Improvement of Image Retrieval-based Visual Localization Using Structured Database

Ayari Akada
dept. of Mechanical Engineering
Kagoshima University
Kagoshima, Japan
k1221553@kadai.jp

Junji Takahashi
dept. of Mechanical Engineering
Graduate School of Science and Engineering
Kagoshima University
Kagoshima, Japan
takahashi@mech.kagoshima-u.ac.jp

Yu Yong
dept. of Mechanical Engineering
Graduate School of Science and Engineering
Kagoshima University
Kagoshima, Japan
yu@mech.kagoshima-u.ac.jp

***Abstract*—In this paper, we propose a method to improve image retrieve for visual localization by structuring the database. We are studying cloud-based positioning infrastructure system that we call Universal Map. It can reduce various cost as compared with the conventional technique. However, it takes time to estimate the position because the retrieval process is performed from a large amount of images in the database. To solve this problem, we reduce the retrieval time by structuring the database. We designed feature vector representing each image in the database and classified them using clustering method called K-means. We also made virtual sensing image and measured the Euclidean distance to each cluster in order to evaluate the classification results. As a result, the correct cluster was selected up to the third closest cluster. Therefore, we could reduce the retrieval time to 20% so far.**

***Index Terms*—Line segments, K-means clustering, Visual localization**

I. Introduction

Recently, with the development of technology, robots are gradually working around us. Thereby, the necessity of visual localization system, especially indoors, is increasing.

There are currently two main methods for visual localization of an autonomous mobile robot. One is Simultaneous Localization and Mapping (SLAM) [1] and the other is setting marks in the surrounding environment [2]. However, both of them have a problem that the cost for equipment and installation is high. Also, it takes a little long time to estimate because there are many calculations.

Then, as a new method, Universal Map (UMap): Cloud-based positioning infrastructure system was proposed by Takahashi [3]. This provides position information by collating images sent from an client such as cellphones with pre-prepared 3D CAD map data on a server. Calculation cost and installation cost on the client side can be greatly reduced because a large device is unnecessary. However, it searches from a large number of images in the database at the same time in matching on the server side, so there is a problem that processing takes time.

Therefore, in this paper, we propose a method to improve efficiency of UMap system by classifying database images and searching for each group.

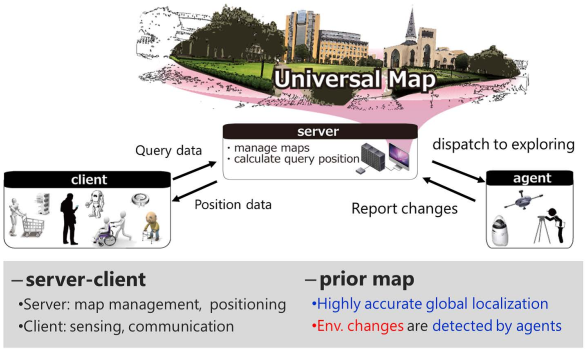
Universal Map
server
· manage maps
· calculate query position
Query data
dispatch to exploring
client
agent
Position data
Report changes
–server-client
•Server: map management, positioning
•Client: sensing, communication
–prior map
•Highly accurate global localization
•Env. changes are detected by agents

Fig. 1. A conceptual diagram of UMap taken from [3].

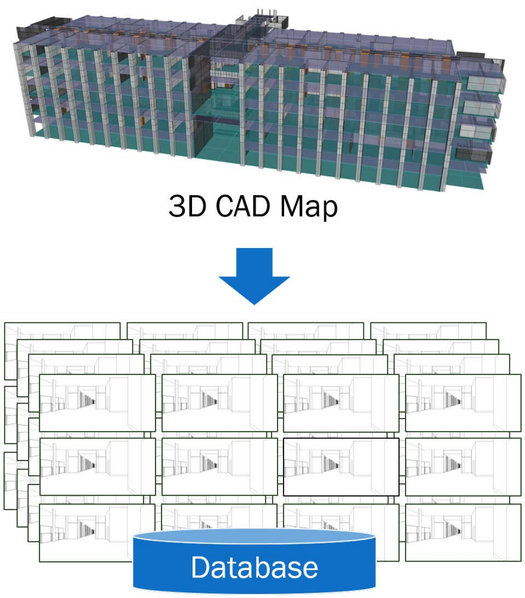

Fig. 2. 3D CAD map and database.

