Outdoor Place Recognition in Urban Environments using Straight Lines

Jin Han Lee, Sehyung Lee, Guoxuan Zhang, Jongwoo Lim, and Il Hong Suh

Abstract— **In this paper, we propose a visual place recognition algorithm which uses only straight line features in challenging outdoor environments. Compared to point features used in most existing place recognition methods, line features are easily found in man-made environments and more robust to environmental changes such as illumination, viewing direction, or occlusion. Candidate matches are found using a vocabulary tree and their geometric consistency is verified by a motion estimation algorithm using line segments. The proposed algorithm operates in real-time, and it is tested with a challenging real-world dataset with more than 10,000 database images acquired in urban driving scenarios.**

I. Introduction

Place recognition algorithms are used in several applications such as simultaneous localization and mapping (SLAM) and autonomous robot navigation. In robotics, the geometry-based localization process often exhibit scalability problems because, during pose estimation, small errors accumulate and eventually it becomes too large to be corrected via geometric reasoning. Therefore, many recent robotics applications utilize appearance-based techniques for the localizations [1]–[4].

Currently, most visual place recognition systems use point features, such as the scale-invariant feature transform (SIFT) [5] or speeded-up robust features (SURF) [6]. In SLAM researches, however, there has been some approaches using lines as landmarks [7]–[10], because lines can effectively convey structural information with fewer number of them, as a line spans over a one-dimensional space, rather than a single point in a space (see Figure 1). However, they have not been widely adopted because tracking lines are harder than tracking points and recognizing them is difficult due to the lack of reliable feature descriptors.

Previously, we proposed a system [11] that reliably recognizes places in structured indoor environments with only line features, and we aim to extend the method to outdoors in this work. In [11], we used a Bayesian filtering framework to reduce the influence of the noisy responses from the vocabulary tree. However, the approach has a limit that all scenes in both of the query and the database need to be in sequential orders because the Bayesian filtering works under that assumption. In this work, however, we utilize a geometric verification algorithm instead of the filtering, and the scenes do not need to be in sequential orders. To the best of our knowledge, there has been no visual place recognition system using only line features in outdoor environments. We utilize the vocabulary tree presented by Nister et al. [12] that we train for finding matching hypotheses. Then, for geometric verification of the hypotheses, we adopt an idea from Zhang [13] to estimate a relative motion between two scenes with line segments. We show that the retrieval performance of a vocabulary tree built with line descriptors works better than a tree built with state-of-the-art point descriptors in a structured outdoor environment, and the potential of using line descriptors in practical visual place recognition systems. We utilize the mean standard-deviation line descriptor (MSLD) proposed by Wang et al. [14] as a descriptor for line segments.

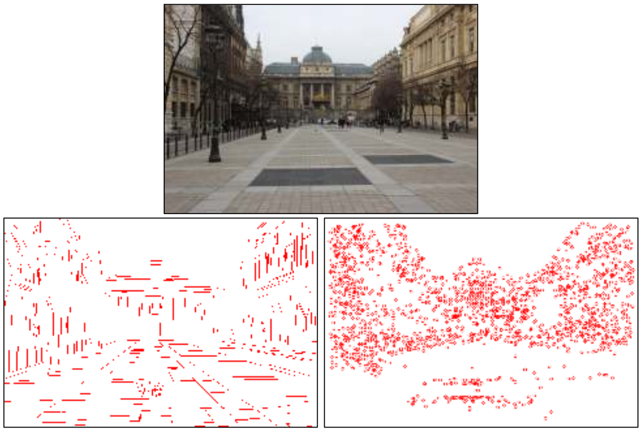


Fig. 1. An outdoor scene, and two figures containing 498 line features and 1758 SURF features, respectively, extracted from the scene. Straight lines are preferred over points, because they represent structural information more effectively.

The main contributions of this paper are as follows:

- A geometric verification algorithm using line segments
- A real-time implementation and experimental validation of a place recognition algorithm that uses only line features under challenging conditions

The remainder of this paper is organized as follows. Section II describes algorithms for finding matching hypotheses using a vocabulary tree, and presents an experimental evaluation of the retrieval performance of the tree trained with line descriptors, by comparing it with another tree trained with SIFT. Section III presents a motion estimation algorithm used to verify candidate matches. In Section IV, we provide results obtained from experiments conducted in

Jin Han Lee and Sehyng Lee are with the Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea. {jhlee, shl}@incorl.hanyang.ac.kr.

Guoxuan Zhang is with the Drexel Auytonomous Systems Lab, Drexel University, Philadelphia, USA. imzgx@hanyang.ac.kr.

Jongwoo Lim and Il Hong Suh are with the Division of Computer Science and Engineering, College of Engineering, Hanyang University, Seoul, Korea. {jlim, ihsuh}@hanyang.ac.kr. All correspondences should be addressed to Il Hong Suh.

urban driving environments containing several environmental changes. This paper concludes in Section V.

