Place recognition using line-junction-lines in urban environments

Tang, Xiaoyu; Fu, Wenhao; Jiang, Muyun; Peng, Guohao; Wu, Zhenyu; Yue, Yufeng; Wang, Danwei

2020

Tang, X., Fu, W., Jiang, M., Peng, G., Wu, Z., Yue, Y., & Wang, D. (2019). Place recognition using line-junction-lines in urban environments. Proceedings of 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 530-535. doi:10.1109/CIS-RAM47153.2019.9095776

https://hdl.handle.net/10356/141837

https://doi.org/10.1109/CIS-RAM47153.2019.9095776

Place Recognition Using Line-Junction-Lines in Urban Environments

Xiaoyu Tang[1], Wenhao Fu[1], Muyun Jiang[1], Guohao Peng[1], Zhenyu Wu[1], Yufeng Yue[1] and Danwei Wang[1]

***Abstract*— Place recognition plays a vital role in eliminating accumulated drift from visual odometry in SLAM system. Bag-of-Words (BoW) -based approach is the most popular solution due to its efficiency and robustness. We propose to use Line-Junction-Line (LJL) to build a BoW for place recognition in urban environments. LJL is a simple structure of two lines with their intersection. Different from point features which are detected based on pixel intensity patterns, it represents structure with physical existence, which is more robust to challenging scenarios. Moreover, its descriptor is distinctive and encodes the relationship between the two lines. Experiments on KITTI dataset show the effectiveness of the proposed method compared to loop detection using BoW trained with either point or line features.**

I. Introduction

Feature-based Simultaneous Localization and Mapping (SLAM) extracts image features and projects them to 3D space as landmark to estimate the robot motion and build the environment map. Error exists in the estimated camera poses in between frames and it accumulates as the robot moves, making both the estimated trajectory and environment map erroneous. Visual place recognition algorithms recognize the place that the robot has visited before such that the robot is able to correct the cumulative error, thus improve the accuracy of the estimated robot trajectory and the consistency of the environment map. However, visual place recognition has been considered a very challenging task in that the appearance of a place can vary drastically during multiple traversals.

Visual place recognition can be viewed as an image retrieval task. Given a newly collected image, features are extracted to compare with the history images. Comparing image features exhaustively is computationally expensive, especially when the trajectory is long and the image database is large. BoW model [1] simplifies the process by quantizing the image features in descriptor space. Local features of an image are converted into a BoW vector, between which of the same location tend to have higher similarity metrics (e.g. L1/L2-scores).

BoW model relies on local features. The most widely used feature is point feature because it is simple to represent and easy to implement in the back-end optimization of SLAM system [2], [3]. However, its performance is still unsatisfactory under certain scenarios, such as motion blur, illumination or view-point change. In recent years, line feature shows promising potential as it expresses structural information of the environment and thus more robust to the above-mentioned challenging scenarios [4], [5]. However, line feature can be hard to track and localize due to the unstable endpoints. Moreover, the spatial relationship among visual words is deprived in BoW model. Many loop detection algorithms integrate a geometrical verification to reject false-positive matches which are geometrically inconsistent. The geometrical verification is based on camera transformation (e.g. 8-point RANSAC) between frames, which is computationally expensive. To address these issues, we propose to use higher level feature that encodes spatial relationship for place recognition.

(a) Detected LJLs.

(b) Detected ORB points

Fig. 1: LJLs and ORB points detected in a frame from KITTI dataset.

In this paper, we propose to utilize Line-Junction-Line (LJL) feature to achieve the goal of visual place recognition. LJL, first proposed to match line segments, is a structure of two lines together with their intersection, which can be easily found in urban scenes. A comparison between detected LJLs and ORB features in a frame from KITTI dataset [6] is shown in Fig. 1. LJLs, which are two lines located at corners, convey the main structure of the scene. Different from the widely used point feature which is detected only based on pixel intensity patterns, LJL is more robust since it is constructed by lines with physical existence, and it encompasses the spatial relationship between the two lines. Its descriptor is more distinctive than line descriptor since it describes the area around the intersection, which contains more texture. Loop closure is detected based on BoW trained using LJLs after temporal consistency check. Experiments on KITTI dataset showed improvement of the proposed method compared to either point or line feature, on both BoW effectiveness and loop detection recall values at 100% precision, with the capability to operate in real-time for key-frame based SLAM system.