II. Universal Map

A conceptual diagram of UMap is shown in Fig. 1. UMap system consists of three parts: a client, a central server, and an agent. The 3D CAD map is prepared in advance in

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

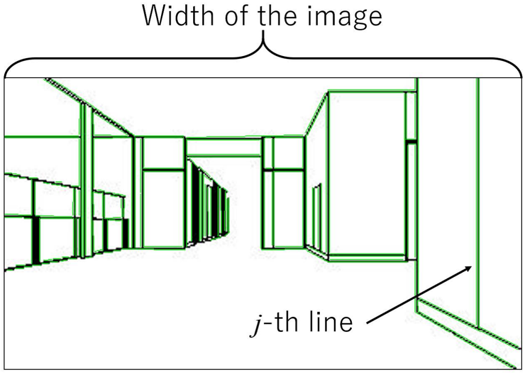

Fig. 3. An example of i-th database image detected by LSD.

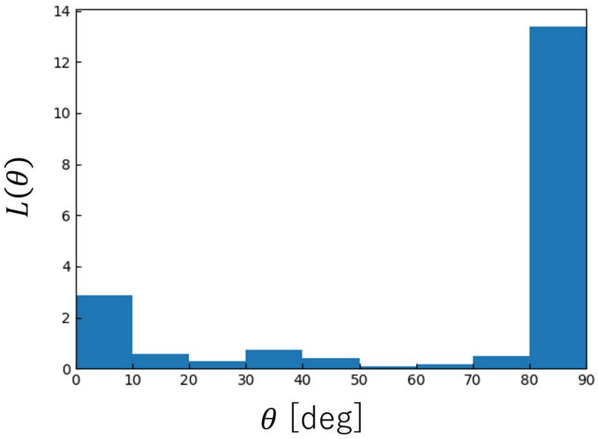

Fig. 4. The graph of the angle and the sum of relative lengths of i-th database image.

the server and 2D line segment database images are created from it by openGL(Fig. 2 [3]). The matching procedure of UMap is shown as follows. First, the client sends the sensing image data to the central server. Second, the server converts recieved image data to line segments image. Third, the logical conjunction between line segments image and all database images are calculated. Finally, a database image with the highest similarity is selected by pixel count. Database images have position information, so the server provides the position information to the client. The agent manages the database of the server and updates if there is a change in the environment.

However, UMap has problems such as large processing time because the amount of database image is very large. For example, When generating database images in 8 directions in 0.2[m] increments within the range of 20[m]×2[m]×1[m], the number of images is $40,000$. In addition, if the size of the database image is 320×180, it is $320 \times 180 = 57.6$[KB] per image(BMP) and it is $40,000 \times 57.6$[KB] $= 2.3$[GB] in total. In this research, we focus on processing time and make it more efficient. We also consider reducing the amount of images handled during visual localization.

III. Feature vectors representing image

A. Line Segments

In the image retrieval process, line segments images are used. The images in the database are generated from 3D CAD by openGL functions and are originally line segments images. The sensing image from the client is converted to line segments image by Line Segment Detector (LSD) [8]. An example is shown in Fig. 3.

B. Feature vectors

First of all, we considered using the histogram of line length and number of lines. However, the number of vectors is large and only the number of each line is evaluated, so we thought it was not suitable for representing images. Then, we considered to evaluate the length of the line itself. On the other hand, we thought that the angle of lines include information of landscape and the number of vectors can be reduced by using angles. Additionally, there is a study that use angular information of scanned lines as features of images [9], so we came up with a feature vector combining angle and length information.

We use the angle of each line and the sum of the length obtained by LSD. Also, we define the relative length so that the same value can be extracted for the same image of different sizes. The relative length L_j is the length of j-th line in the i-th database image divided by the width of the image and is given as follows,

$$L_j = \frac{Length\ of\ j\text{-}th\ line}{Width\ of\ the\ image}. \tag{1}$$

The angle of the line is expressed in the range of 0 [deg] to 90 [deg] and relative lengths having the same angle are added up. The angle of j-th line is θ_j and the sum of relative lengths at $\theta(=\{0, 10, ..., 80[\text{deg}]\})$ is $L(\theta)$. These are expressed in 10 [deg] increments, so $L(\theta)$ is expressed as follows,

$$L(\theta) = \sum_{j=1}^{J} L_j \times \alpha, \tag{2}$$

$$\alpha = \begin{cases} 1(\theta \leqq \theta_j < \theta + 10) \\ 0(others). \end{cases}$$

Using $L(\theta)$, we define the feature vector $\boldsymbol{v}_i$ for representing i-th image as follows,

$$\boldsymbol{v}_i = \{L(0), L(10), ..., L(80)\}. \tag{3}$$

The graph of the angle and the sum of relative lengths of i-th image is shown in Fig. 4.