II. Scene Representation with Line Segments

A. Line Extraction and Description

In order to extract line segments, we devised a simple but reliable extractor inspired from [15]. Given an image, Canny edges are detected first and the system extracts line segments as follows: At an edge pixel the extractor connects a straight line with a neighboring one, and continues fitting lines and extending to the next edge pixel until it satisfies co-linearity with the current line segment. If the extension meets a high curvature, the extractor returns the current segment only if it is longer than 20 pixels, and repeats the same steps until all the edge pixels are consumed. Then with the segments, the system incrementally merges two segments with length weight if they are overlapped or closely located and the difference of orientations is sufficiently small.

Descriptor vectors for the segments are generated using MSLD [14]. For each segment, the MSLD first identifies the perpendicular direction $\mathbf{d}_\perp$ with its average gradient direction, and parallel direction $\mathbf{d}_\|$ rotated 90 degrees from $\mathbf{d}_\perp$ in clockwise. For every pixel on the segment, it sets c subregions each with a size of $r \times r$ along to the $\mathbf{d}_\perp$ in a non-overlapping manner. If a line segment consists of l pixels, it results $c \times l$ subregions on the segment. In this work we use the same settings $c = 9$, $r = 5$ as in [14]. In each subregion, accumulating distributed gradients along the direction $\mathbf{d}_\perp$, $\mathbf{d}_\|$ and their opposite directions results a histogram with four bins. With the mean and standard deviation of the histograms calculated along the $\mathbf{d}_\|$ results $(4+4) \times 9 = 72$ dimensional vectors. This statistical representation allows robust matching between two line segments with noisy locations of end points.

B. Vocabulary Tree

The visual bag-of-words approach maps an arbitrary feature to a visual word using a pre-built dictionary, and represents the scene with the set of words to recognize it. In other words, to get a dictionary, it divides the feature space by clustering given huge number of training features. Then for each of arbitrary features, it assigns one of the cluster index to the feature to efficiently represent scenes. The vocabulary tree [12] is one of the most popular algorithms among the visual bag-of-words family. It hierarchically divides the feature space to offer more efficient and effective way in both of training and querying phases, and it enables online database insertion and querying when it is utilized with an inverted file mechanism.

We extracted eight million MSLD descriptors from eight tourism videos of historical buildings in Europe, and used them as the training set to build a vocabulary tree. Then, we performed hierarchical k-means clustering of branching factor $k = 50$, number of levels $l = 3$, with the training set resulting a tree with 127551 nodes. Following the analysis of the authors of [12], we use only 125000 leaf nodes in this work. In Section II-C, we experimentally evaluate the retrieval performance of the tree.

In the phase of database construction, every descriptor vector in the scenes inserts the ID of the image to the corresponding leaf node. Similarly, when querying a scene, also every descriptor vector in the query image traverses through the tree to reach a leaf node, then images of the ID listed in the node represent potential candidate matches and receive votes. In this voting scheme, we use the normalized difference with term frequency-inverted document frequency (TF-IDF) weighting [16] in the L_1-norm [12].

We define the query $\mathbf{q}$ and the database $\mathbf{d}$ vectors as follows:

$$q_k = n_k w_k, \tag{1}$$

$$d_k = m_k w_k. \tag{2}$$

Here,

$$n_k = \frac{\text{number of word } k}{\text{number of total words in query scene}} \tag{3}$$

is the term frequency of the word k in the query image, and

$$m_k = \frac{\text{number of word } k}{\text{number of total words in the database scene}} \tag{4}$$

is the term frequency of word k in a database image. Moreover, w_k is the inverted-document frequency given by

$$w_k = \ln \frac{N}{N_k}, \tag{5}$$

where N is the total number of database images, and N_k is the number of database images containing the word k. Then, if the word k is observed in the query scene, the score assigned to the database image i is given by

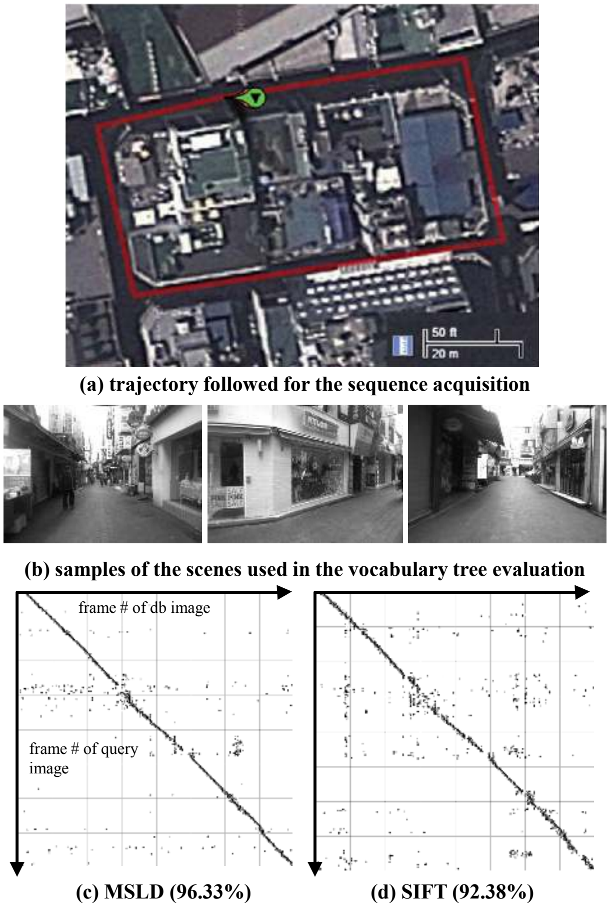
$$s_i = 2 + \sum_{k | q_k \neq 0, d_k \neq 0} \big(|q_k - d_k| - q_k - d_k \big). \tag{6}$$