The rest of the paper is structured as follows: Section II

*This work was supported by the ST Engineering-NTU Corporate Laboratory through the NRF Corporate Laboratory@University Scheme.

[1]All authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 {xiaoyu003, whfu, James.Jiang, peng0086, zhenyu002, yyue001, edwwang}@.ntu.edu.sg

discusses the related work in place recognition and BoW. Section III introduces the extraction of LJL and the BoW model. Section IV is detailed explanation of loop detection algorithm. The performance of the proposed method is evaluated in section V and section VI concludes the presented work.

II. Related Work

A. Visual Place Recognition

Global descriptor represent the image in a holistic way. It is fast to compute but less robust to illumination and viewpoint change. Color histograms [7] and Gist descriptor [8] were utilized to represent the image in place recognition. To overcome the low tolerance to illumination and viewpoint change, SeqSLAM [9] proposed to recognize consecutive images instead of searching for single frame. However, some scenarios, such as occlusions and camera rotations, are still challenging for methods using global descriptors.

The problems have been addressed more intensively with the development of local descriptors. FAB-MAP [10] is one of the earliest and most effective technique. The author proposed a probabilistic framework of learned visual words co-occupancy, where the visited places are represented by BoW of the objects in the scene. The system estimates the location model using a recursive Bayesian model. In the following work of FAB-MAP 2.0 [11], a sparse estimation scales the scheme's applicability by multiple orders of magnitude. Glvez-Lpez and Tards [2] proposed a faster solution by representing an image using a visual word vector, which is generated by a tree-structured BoW with binary descriptor (BRIEF [12]). Adjacent frames are grouped into islands to reinforce the individual image matching results. Mur-Artal and Tards [3] further developed the framework using ORB [13] descriptor to achieve rotation invariance. Since there is no spatial information utilized in the BoW schemes, the above two algorithms require a geometry verification based on the transformation between the candidate matching pairs to validate the correct recognized place. Bampis et al. [14] utilized the spatial information and proposed Structural-Aware Viewpoint-Invariant High-Order Visual-Words (SVHV) for BoW. Visual words are grouped into SVHVs based on optical flow between successfully matched features in consecutive frames. Though the algorithm considered the spatial relationship between features, it still suffers from the constraints of point feature and it is only valid when there is little camera rotation. Instead of training BoW in advance, there are also methods that generate visual words in an on-line manner. Angeli et al. [15] proposed a probabilistic framework based on Bayesian filtering which combines both SIFT [16] feature and color histograms. [17] proposed to build a binary BoW incrementally by tracking features in consecutive frames. Loop closure candidates are detected by a likelihood function, which is further verified by a temporal consistency check. [18] also proposed a loop detection method using incremental BoW strategy with dynamic time islands to deal with similar frames that are close in time.

In recent years, Convolutional Neural Network (CNN) has been utilized to address extreme appearance change in place recognition [19], [20], [21]. Despite their impressive results, they are still disconnected from the SLAM system due to large demand for computational resources.

B. Line Feature in Place Recognition

In feature based SLAM system, features play an essential role on accurate localization and consistent map building [22], [23]. The most widely used feature is point feature [24]. ORB point [13] is fast to compute, easy to localize and convenient to implement in the back-end optimization of SLAM system. Nevertheless, line feature conveys structural information of the environment and more robust to appearance change, which benefits the task of visual place recognition, as well as SLAM system.

With the development of line segment detection algorithms [25], [26] and line descriptors [27], [28], line feature was utilized in place recognition and localization. Lee et al. [4] first built a vocabulary tree using line descriptor (MSLD [27]) in a Bayesian filtering framework for place recognition in indoor environment. Their following work [5] achieved outdoor place recognition with a geometry consistency check using a line-based motion estimation. The results showed the effectiveness of place recognition using line feature in man-made environment. More recently, binary point descriptor [13] and line descriptor [28] are associated in place recognition task. Yang et al. [29] modified vocabulary tree to combine point and line feature for effective place recognition for texture-less scenes. Gomez-Ojeda et al. [30] built BoW for points and lines respectively and designed a dual search based on the fact that the two features are good at describing different scenes.