IV. Clustering method

K-means [7] is often used as a method of classifying images. K-means is one of the clustering methods and it is possible to classify a set of data into K pieces of clusters given without knowledge prepared. It can also be implemented relatively easily, so we used it to classify images.

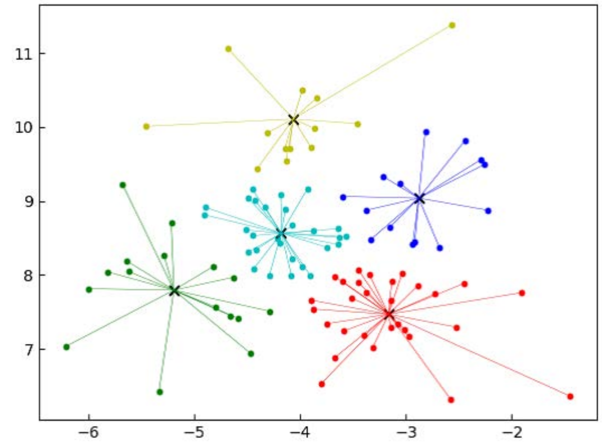

Fig. 5. An example of K-means.

The processing procedure is follows. $X(=\{x_1, x_2, ..., x_N\})$ is the set of N pieces of data and $\mu_k(k = 1, ..., K)$ is the centroid of the cluster. First, the position of each centroid is randomly determined by the number of K. Next, each data is assigned the label of the closest centroid cluster. Then, calculate the centroid for each label and the evaluation function is given as follows,

$$E = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk}(x_n - \mu_k)^2, \tag{4}$$

$$r_{nk} = \begin{cases} 1(x_n \in k) \\ 0(x_n \notin k). \end{cases}$$

Calculate this procedure iteratively, and if E becomes the minimum, classification will end. The convergence position of the centroid depends on the initial value. An example of K-means is shown in Fig. 5.

V. Experiment

We conducted two experiments. First, we investigated the optimum number of divisions K. Next, we divided the database images into K clusters and examined how much cost can be reduced. An outline of these experiment is drawn in Fig. 6. The procedure is shown as follows.

1) Make database images and classify it into K clusters.
2) Create virtual sensing image.
3) Compare the feature vectors of the virtual sensing image and the centroid of the cluster, and measure the Euclidean distance.

In this experiment, we use the 3D CAD map of O-building in Aoyamagakuin University. The total area of the all floor of the building is $11,280[\mathrm{m}^2]$. The top view of the 5th floor of the O-building and the range of the generation position of the database images are shown in Fig. 7.