In this scoring scheme, the words with high “term frequency” (*i.e.,* they frequently appear in an image) receive higher scores. Meanwhile, words with high “inverted-document frequency” (*i.e.,* they also frequently appear in other images) are penalized.

C. Evaluation of the Vocabulary Tree in an Outdoor Environment

In order to verify the performance of the vocabulary tree of MSLD line descriptors, we performed experimental comparison with another vocabulary tree trained with SIFT in the identical environment. We used a standard implementation of the SIFT from [19]. For this experiment, we acquired images with a robot-equipped camera in Myung-dong, Seoul, while travelling a 235 meter-long loop twice. In building the vocabulary tree with the SIFT features, we used the same settings that are used to build the vocabulary tree with the MSLD features (*i.e.*, the same eight videos, eight million SIFT features, $k = 50$, and $l = 3$).

In the sequence, the 721 scenes acquired from the first travel are used as a database, and the 682 scenes from the second travel are used as queries. In Figure 2, (a) shows the



(a) trajectory followed for the sequence acquisition

(b) samples of the scenes used in the vocabulary tree evaluation

(c) MSLD (96.33%)

(d) SIFT (92.38%)

Fig. 2. Experimental evaluation of the trees

trajectory followed for the sequence acquisition, (b) shows some examples of the scenes, and (b) and (c) are the results of the evaluation. The top five scored returns from the query are represented by points darkened according to their scores. Since it is difficult to obtain the actual trajectories of the robot, we tried to maintain the velocity of the robot to be constant, and to follow the almost same trajectories. Then, we can approximate its retrieval accuracy by following two steps: First, we adjust the scale of the vertical axis to be equal to the scale of the horizontal axis. Then, with an assumption that correct matches should be on the diagonal line, we consider a query is successful if at least one of the top five returns is not farther than ten frames from the diagonal line. In the case of MSLD, the number of successful returns was counted as 657, and it was 630 in the case of the SIFT. As shown in Figure 2.(b) and (c), we observe that the vocabulary tree built with the SIFT descriptors shows a little more spread distribution of the points along the diagonal line than the case of the MSLDs. Because the tested area was very structured, it is more reasonable to attribute this result to the experimented environment and not the performances of the SIFT or the MSLD. The results are given in Table I.

TABLE I

EXPERIMENTAL EVALUATIONS OF THE VOCABULARY TREES

	MSLD	SIFT
edge threshold	.	500
line length	20	.
# database	721	
# query	682	
# success return	657	630
% success	96.33%	92.38%
# feature	148,513	625,338

From top to bottom, each row indicates **edge threshold**: an edge threshold used in keypoint extraction for SIFT features, **line length**: threshold of minimum length in line segment extraction, **# database**: number of scenes stored in the database, **# query**: number of query scenes, **# success return**: number of scenes counted as successful return, **% success**: percentage of the successful returns to the database, **# feature**: total number of features used in this evaluation

III. Motion Estimation using Line Segments

More scenes in the database, higher ambiguity in the best hypothesis selection is unavoidable if the vocabulary tree is used alone for finding the best match because it does not take into account any geometric information of the features. Therefore, geometric verification of the feature configurations is employed to improve the retrieval accuracy. In most visual place recognition systems that use point features, multiple-view geometry such as epipolar constraint is used to find only consistent matches using five point algorithm [17] or eight point algorithm [18]. Under an assumption that corresponding feature points in two views come from the same rigid 3D scene, it verifies them with the geometry of features. In case of line matches, it is well known that a relative motion between two views cannot be determined from any number of line matches [7]. However, there has been some algorithms which computes the relative motion by maximizing the overlap of the matched line segments. In this work, we use a similar approach as Zhang [13] to estimate motion between a query and a hypothesis image using line

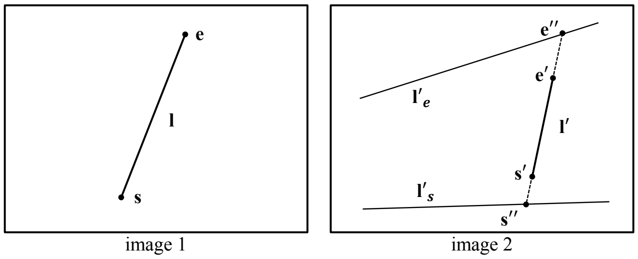


Fig. 3. Two line segments in correspondence. In this case, the overlap length is defined as the length of the line segment $\mathbf{l}'$.