However, line segment is hard to localize because the endpoints are unstable and can shift along the lines. Besides, the line descriptor describes the gradient characteristics within the area along the two sides of the line, which are usually texture-less planes, making the descriptor less distinctive. Therefore, we propose to use two lines together with their intersection as feature for place recognition. The descriptor describes the local area around the intersection, which contains more texture than the area along the lines, and also encodes the spatial relationship between the two lines, making it more robust for place recognition than point or line feature.

III. Line-Junction-Line BoW

This section explains the method we proposed using LJL for place recognition. Similar to other features used in BoW model, we trained a BoW using 4 million LJL descriptors extracted from videos taken by a camera mounted on a car travelling in downtown areas of cities in the U.S.. The query results are checked with temporal consistency for loop detection.

A. Feature Extraction

After extracting line segments from the image, LJL is constructed and a descriptor for the structure is generated as proposed in [31].

1) LJL Construction: In order to extract line segments from image, we adopt LSD [26], a linear-time line segment detector, which provides highly repeatable results with sub-pixel precision. However, the endpoints of line segments are

Fig. 2: Illustration of LJL descriptor.

usually unstable and can shift along the line. Lines with physical intersection might not have intersections in the image. Therefore, we search neighbouring lines in a local rectangle region near a target line segment and find the intersection. Lines with at least one endpoint in the rectangle region are extended to find the intersection with the target line. To make sure that the LJL is reliable, the intersection should be also within the local rectangle region of the target line. Such construction criteria ensures most of LJLs are extracted from real structures and thus more robust in challenging scenarios.

2) LJL Description: After the construction of LJL, a descriptor is generated using gradient orientation histograms in the region around the intersection, as illustrated in Fig. 2. The local region is two concentric circles centered at the intersection, where the radius of the smaller circle is half of the radius of the larger one. The two circles are divided into four parts by the two lines and their extended lines. Each part is composed of a sector and a ring-shaped area. The ring-shaped area is further divided into 3 sub-regions evenly, making the area of each sub-region same with that of the inner sector. The gradient histograms are calculated from every sub-regions and sectors, with 8 bins in each histogram, yielding a vector of 128 dimensions. In order to eliminate the effect of illumination change, the descriptor is normalized based on the size of the bins.

B. BoW Model

BoW is a learned visual words which can convert the local feature descriptors in an image into one vector. It is trained offline by segmenting the feature space into W visual words, with a tree-structure to accelerate the searching speed. The descriptors are clustered into k_w kernels by performing k-medians clustering using k-means++ [32]. These kernels are the nodes in first level of the vocabulary tree and descriptors are assigned to the closest kernel. Similarly, the following levels are created using the associated descriptors to each kernels until there are L_w levels, resulting in W visual words. To eliminate the effect of indiscriminative words, each word is assigned with a weight based on term frequency-inverse document frequency, *tf-idf* [1]:

$$w_i = \log \frac{N}{n_i} \tag{1}$$

where N is the number of training images, n_i is the number of images with word i. The more often a word appears, the smaller weight it has. To convert an image to a BoW vector, features traverse the tree from the root nodes to the leaf nodes, by minimizing the distance with nodes in each level. Here we trained the BoW using the setting of $k = 10$, $L = 5$.

To evaluate the similarity between two BoW vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, we calculate L_1-norm, which varies in [0, 1]:

$$s(\mathbf{v}_1, \mathbf{v}_2) = 1 - \frac{1}{2}\left|\frac{\mathbf{v}_1}{|\,\mathbf{v}_1\,|} - \frac{\mathbf{v}_2}{|\,\mathbf{v}_2\,|}\right| \tag{2}$$

C. Loop Detection

To detect loop closures, we apply the method proposed in [2].