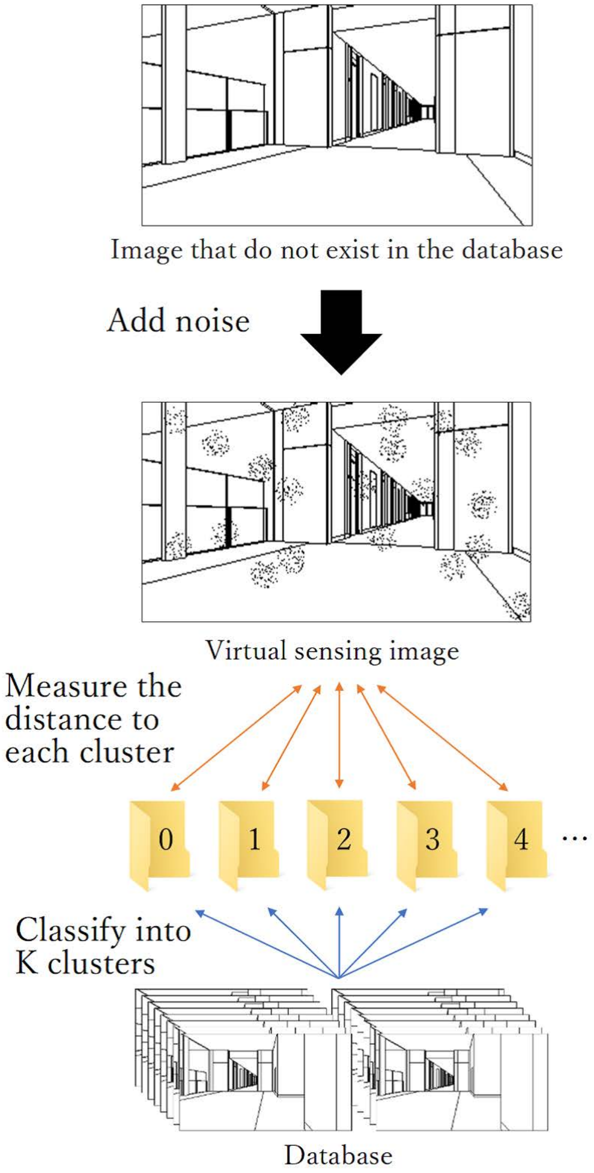
Image that do not exist in the database
Add noise
Virtual sensing image
Measure the distance to each cluster
0
1
2
3
4
…
Classify into K clusters
Database

Fig. 6. Outline of the experiment.

A. Overview of clustering database

We created 29,600 database images with a grid width of 0.2 [m] and 16 directions. It's direction is shown in Fig. 10.

Line segments of each image are detected by LSD, and they are represented by a feature vectors from a graph of the angle and the sum of relative lengths. Using this vectors, database images were classified into K clusters by K-means. The maximum value of each component of the feature vectors is around 10, so we design the algorithm to stop the iteration when the average of the moving distance of the centroid of each cluster becomes 0.01 or less. At this time, the cluster which each database images belongs to and the centroid of the cluster are recorded. An example of images classified into each cluster is shown in Fig. 8.

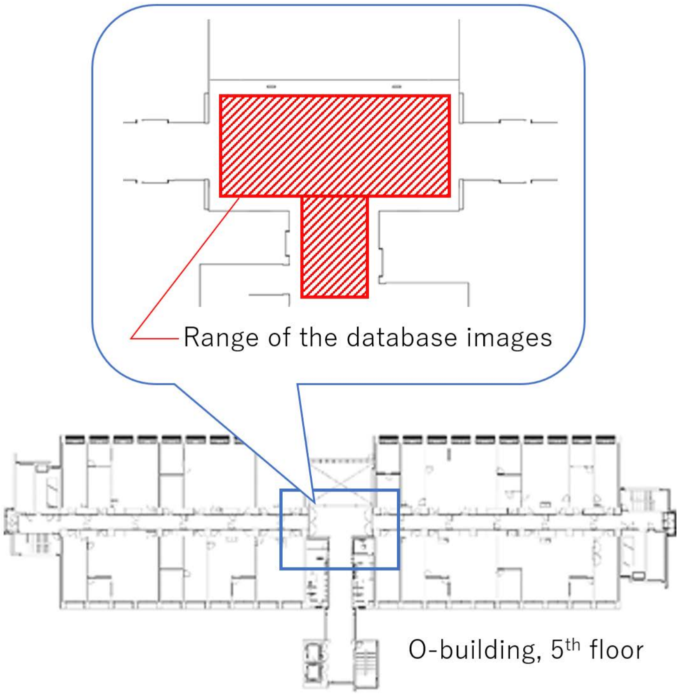

Fig. 7. The 5th floor of the O-building in Aoyamagakuin University and the range of the database images position.