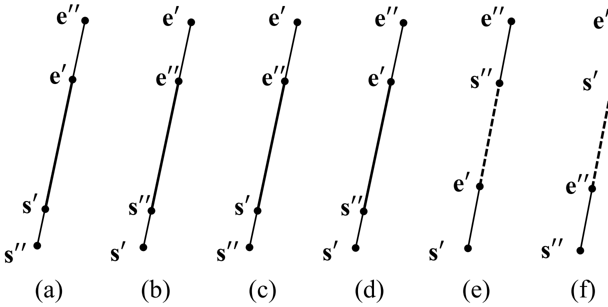


Fig. 4. All possible cases of the two line segments. In each case of (a)-(d), the overlap length is defined as the length of the thick line. In (e) and (f), the overlap lengths are defined as the gap represented by the dotted lines.

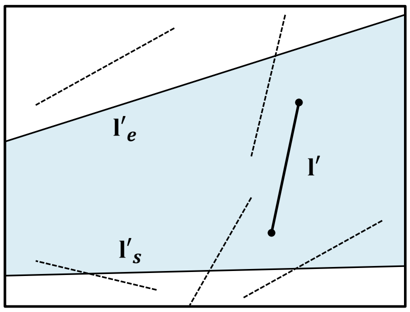


Fig. 5. The epipolar geometry reduces the search region for a line segment $\mathbf{l}$. The matching line segment $\mathbf{l}'$ should have at least one of the its endpoints in the region.

segments. However, the following different techniques are adopted for our objective.

- Instead of using the downhill simplex method for optimization in [13], we utilize a nonlinear least square method to guarantee real-time performance.
- In the design of the cost function, we use a much simpler cost function and utilize a robust loss function to reduce the effects of outliers in feature correspondences.

A. Maximizing Overlap Length of the Matched Segments under Epipolar Geometry

Figure 3 shows the definition of the overlap length of the matched segments. We denote a line segment in the first image as $\mathbf{l}$ and a corresponding line segment in the second image as $\mathbf{l}'$. The rotation matrix and translation vector between two images are denoted as $\mathbf{R}$ and $\mathbf{t}$, respectively. The essential matrix $\mathbf{E}$ becomes $[\mathbf{t}]_\times \mathbf{R}$ where $[\mathbf{t}]_\times$ denotes the skew symmetric matrix of $\mathbf{t}$. The epipolar line $\mathbf{l}'_\mathbf{p}$ in the second image of a point $\mathbf{p}$ in the first image can be written as

$$\mathbf{l}'_\mathbf{p} = \mathbf{E}\tilde{\mathbf{p}}, \tag{7}$$

where $\tilde{\mathbf{p}}$ is the homogeneous coordinate of the point $\mathbf{p}$. For the epipolar lines $\mathbf{l}'_\mathbf{e}$ and $\mathbf{l}'_\mathbf{s}$ of the two end points $\mathbf{e}$ and $\mathbf{s}$ of $\mathbf{l}$, we can then compute their intersections with the matched line $\mathbf{l}'$ as $\tilde{\mathbf{e}}'' = \mathbf{l}' \times \mathbf{l}'_\mathbf{e}$, $\tilde{\mathbf{s}}'' = \mathbf{l}' \times \mathbf{l}'_\mathbf{s}$, respectively. Because the line segments are oriented, and if the motions are relatively small, the possible combinations of the two segments can be considered as shown in Figure 4. We denote the overlap length in the second image as L' and it can be calculated simply by the following equation.

$$L' = \frac{1}{2}\left(\|\mathbf{v}_{\mathbf{e}''\mathbf{s}''}\| + \|\mathbf{v}_{\mathbf{e}'\mathbf{s}'}\| - \|\mathbf{v}_{\mathbf{e}''\mathbf{e}'}\| - \|\mathbf{v}_{\mathbf{s}''\mathbf{s}'}\|\right), \tag{8}$$

where $\mathbf{v}_\mathbf{ab}$ represents a vector between points $\mathbf{a}$ and $\mathbf{b}$. The overlap length L' is calculated only if $(\mathbf{e}' - \mathbf{s}') \cdot (\mathbf{e}'' - \mathbf{s}'')$.