1) Database query: When a new image is collected at time t_i, the image is converted to a BoW vector $\mathbf{v}_{t_i}$. The database of BoW vectors computed from history images is queried, obtaining similarity score $s(\mathbf{v}_{t_i}, \mathbf{v}_{t_j})$, corresponding to matching candidate $< \mathbf{v}_{t_i}, \mathbf{v}_{t_j} >$. The score is normalized using the similarity score of the previous image $s(\mathbf{v}_{t_i}, \mathbf{v}_{t_{i-1}})$, resulting in η:

$$\eta(\mathbf{v}_{t_i}, \mathbf{v}_{t_j}) = \frac{s(\mathbf{v}_{t_i}, \mathbf{v}_{t_j})}{s(\mathbf{v}_{t_i}, \mathbf{v}_{t_{i-1}})} \tag{3}$$

where $\mathbf{v}_{t_{i-1}}$ is the BoW vector of the previous image. The normalized score indicates the relative similarity to the previous frame, based on the assumption that the previous image is the ideal loop, such that it can be adopted in wider types of scenarios. However, if $s(\mathbf{v}_{t_i}, \mathbf{v}_{t_{i-1}})$ is too small (e.g. when the camera motion is pure rotation), η will be incorrectly too large. To solve this problem, a threshold α is set for $s(\mathbf{v}_{t_i}, \mathbf{v}_{t_{i-1}})$. If $s(\mathbf{v}_{t_i}, \mathbf{v}_{t_{i-1}})$ does not reach the threshold, the match is rejected in loop detection.

2) Match grouping: To avoid the competence between images that are close to each other, single matches are grouped together and treated as one match. Time stamps t_{j_1},..., t_{j_n} are aggregated into one group notated as T_j, and matching candidates $< \mathbf{v}_{t_i}, \mathbf{v}_{t_{j_1}} >$,..., $< \mathbf{v}_{t_i}, \mathbf{v}_{t_{j_n}} >$ are grouped into one candidate $< \mathbf{v}_{t_i}, \mathbf{V}_{T_j} >$, with the grouped score H:

$$H(\mathbf{v}_{t_i}, \mathbf{V}_{T_j}) = \sum_{k=j_1}^{j_n} \eta(\mathbf{v}_{t_i}, \mathbf{v}_{t_k}) \tag{4}$$

Apart from preventing consecutive images competing with each other, match grouping also gives emphasis on correct loops. If two frames I_{t_i} and I_{t_j} is real loop closure, $s(\mathbf{v}_{t_i}, \mathbf{v}_{t_j})$ is likely to be similar to $s(\mathbf{v}_{t_i}, \mathbf{v}_{t_{j\pm1}})$, $s(\mathbf{v}_{t_i}, \mathbf{v}_{t_{j\pm2}})$,..., resulting in a longer group. H is summation of η and prefers longer group. The group with the highest H score is the matching group, in which the frame with the highest similarity score is selected as the matching candidate, and it is verified with temporal consistency check.

3) Temporal consistency check: Matches of real loop closure should be consistent in consecutive query images. The match $< \mathbf{v}_{t_i}, V_T >$ must be consistent with k previous matches $< \mathbf{v}_{t_{i-1}}, V_{T_1} >$,..., $< \mathbf{v}_{t_{i-k}}, V_{T_k} >$, and T_j and T_{j+1} are close enough. If a group passes the temporal consistency check, the match with the highest normalized score $< \mathbf{v}_{t_i}, \mathbf{v}_{t_j} >$ is considered the loop detection result.

IV. Experimental Evaluation

A. Methodology

1) Dataset: We used sequences from KITTI odometry dataset. KITTI contains stereo image sequences in urban or