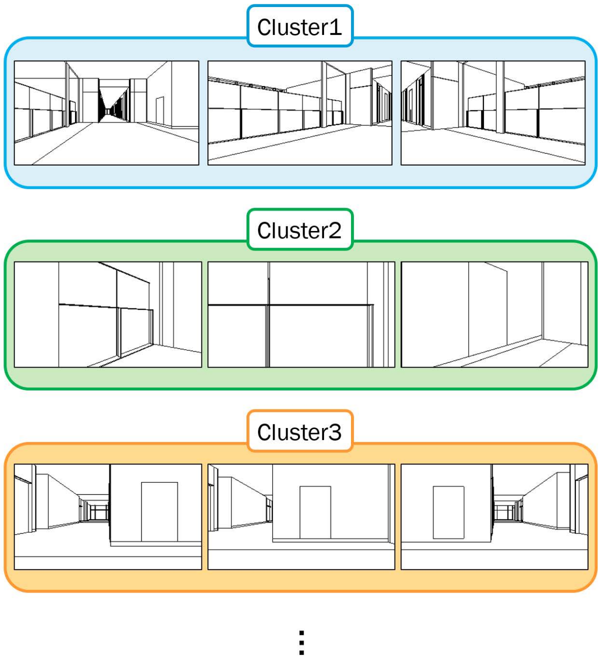

Fig. 8. An example of classified images into each cluster.

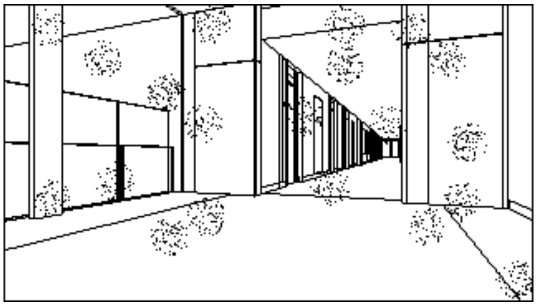

Fig. 9. Virtual sensing image.

B. Virtual sensing image

We prepared an image that do not exist in the database and added black dot noise to it with a paint tool. This is a virtual sensing image shown in Fig. 9. we prepared 40 number of virtual sensing image. The geometrical relation between the virtual sensing image and database images are described in Fig. 10.

C. The optimum division number K

As described in IV, in the K-means method, the division number K has to be determined in advance. Therefore, we devised a separation metrics based criteria and our original cost function based criteria, and we used them to find the optimum number of divisions K.

We divided $10,000$ database images by changing the number of $K(= 2 \sim 10, 15, 20)$. We measured the Euclidean distances between the feature vectors of the virtual sensing data and each cluster's centroid in the case of dividing by each K, and we recorded which order cluster is the correct cluster.

1) Separation metrics based criteria: The separation metrics [10] is a threshold at which separation of classes obtained by dividing data is maximized. The separation metrics is the variance between classes divided by the variance within the class. The variance within the class represents the average spread within one class and the variance between classes represents spread of classes. The total number of class i data is n_i and the average value is μ_i, the variance is σ_i. The variance within the class σ_W^2 and the variance between classes σ_B^2 and the separation metrics J are given as follows,

$$\sigma_W^2 = \sum_{i=1}^{I} \frac{n_i}{n_1 + ... + n_I} \sigma_i^2 \tag{5}$$

$$\sigma_B^2 = \sum_{i=1}^{I} \frac{n_i(\mu_i - \mu_{all})^2}{n_1 + ... + n_I} \tag{6}$$

$$\mu_{all} = \sum_{i=1}^{I} \frac{n_i \mu_i}{n_i + ... + n_I}$$

$$J = \frac{\sigma_B^2}{\sigma_W^2}. \tag{7}$$

The result of examining the separation metrics by changing the number of divisions K is shown by orange line in Fig. 11.

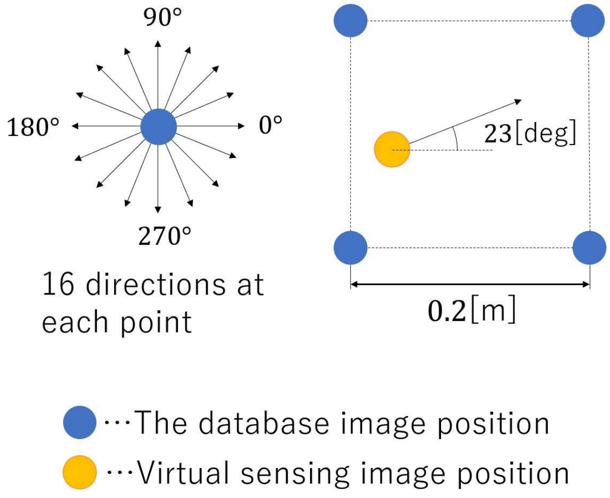
90°
180°
0°
270°
16 directions at each point
23[deg]
0.2[m]
●…The database image position
●…Virtual sensing image position

Fig. 10. Position and direction of database image and virtual sensing image.

