**Appearance based filtering of matched line segments with topological constraints**

Ik Hyun Jo, Jae Seok Jang, Soon Ki Jung

School of Computer Science and Engineering, Kyungpook National University

80 Daehakro, Buk-gu

Daegu, Republic of Korea

dlrgus26@gmail.com, jaeseok.jang@gmail.com, skjung@knu.ac.kr

**Abstract**

**We present a filtering method of matched line segments based on topological constraints. To get an initial matched pairs of line segments, we adopted a Line Band Descriptor (LBD) descriptor[1] to the authentic line matching method. Our filter tests every matched line pairs from two images which are taken with a slightly different viewing angle. Our basic idea is that, if two images have small view difference, topological constraints of the paired lines should remain the same for both images. Our filter works in a way that if three pairs of lines from the two images are parallel, then their ordering of the paired line segments is first checked. The order of those three pairs of line segments should be same in both images. If three lines are non-parallel, our filter checks their ordering of the cross points between the lines. The filter considers one reference line segment and randomly selected two line segments which intersect with the reference line and they are considered as supporting lines. We find cross points on the reference line and these points are used for our filtering method. With our filtering, violation scores are calculated and used for culling mis-matched pairs. Our method provides two crucial advantages over the existing line matching technique. First is the increase in accuracy. Not only using appearance information but also using topological constraints for line matching, we can get more accurate matching line pairs. Second, our filter can also work in real-time. Adoption samples before performing filtering steps make our filter more adapted for many real-time vision applications.**

Keywords- line matching; line filtering; line correspondence;

**I. Introduction**

Corresponding feature extraction is a key stage in many computer vision task such as visual SLAM ([Simultaneous localization and mapping](https://www.google.co.kr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjQmrjQjKnXAhVKjpQKHQW5CW0QFggkMAA&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSimultaneous_localization_and_mapping&usg=AOvVaw2LyVkb1i-4IcIjuDyuOXwY)) [2], stereo matching[3], scene reconstruction[4], etc. More accurate matched line pairs allow better result for these tasks. Line matching methods can be performed in different ways; (1) based on global geometric constraint such as epipolar geometry[5], (2) appearance based matching such as Mean-Standard Deviation Line Descriptor (MSLD) [6] and Line Band Descriptor (LBD) [1]. However, some vision techniques such as visual SLAM cannot use epipolar geometry for matching. In this case, appearance based feature can be a good way to get corresponding line pairs. But there is no assurance that the descriptors will be perfect because these methods consider only how the line segments appear. So there are some techniques like RANSAC[7], that can find obtrusive line pairs after matching. For real-time application, this method cannot be applied because of high computational cost.

In this paper, we implemented a simple but very powerful method of filtering mismatched line pairs from appearance based matching with topological constraints.

The paper is organized in five sections. Section II provides details about previous line feature matching and topology filter. Section III describes our main idea of filter followed by section IV with the experimental results. Finally, we concludes in section V.

**II. Related work**

*A. Line matching descriptor*

For line matching, we introduce two popular representative descriptors, MSLD and LBD. Both descriptors are based on appearance. To build MSLD, Zhiheng defines a pixel support region (PSR) and makes a line gradient description matrix (GDM) by characterizing each sub-region into a vector. Then, MSLD is built with mean and standard deviation of GDM column vectors. LBD descriptor defines a line support region (LSR) like PSR. LBD descriptor is also built by sub-region, mean and standard deviation, but its shape is different from MSLD. One of the main difference between LBD and MSLD is multi-scale line detection strategy and it allows LBD more robust to image transformations.