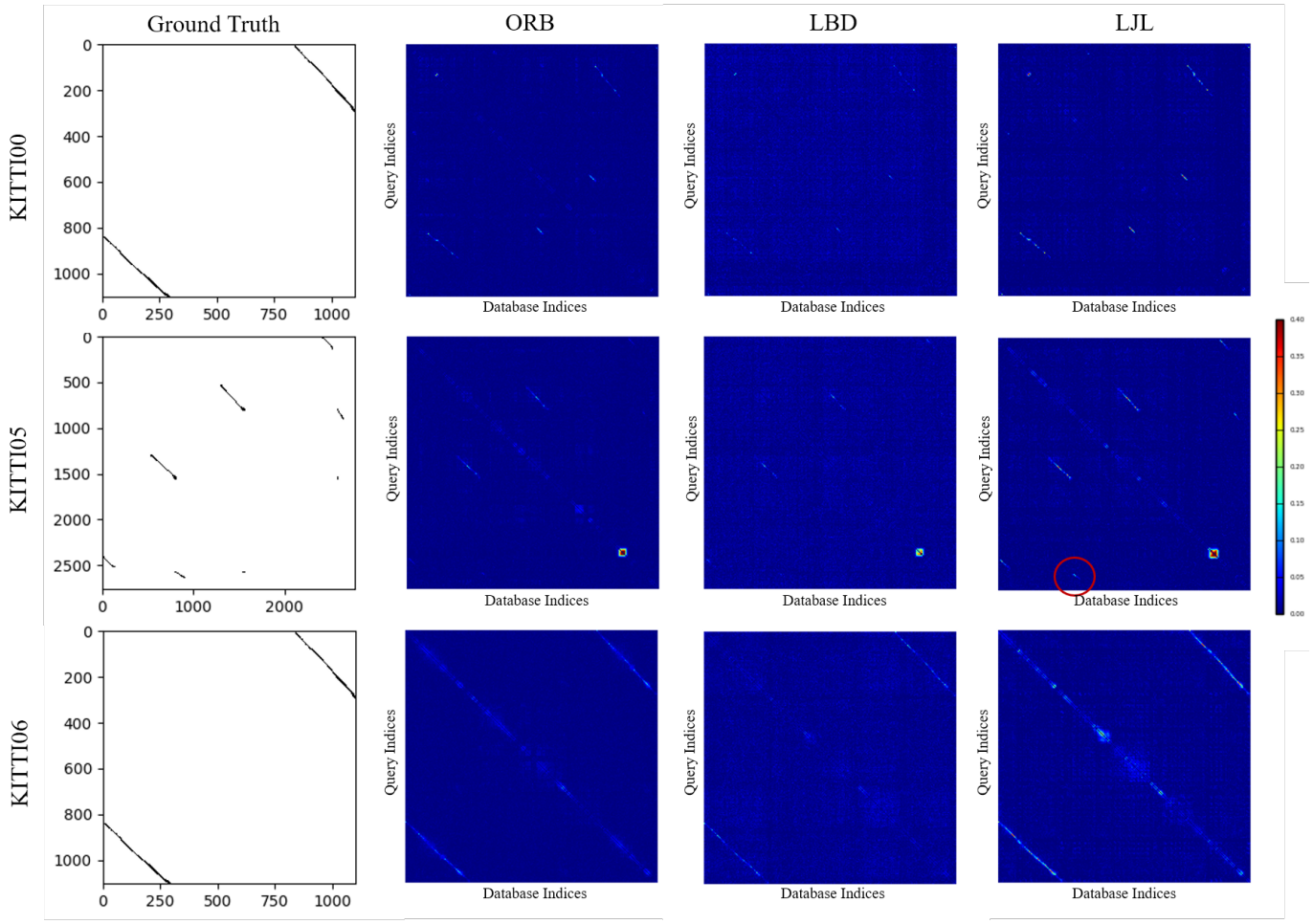


Fig. 3: Similarity matrices generated by querying BoW on sequence 00, 05 and 06 from KITTI. **First column:** ground truth of loop closure calculated from ground truth camera pose with distance threshold 5m. **Second column:** similarity matrices generated by querying ORB BoW provided by [30]. **Third column:** similarity matrices generated by querying LBD BoW provided by [30]. **Forth column:** similarity matrices generated by querying the LJL BoW we trained.

sub-urban environment with precise ground truth camera pose. We selected sequence 00, 05, and 06, which are mostly urban scenes and contain large amount of loop closure. We used image sequences from the left camera only.

2) Ground truth: To measure the accuracy of the loop detection results, we need the ground truth loop closures to compare the results with. There is no ground truth label of loop closures in KITTI dataset, while we obtained it from the ground truth camera pose. In case consecutive frames are labeled as loop closure, we discarded frames that are within 100 frames from each other. In Fig. 3, the first column shows the loop closure ground truth calculated from the odometry. The horizontal and vertical axes are database frame indices and query frame indices, respectively. Black dots indicate that the corresponding query frame and database frame are at the same place.