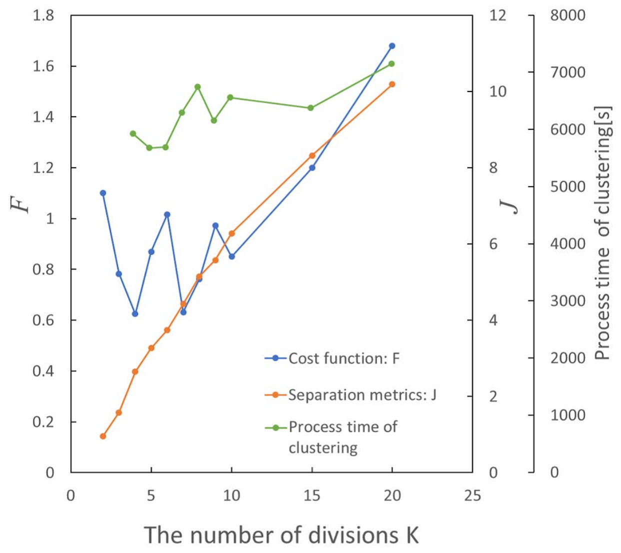

Fig. 11. The separation metrics and the cost function graph.

According to this, it is clear that the separation metrics increases as the number of divisions increases. Therefore, a threshold value of the number of divisions K could not be obtained with the method using the separation metrics, so we created a new function.

2) Cost function based criteria: We created a function that considers the amount of data and the cost of searching the correct cluster when changing the number of divisions. The number of divisions is K and the probability that the m-th closest cluster is correct is a_m, and the cost function F is defined as follows,

$$F = \{(a_1 \times 1) + ... + (a_m \times m)\} \times \frac{m}{K}. \quad (8)$$

K with the smallest cost function value is the optimum division number. A graph of the value of the cost function for each division number is shown by blue line in Fig. 11. Therefore, it was found that the optimal division number for $10,000$ database images is 4.

TABLE I
RESULT OF THE EXPERIMENT

Virtual sensing image number	Correct cluster	Close cluster 1st	2nd	3rd
1	9	9		
2	10	9	10	
3	3	3		
4	10	10		
5	6	11	6	
6	3	3		
7	7	7		
8	5	5		
9	9	9		
10	11	11		
11	3	3		
12	5	0	5	
13	10	10		
14	10	10		
15	11	6	11	
16	7	7		
17	7	7		
18	9	9		
19	0	0		
20	4	4		
21	3	3		
22	8	5	11	8
23	3	3		
24	9	9		
25	3	3		
26	8	8		
27	6	6		
28	9	9		
29	3	3		
30	8	11	3	8
31	4	4		
32	7	9	7	
33	3	3		
34	9	5	9	
35	8	11	6	8
36	11	6	11	
37	7	9	7	
38	9	9		
39	7	9	7	
40	10	10		
Rate of correct cluster up to 1st				70%
Rate of correct cluster up to 2nd				92.5%
Rate of correct cluster up to 3rd				100%

D. Evaluation of cost reduction

We investigated how much data volume and cost can be reduced using $29,600$ database images. As described above, it is optimal to divide $10,000$ database images into $K = 4$. The number of database images we used is about three times, so we divided $29,600$ images into $K = 4 \times 3 = 12$. Like the previous experiment, we measured the Euclidean distance between the virtual sensing image and each centroid using feature vectors. The result of the experiment are shown in Table I. The probability that the closest cluster is the correct cluster is 70% and the probability that the correct cluster was selected up to the second closest cluster was 92.5%, the probability that the correct cluster was selected up to the third closest cluster was 100% Therefore, the number of database images to be retrieved can be reduced to $3 \div 12 = 1/4$ by classifying them using the feature vectors. In this case, database images are about 57.6[KB] per image, so the total is $29,600 \times 57.6$[KB]$=$ 1.7[GB], but we can reduce the amount of database images handled during image retrieval to 1.7[GB]$\div 4 = 0.426$[MB]$=$ 426[MB]. Also, we could reduce the time taken to read the database images in the program from $38,051$[ms] to $8,832$[ms].

