International Conference on Informatics, Electronics and Vision*, pp. 70–75, 2018.

[11] D. Robertson and R. Cipolla: An image-based system for urban navigation, *Proc. British Machine Vision Conf.*, pp. 819–828, 2004.

[12] S. Achar, C.V. Jawahar, and K.M. Krishna: Large scale visual localization in urban environments, *Proc. IEEE International Conference on Robotics and Automation*, pp. 5642–5648, 2011.

[13] H. Taira, M. Okutomi, T. Sattler, and M. Cimpoi: InLoc: Indoor visual localization with dense matching and view synthesis, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7199–7209, 2018.

[14] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S.M. Seitz, and R. Szeliski: Building Rome in a day, *Comm. ACM, Vol. 54, No. 10*, pp. 105–112, 2011.

[15] Y. Li, N. Snavely, D.P. Huttenlocher, and P. Fua: Worldwide pose estimation using 3D point clouds, *Proc. European Conference on Computer Vision*, pp. 15–29, 2012.

[16] T. Sattler, B. Leibe, and L. Kobbelt: Efficient & effective prioritized matching for large-scale image-based localization, *IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 39, No. 9*, pp. 1744–1756, 2017.

[17] A. Bartoli and P. Sturm: Structure-from-motion using lines: Representation, triangulation and bundle adjustment, *Computer Vision and Image Understanding, Vol. 100, No. 3*, pp. 416–441, 2005.

[18] B. Micusik and H. Wildenauer: Structure from motion with line segments under relaxed endpoint constraints, *Proc. International Conference on 3D Vision*, pp. 13–19, 2014.

[19] B. Micusik and H. Wildenauer: Descriptor free visual indoor localization with line segments, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3165–3173, 2015.

[20] G.D. Tipaldi, D. Meyer-Delius, and W. Burgard: Lifelong localization in changing environments, *The International Journal of Robotics Research, Vol. 32, No. 14*, pp. 1662–1678, 2013.

[21] M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale: The gist of maps: Summarizing experience for lifelong localization, *Proc. IEEE International Conference on Robotics and Automation*, pp. 2767–2773, 2015.

[22] T. Krajinčik, J.P. Fentanes, M. Hanheide, and T. Duckett: Persistent localization and life-long mapping in changing environments using the frequency map enhancement, *Proc. IEEE/RSJ International Conference on Intelligent Robotics*, pp. 4558–4563, 2016.

[23] N. Shaik, T. Liebig, C. Kirsch, and H. Müller: Dynamic map update of non-static facility logistics environment with a multi-robot system, *Proc. German Conference on Artificial Intelligence*, pp. 249–261, 2017.

[24] P. Biber and T. Duckett: Experimental analysis of sample-based maps for long-term SLAM, *Int. J. Robotics Research, Vol. 28, No. 1*, pp. 20–33, 2009.

[25] K. Konolige and J. Bowman: Towards lifelong visual maps, *Proc. International Conference on Intelligent Robotics*, pp. 1156–1163, 2009.

[26] R.G. Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall: LSD: A line segment detector, *Image Processing On Line, Vol. 2*, pp. 35–55, 2012.