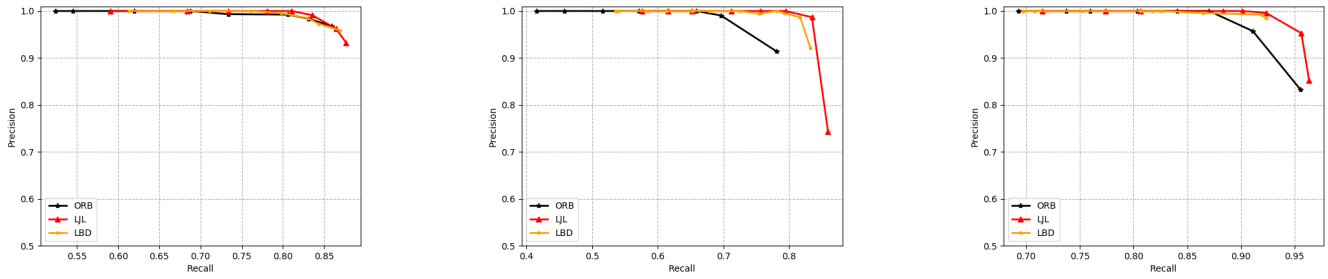
3) Evaluation metrics: We evaluated both the BoW effectiveness and the loop detection results. The BoW effectiveness is evaluated using similarity matrix, which is the similarity scores between query image and database images. The larger the difference between loop closure frame pairs and non loop closure frame pairs is, the more likely the retrieval result is correct. We compare it with similarity matrices obtained by BoW trained using ORB and LBD, which are provided by [30]. The loop detection results are evaluated using precision and recall, which are defined as the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

where TP denotes the number of true-positive detection, FP corresponds to the false-positive and FN to the false-negative ones. Hence precision means the ratio between the number of correct loop detection and the total number of loop detection, and recall is the ratio between the number of correct loop detection and the number of real loop closure events in the ground truth. Note that the loop detection in our experiments is without geometrical verification.

4) Parameter settings: For LJL, we used the LSD in OpenCV to detect line segment, and set the expanding width w as 5 pixels to make sure that the feature is more likely to be extracted from real structure, and there is also tolerance for unstable endpoint position. In the loop detection algorithm, we set $k = 3$ for temporal consistency check. Other parameters



(a) KITTI00 (b) KITTI05 (c) KITTI06

Fig. 4: Precision-recall curves of loop detection on images sequences from KITTI dataset using BoW trained with different features.

were kept same as the authors provided. We extracted 1000 ORB points from each frame as [3] did.

B. BoW effectiveness

we evaluated the effectiveness of the BoW using similarity matrix. Similarity matrix contains similarity score between query and database images, which varies between 0 and 1. As shown in Fig. 3, the score is represented by color, the second and third columns are results of ORB and LBD BoW from [30], respectively. The last column is the result generated by the BoW we trained.

Overall, the scores are higher at real loop closure of LJL similarity matrices compared to that produced by ORB and LBD. The difference among similarity scores of LBD BoW is generally smaller compared to that produced by the other two descriptors. While there are some high similarity scores at non loop closure pairs in ORB BoW similarity matrices, which tends to produce false positives. Note that the high scores around the diagonal on the lower right in sequence 05 is because the car stops at a crossroads for some time. In particular, the part that is illustrated in the red circle corresponds to real loop closure, while there is no obvious higher similarity scores in either ORB or LBD similarity matrices.

C. Loop detection results

We changed the threshold α to get different sets of precision and recall values and plotted them as precision-recall curves, as shown in Fig. 4. The detection results of LJL BoW achieve the best precision-recall performance among the three features. Following is that of LBD, which obtains very competent precision and recall performance. However, we care more about precision than recall in loop detection, because a wrong detection would result in a wrong pose graph, which can be highly hazardous for robot navigation. Therefore, we measured the recall values at 100% precision, which can be found in Table I. Loop detection using LJL BoW achieves the highest recall values at 100% precision in all three sequences. The precision-recall curve of ORB BoW on sequence 00 is similar to the other two curves because there are quite a number of scenes which are mostly vegetation, which lacks structural information.

TABLE I: Comparison of the loop detection recall rates (%) for 100% precision using BoW trained using different descriptors.