VI. Related works

In recent years, various visual localization techniques have been proposed. Especially, the SLAM [1] approaches are representative methods of visual localization. The SLAM performs visual localization and map generation at the same time by using onboard sensors on a mobile robot. However, there is a problem that the device and the computation cost are large. In this regard, our proposing UMap does not need to prepare large sensor device on the robot. Additionally, since the calculation for localization is done on the server side, the computation cost on the client side is small.

There is a method of performing visual localization by comparing photograph taken by the client with photographs of database [5]. However, this method needs so many photographs as a database. On the other hand, our proposing UMap is different in that it does not need to collect pictures because UMap generates database images from 3D CAD. Also, the size of the database is relatively small because UMap uses 2D line segment images.

There is a research to create a database for visual localization using Structure from Motion(SfM) [4]. SfM is a technique of restoring 3D shape of the subject and the relative position of the camera from a large number of photographs with different viewpoints. This is different from UMap in how to create database images, and UMap does not need to prepare photos.

There is a method of setting a mark in the environment where a robot with a sensor moves [2], but the installation cost of the mark is large. UMap is visual localization system that does not set anything on the environment side and does not need any special device.

There are also various clustering methods for classifying data with similar features and several studies have applied it to image classification. There is research to classify using image color and texture features [6] and images are classified by K-means method using histograms of color and texture features. In this paper, we propose a method of classifying database images by K-means using feature vectors obtained from line segments of images, not histograms.

VII. Conclusion

In this paper, we improved image retrieval based visual localization using structured database. Line segments in the image were detected by LSD and the angle of the line and the sum of relative lengths was used as a feature vectors representing the image. We used this feature vectors to classify database images with K-means and conducted experiments to find out the optimum division number K and examined how much cost can be reduced. As a result, we found the optimum number of K by using the cost function and the correct answer was selected up to the third closest cluster. Therefore, we could reduce the retrieval cost to 20% so far.

In the future, we conduct experiments in different buildings and verify validity of feature vectors. Also, we will classify database finer and make it hierarchical so that we can reduce the retrieval time more. In addition, we are considering a method of storing classified database images in a lot of servers.

Acknowledgment

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

References

[1] Y. Zhao, T. Wang, X. Deng, W. Qin and X. Zhang, “Improved Particle Swarm Optimization for Multi-robot SLAM,” in proceeding of the 27th IEEE International Symposium on Robot and Human Interactive Communication, Aug 2018.

[2] R. D’Andrea, “A revolution in the warehouse: A retrospective on Kiva systems and the grand challenges ahead,” IEEE Transactions on Automation Science and Engineering vol. 9, Oct 2012, pp. 638–639.

[3] J. Takahashi, “A concept and recent result of UniversalMap: Cloud-based positioning infrastructure system,” in proceeding of the third International Workshop on Smart Sensing Systems, June 2018.

[4] T. Sattler, B. Leibe and L. Kobbelt, ”Fast image-based localization using direct 2D-to-3D matching,” 2011 International Conference on Computer Vision, Barcelona, 2011, pp. 667-674.

[5] F. Fraundorfer, C. Engels and D. Nister, ”Topological mapping, localization and navigation using image collections,” 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, 2007, pp. 3872-3877.

[6] C. Bai, J. Zhang, Z. Liu and W. Zhao, “K-means based histogram using multiresolution feature vectors for color texture database retrieval,” Multimedia Tools and Applications vol. 74 issue 4, Feb 2015, pp. 1469–1488.

[7] C. Yin and S. Zhang, “Parallel implementing improved k-means applied for image retrieval and anomaly detection,” Multimedia Tools and Applications vol. 76 issue 16, Aug 2017, pp. 16911–16927.

[8] R. G. Gioi, J. Jakubowicz, J-M. Morel and G. Randall, “LSD: a Line Segment Detector,” Image Processing On Line vol. 2, 2012, pp. 35–55.

[9] B. Micusik and H. Wildenauer, “Descriptor Free Visual Indoor Localization with Line Segments,” 2015 IEEE Conference on Computer Vision and Pattern Recognition, June 2015, pp. 3165–3173.

[10] L. M. Blumberg, ”Metrics of separation performance in chromatography. Part 1. Definitions and application to static analyses,” Journal of Chromatography A vol.1218, issue32, 12 Aug 2011, pp. 5375–5385.