We should consider a symmetric role of the both images. Therefore, we denote the overlap length L in the first image and calculate it in the same way. Moreover, the overlap lengths L'_i, L_i are divided by l_i, l'_i, respectively, to remove the influence of the length of the line segments, where l_i and l'_i denote the lengths of the line segments $\mathbf{l}_i$ and $\mathbf{l}'_i$, respectively.

Then, if a sufficient number of correspondences are given, by maximizing the overlap lengths defined by the whole matched segments the relative motion between the two views can be determined [13]. Finally, the motion estimation problem can be defined as minimizing following cost function for all i-th correspondences.

$$\sum_i \left((1 - L_i/l_i)^2 + ((1 - L'_i/l'_i)^2\right). \tag{9}$$

B. Optimization using a Nonlinear Least Square

For the optimization, we have implemented the Levenberg-Marqardt (LM) iteration method to achieve real-time performance. The LM is a widely used nonlinear least square method which shows good results by augmenting its normal equation so that transitions between Gauss-Newton and gradient methods occur according to its convergence. In order to reduce the influence of outliers in feature correspondences, we adopt the Cauchy loss function given by

$$\rho(c) = s^2 \log(1 + c^2/s^2), \tag{10}$$

where c is the cost and s is some constant. This function approximates c^2 for small values of c, and the s determines the range of the approximation. In this work, we empirically set the s to 0.3. The resulted cost function is as follows.

$$C = \sum_i \rho\left(\left((1 - L_i/l_i)^2 + ((1 - L'_i/l'_i)^2\right)\right). \tag{11}$$

Since the problem is nonlinear, initial guesses are important to obtain an acceptable solution. Similar to [13], we use icosahedrons to get uniformly distributed initial samples. For translation vector $\mathbf{t} \in \mathbb{R}^3$, we get 40 samples from a hemisphere of a tessellated icosahedron because if $\mathbf{t}$ is a solution, so is $-\mathbf{t}$. Since the scale of the translation $\mathbf{t}$ is inherently unrecoverable, we assume the $\mathbf{t}$ of unit length and use it in the spherical coordinate system in the optimization. Therefore, the $\mathbf{t}$ would be (ϕ, θ) in $\mathbb{R}^2$. For the rotation vector $\mathbf{r} \in \mathbb{R}^3$, we also sample 20 unit vectors from the faces of the original icosahedron (i.e. not tessellated). Since the angle-axis representation of the $\mathbf{r}$ has its norm as the rotation angle, we multiply each sample with $\frac{\pi}{6}$, $\frac{\pi}{3}$, resulting 40 samples for

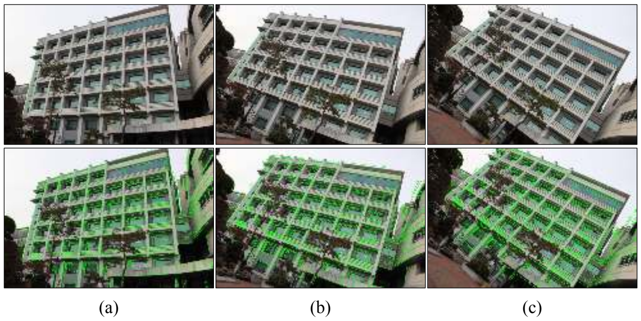


Fig. 6. Examples of warping of line segments. With an assumption of infinite depth of endpoints, the line segments in (a) are warped onto (b) and (c) with the relative motions estimated by the proposed algorithm.

$\mathbf{r}$. Adding a zero vector to the set of rotation samples results total 40×41=1640 samples. With those initial samples, the system calculates initial costs using the Equation (11). Then, only 10 samples which yield the smallest cost are used to carry out the optimization process independently. The final $\mathbf{r}$ and $\mathbf{t}$ resulting the minimum cost are accepted as the motion between the two scenes.

C. Geometric Verification of Two Scenes

As shown in Figure 5, the epipolar geometry reduces the search space for line segments. Furthermore, if we assume long distances of the 3D line segments in the world from the camera, we can warp their imaged segments from one image to another by treating their endpoints as in infinite depths. Then, the angle difference between the warped segment and the matching segment should be small. Therefore, the system searches line segments which holds those two constraints, and returns the matches if the distance of the two descriptor vectors of the segments are closer than a given threshold, η_d, and its nearest neighbor distance ratio is smaller than a threshold, η_r. Figure 6 shows two examples of the warping of the line segments with our motion estimation implementation. The images in column Figure 6. (a) show the reference image and extracted line segments, and the columns Figure 6. (b) and (c) show the target images and warped line segments from (a).

The hypothesis with the maximum number of the matches or with the minimum cost of the matches can be chosen as the recognized scene. We tested each scheme, and it returned some false positives which were not generated in the case of using the other scheme. Therefore, we define a score of the hypothesis, g_i, and it is calculated as following equation.

$$g_i = \sum_j \frac{1}{d_j \sqrt{1 + \left(\frac{1}{d_j}\right)^2}}, \tag{12}$$

where d_j denotes the distance of the descriptor vectors of the j-th match. The scoring scheme takes into account both of the distance of descriptor vectors as well as the number of the matches. Finally, the hypothesis of the top score is returned as the recognized scene if the score g is higher than a given threshold, η_g. All the parameters and thresholds mentioned so far are given in Table II.