	KITTI00	KITTI05	KITTI06
ORB	68.78	65.84	87.03
LBD	76.12	67.86	82.48
LJL	**81.09**	**79.46**	**90.15**

TABLE II: Execution time of each part of the algorithm.

Procedure		Execution time (ms)
LJL extraction	LSD	34.81
	Construction	13.67
	Description	41.87
Loop detection		2.14
Total		92.49

D. Execution time

We ran the algorithm on 3 image sequences from KITTI dataset of 8403 images, and recorded the average execution time of each step of the algorithm. The experiments were done on an Intel Xeon E5-1630 v3 @ 3.70GHz CPU. The results are shown in Table II. The total execution time per query is 92.49ms, making it eligible to operate the algorithm in real-time for a key-frame SLAM framework (100-200ms per frame).

V. Conclusion

We proposed a simple method for place recognition in urban environment using LJL feature, which conveys structural information of lines and the relationship between them, also encodes spatial relationship between two lines in its descriptor. The BoW trained using LJLs achieves better performance on BoW effectiveness and loop detection results, compared to that using either point or line feature. Though slower than binary descriptor, the proposed loop detection algorithm can operate in real-time for key-frame based SLAM. Further improvement can be made on reducing the execution time in the feature extraction step to achieve better efficiency.

References

[1] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *null*. IEEE, 2003, p. 1470.

[2] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[3] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based slam," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 846–853.

[4] J. H. Lee, G. Zhang, J. Lim, and I. H. Suh, "Place recognition using straight lines for vision-based slam," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3799–3806.

[5] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh, "Outdoor place recognition in urban environments using straight lines," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5550–5557.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[7] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omnidirectional camera," *IEEE Transactions on robotics and automation*, vol. 16, no. 6, pp. 890–898, 2000.

[8] G. Singh and J. Kosecka, "Visual loop closing using gist descriptors in manhattan world," in *ICRA Omnidirectional Vision Workshop*, 2010.

[9] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1643–1649.

[10] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[11] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[12] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.

[14] L. Bampis and A. Gasteratos, "Revisiting the bag-of-visual-words model: A hierarchical localization architecture for mobile systems," *Robotics and Autonomous Systems*, vol. 113, pp. 104–119, 2019.

[15] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, pp. 1027–1037, 2008.

[16] P. C. Ng and S. Henikoff, "Sift: Predicting amino acid changes that affect protein function," *Nucleic acids research*, vol. 31, no. 13, pp. 3812–3814, 2003.

[17] S. Khan and D. Wollherr, "Ibuild: Incremental bag of binary words for appearance based loop closure detection," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5441–5447.

[18] E. Garcia-Fidalgo and A. Ortiz, "ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3051–3057, 2018.

[19] N. Sunderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," *arXiv preprint arXiv:1501.04158*, 2015.

[20] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[21] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *European conference on computer vision*. Springer, 2016, pp. 3–20.

[22] Y. Yue, P. G. C. N. Senarathne, C. Yang, J. Zhang, M. Wen, and D. Wang, "Hierarchical probabilistic fusion framework for matching and merging of 3-d occupancy maps," *IEEE Sensors Journal*, vol. 18, no. 21, pp. 8933–8949, Nov 2018.

[23] Y. Yue, C. Yang, Y.Wang, P. G. C. N. Senarathne, J. Zhang, M. Wen, and D. Wang, "A multi-level fusion system for multi-robot 3d mapping using heterogeneous sensors," *IEEE Systems Journal*, 2019.

[24] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[25] C. Akinlar and C. Topal, "Edlines: A real-time line segment detector with a false detection control," *Pattern Recognition Letters*, vol. 32, no. 13, pp. 1633–1642, 2011.

[26] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.

[27] Z. Wang, F. Wu, and Z. Hu, "Msld: A robust descriptor for line matching," *Pattern Recognition*, vol. 42, no. 5, pp. 941–953, 2009.

[28] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[29] S. Yang, W. Mou, H. Wang, and S. S. Ge, "Place recognition by combining multiple feature types with a modified vocabulary tree," in *2015 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE, 2015, pp. 1–6.

[30] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4503–4508.

[31] K. Li, J. Yao, X. Lu, L. Li, and Z. Zhang, "Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation," *Neurocomputing*, vol. 184, pp. 207–220, 2016.

[32] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.