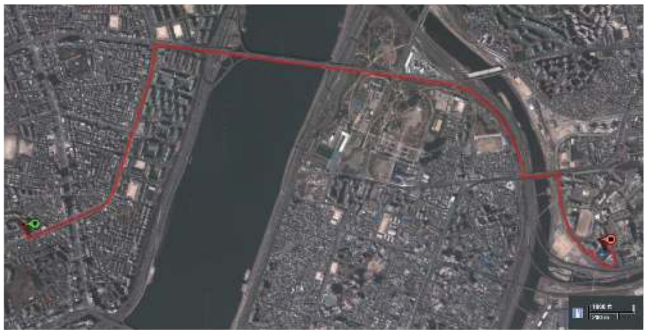


Fig. 7. The trajectory followed for the acquisition of the sequences used in the experiments.

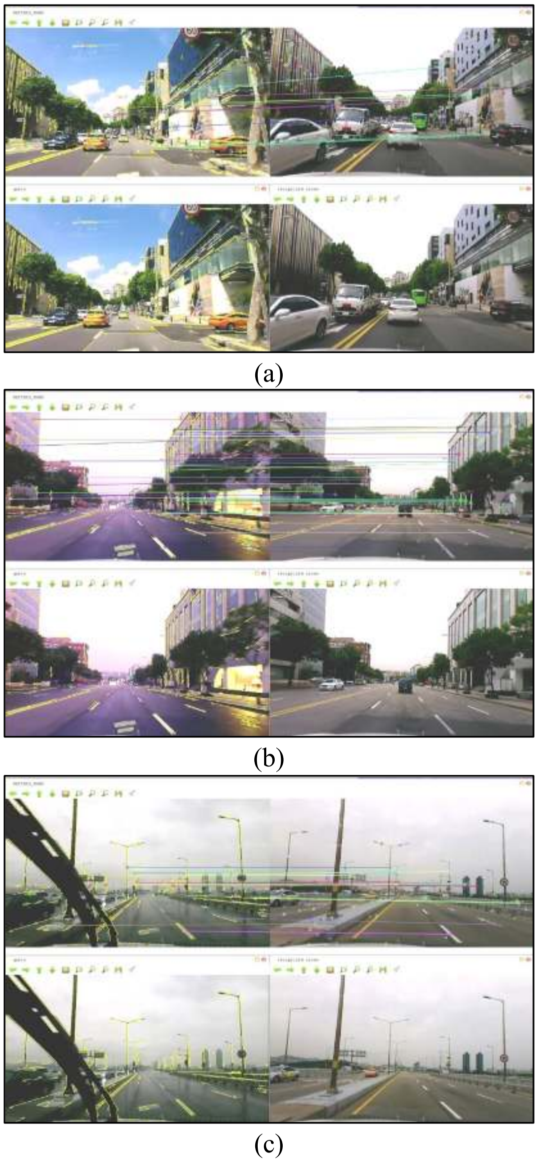


Fig. 8. Experimental results under three environmental changes: (a) season change, (b) illumination change, and (c) weather change. In each case, the left-bottom image is a query scene, and the right-bottom image is a recognized scene. The result of the motion-guided feature matching is shown above the query and recognized scenes.

IV. Experiments

In this section, we present the experimental results conducted under three environmental changes: season, illumination, and weather. For image acquisition, we used a black box camera (DBL-100, Dabonda, 130-degree of FOV.) equipped in a vehicle, and the optical distortions are removed before the experiments. The database contains 10,439 scenes, and they were gathered in three different days at around noon in the middle of September, 2013, which were in the fall, with driving scenarios in Seoul. About half of the scenes were acquired on roads, and the rest were gathered in the campus of Hanyang University. The trajectory followed in the sequence acquisition was about seven kilometers long.

Three sequences were also gathered in different days to use them as queries. The first sequence is gathered in summer in order to use it for the experiment under season change, and the second one is gathered in an early morning before sunrise for the experiment under illumination change. The last sequence is gathered in a rainy day for the experiment under weather change. All the experiments were performed in real-time using an Intel i7-2600K processor. The results are given in Table II, and demo videos can be seen at [20].

The flow of the algorithm in this experiment is as follows:

- The current input scene is queried to the vocabulary tree, and the tree returns the top m hypotheses.
- For each of the returns, features in the current scene and the hypothesis scene are matched with a distance threshold, η_{di} of descriptor vectors and a ratio threshold, η_{ri} of the nearest neighbor distances. If the ratio of the number of matches to the number of features in the query scene is higher than a threshold, η_a, the hypothesis takes further steps, or is discarded.
- For each of the hypotheses come from the previous step, the system estimates motions between the query and the hypotheses scenes.
- With the motions, the system matches features again between the two scenes with weaker thresholds, than in the second step (*i.e.* $\eta_d > \eta_{di}, \eta_r > \eta_{ri}$), and calculates the score g_i for each hypothesis.
- The top scored hypothesis i is returned as the recognized scene if g_i is higher than a threshold.

A. Experiment under Season Change

In this experiment, we used a sequence gathered in a summer while database scenes were gathered in a fall. Figure 8. (a) shows examples of the query and recognized scenes, and a motion-guided MSLD matching result between the two scenes is also given. As shown in Table II, total 872 scenes are queried and 356 and 205 scenes are recognized with different thresholds $\eta_a = 0.05$ and $\eta_a = 0.10$, respectively.

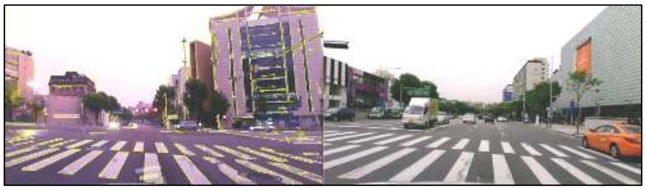


Fig. 9. An example of false positives. The repeated pattern in the database (right) scene satisfies both of close distance of descriptor vectors and geometric configurations with a pattern on the query (left) scene, and this leads to false positives.

TABLE II

Parameter settings and performances

	A		B		C	
resolution	720×405					
line length	20					
# database	10,439					
# msld db	1,546,231					
$\boldsymbol{\eta_{di}}, \boldsymbol{\eta_{ri}}$	0.4, 0.6					
$\boldsymbol{\eta_a}$	0.05	0.10	0.05	0.10	0.05	0.10
$\boldsymbol{\eta_d}, \boldsymbol{\eta_r}$	0.7, 0.7					
$\boldsymbol{\eta_g}$	0.05					
# query	872		1573		1917	
# msld query	144,229		235,704		286,012	
# recog	356	205	1199	966	1227	771
# false pos	0	0	8	0	14	0
# false neg	516	667	374	607	690	1146
avg time line [ms]	22.65		19.06		18.40	
avg time query tree [ms]	6.13	6.05	5.25	5.32	5.52	5.54
avg time opt [ms]	10.87	12.57	11.55	13.33	18.69	18.39
avg time init match [ms]	2.63	2.39	2.32	2.31	2.34	2.31
avg time init cost [ms]	4.86	5.98	5.75	6.93	5.09	6.64
avg time epi match [ms]	2.82	2.61	2.35	2.28	2.65	2.73
avg time query [ms]	129.39	106.63	222.60	193.69	133.18	114.54

From top to bottom, each row indicates **resolution**: image resolution used, **line length**: minimum length threshold in line extraction, **# database**: number of scenes in the database, **# msld db**: total number of MSLD descriptors in the database, $\boldsymbol{\eta_{di}}, \boldsymbol{\eta_{ri}}, \boldsymbol{\eta_a}, \boldsymbol{\eta_d}, \boldsymbol{\eta_r}, \boldsymbol{\eta_g}$: please refer to the main text, **# query**: number of queried scenes, **# msld query**: total number of MSLD descriptors in the query scenes, **# recog**: number of recognized scenes, **# false pos**: number of false positives, **# false neg**: number of false negatives, **avg time line**: average elapsed time for line segments extraction, **avg time query tree**: average elapsed time in querying to the vocabulary tree, **avg time opt**: average elapsed time of optimization for motion estimation, **avg time init match**: average elapsed time for initial MSLD matching, **avg time init cost**: average elapsed time for calculation of the initial costs, **avg time epi match**: average elapsed time for the motion-guided MSLD matching, **avg time query**: average elapsed time for a query.

B. Experiment under Illumination Change

For this experiment, we gathered a sequence starting at AM 6:01, 11 September, 2013, which is eight minutes earlier from the sunrise in Seoul. We can observe motion blurs on the sides of the images because the camera maximized its exposure. As shown in Table II, however, it results the least number of false negatives. We analyze this as an effect of the uncrowded roads. When the threshold $\eta_a = 0.05$, this experiment shows eight false positives, and an example of the false positives is shown in Figure 9. As shown in the figure, the query and the database scenes commonly have a repeated pattern on the roads satisfying both of the close distances of the descriptor vectors and the geometric configuration of the features, and this leads to the false positive. However, the false positives are not generated with stronger threshold $\eta_a = 0.10$ because it discards the hypothesis, but this also increases the number of true negatives.

C. Experiment under Weather Change

For this experiment, we acquired a sequence in a rainy day, and the windshield wipers of the vehicle were in operation. This experiment generates 14 false positives in 1,227 recognitions. By adjusting the threshold η_a from 0.05 to 0.10, the false positives are removed, but it also increases the number of the false negatives as in the other experiments. Figure 8. (c) shows an example of this experiment. Although raindrops on the windshield and the wipers made blur and occlusions, the system was not much affected.

D. Experimental Results

We evaluated the proposed algorithm in three different conditions. The precisions of the three experiments were 99.76%, 99.33%, and 98.86% in the same thresholds, respectively. The thresholds were set so that false-positive results are minimized. The experimental results revealed that our method can robustly recognize the place in significantly changed environments. It also denotes that implying the thresholds can be generalized to various environmental changes. In addition, the computational time measured as averagely 150 ms makes the demonstration real-time.

V. Conclusion

In this paper, we proposed an outdoor place recognition algorithm using only straight line features. A vocabulary tree built with line descriptors is used to find candidate matches, and a motion estimation algorithm is used to verify them. In order to evaluate the retrieval performance of the vocabulary tree built with MSLD line descriptors, we performed an experimental comparison with other tree built with SIFT, and the vocabulary tree trained with the line features exhibits better results in a structured outdoor environment. We tested our algorithm with three challenging environmental changes such as season, weather, and illumination. The database scenes consist of more than 10,000 images, and the experimental results demonstrated the real-time performance, and reliable accuracy of the precision rate higher than 98%.

ACKNOWLEDGMENT

This research was supported by the Global Frontier R&D Program on "Human-centered Interaction for Coexistence" funded by the National Research Foundation of Korea grant funded by the Korean Government (MEST) (NRF-M1AXA003- 2011-0028353). This work was also supported by the Industrial Strategic Technology Development Program (10044009) funded by the Ministry of Knowledge Economy (MKE, Korea).

References

[1] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based Maps," *The International Journal of Robotics Research (IJRR)*, vol.29, no.8, pp.941-957, July 2010.

[2] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research (IJRR)*, vol.27, no.6, pp.647-665, June 2008.

[3] A. Angeli, D. Filliat, S. Doncieux, and J. -A. Meyer, "Fast and Incremental Method for Loop-Closure Detection using Bags of Visual Words," *IEEE Transactions on Robotics (TRo)*, vol.24, no.5, pp.1027-1037, Oct. 2008.

[4] D. Filliat, "A Visual Bag of Words Method for Interactive Qualitative Localization and Mapping," *in Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp.3921-3926, April 2007.

[5] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol.60, no.2, pp.91-110, Nov. 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded-Up Robust Features," *Computer Vision - ECCV 2006*, vol.3951, pp.404-417, Jan. 2006.

[7] R. I. Hartley, "A Linear Method for Reconstruction from Lines and Points," *in Proc. of the Fifth International Conference on Computer Vision (ICCV)*, pp.882-887, June 1995.

[8] G. Klein, D. Murray, "Improving the Agility of Keyframe-based SLAM," *in Proc. of the tenth European Conference on Computer Vision: Part II (ECCV)*", pp.802-815, Jan. 2008.

[9] M. Chandraker, J. Lim, and D. Kriegman, "Moving in Stereo: Efficient Structure and Motion using Lines," *in Proc. of the IEEE 12th International Conference on Computer Vision (ICCV)*, pp.1741-1748, Sept.29 2009-Oct. 2 2009.

[10] G. Zhang and I. H. Suh, "A Vertical and Floor Line-based Monocular SLAM System for Corridor Environments," *International Journal of Control, Automation and Systems (IJCAS)*, vol.10, no., pp.547-557, June 2012.

[11] J. H. Lee, G. Zhang, J. Lim, and I. H. Suh, "Place Recognition using Straight Lines for Vision-based SLAM," *in Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.

[12] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," *in Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol.2, no., pp.2161-2168, June 2006.

[13] Z. Zhang, "Estimating motion and structure from correspondences of line segments between two perspective images," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol.17, no.12, pp.1129-1139, 1995.

[14] Z. Wang, F. Wu, and Z. Hu, "MSLD: A Robust Descriptor for Line Matching," *Pattern Recognition*, vol.42, no.5, pp.941-953, May 2009.

[15] H. Bay, V. Ferraris, and L. Van Gool, "Wide-Baseline Stereo Matching with Line Segments," *in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol.1, no., pp.329-336, June 2005.

[16] J. Sivic and A. Zisserman, "Video Google: a Text Retrieval Approach to Object Matching in Videos," *in Proc. of the IEEE 9th International Conference on Computer Vision (ICCV)*, vol.2, no., pp.1470-1477, Oct. 2003.

[17] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol.26, no.6, pp.756-770, 2004.

[18] R. I. Hartley, "In defense of the eight-point algorithm." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol.19, no.6, pp.580-593, 1997.

[19] http://opencv.willowgarage.com/wiki/

[20] http://youtu.be/4wGvINrHoy8