*B. Topology filter*

Our main idea is inspired by Herbert Bay’s topology filter [3]. It is semi-local spatial arrangement of features of two views. In this filter technique, mid-points of lines are used to clarify their alignments and three lines make one test case. Two mid-points make a line and remaining mid-point’s position determines the test result. If remaining mid-points in both images do not lie on the same side from the line formed by two points, in this case, it doesn’t pass the filter. The filter tests all triplet cases of every matched line features. However, the main idea has two drawbacks. First, mid-points are not robust because the point positions depend on measured lengths of the line segments so much. If lines which are matched as pair have far different lengths, then the filter may produce wrong results. Second is the computation time. Testing every triplet case takes too much calculation time and it depends on the counts of line features in the image also. If the number of line features increases, test cases also increase exponentially. Therefore, we suggest a filter that can compare alignments of line features directly without mid-points. Also we adapted a sampling to make test cases and it allows us perform in real-time.

**III. Triplet filter**

*A. Initial matching*

Initial matching step is a crucial step before filtering. For initial matching, we adopted LBD descriptor because of two reasons. First, it shows good performance compared to other methods like MSLD or color based matching. Second, LBD descriptors can be easily converted into binary form for matching algorithm. It is a very important factor for overall computation time. Thus, computation time for filtering will take less eventually leading to the application in real time scenarios.

LBD is calculated and converted into binary form, then using hamming distance calculation to find a closest line pair from both images. Comparing occurs for both sides, image1 comparing to image2 and image2 to image1. Initial matched line segments are selected if the line pairs have the same matched when compared from the selected two images.

*B. Sampling*

Every matched line pairs are considered as a reference line and each line segment is tested with randomly selected $N$ sets of two lines. We consider only two cases; first is when three lines are parallel and second is when there are no parallel lines with the reference line. Usually parallel lines in 3D scene are non-parallel in 2D image because of the change in the perspective view. If the angle between two lines is smaller than 30 degrees, we consider those lines as parallel lines.

*C. Line filter*

The basic idea of our filter is that if the view difference between two image frames are small, topological relation among line segments should be remain. Our filter works in two ways. One is for three parallel lines case and second is non-parallel lines with a reference line.

Fig. 1 shows a test case for three parallel lines. In this scenario, the filter checks ordering of line segments in both images and if the order does not match, we consider this case as a violation case. Fig. 2 shows a test case about non-parallel line segments. To check a topological consistency, we used intersection points of lines. Find intersection point of L1 and L2 lines and also intersection point for lines L1 and L3. L1 has direction from LBD descriptor and intersection points with L2 and L3 lie on L1. If these intersection points on line L1 alignment changes, we consider this as a violation case.

We define violation score in equation (1):

$V= N\_{v} / N$ , (1)

While $N\_{v}$ is the number of violation cases of each line and N is total testing cases for each line. If $V>threshold$, we consider tested line matched pairs are incorrect and remove this matched pairs.



Fig. 1 Test case of three parallel lines.



Fig. 2 Test case of non-parallel lines with testing line L1.

**IV. Experiments**

*A. Data set and testing environments*

Our tests are performed on AMD Phenom(tm)II X6 1090T Processor, 4 GB Memory, Ubuntu 14.04.1 OS. Tested images are taken by a monocular camera. Each pair of images has small view difference. Fig. 3 shows a series of datasets; (a) to (e) are datasets taken in the indoor environment, and (f) to (h) images are taken in the outdoor environment. For our experiment, we set the threshold value to 0.40.



Fig. 3 Small view difference image pair sets. (a)-(e), indoor environments and (f)-(h), outdoor environments.

*B. Experimental results*

Table I shows that the number of matched line pairs after filtering is reduced compared to the initial matching result. However, we could get results with an increased accuracy after filtering. In the indoor environments, (a) to (e) cases show that the initial match already show high accuracy as compared with outdoor environment cases (f) to (g). The results show low accuracy because of repeated buildings patterns. Our filter remove not only mis-matched line pairs but also proper line pairs but it still provides higher final accuracy.

TABLE I

COMPARE ACCURACY OF INITIAL MATCHING RESULT

AND FILTERED RESULT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| InputImage | InitialMatched Lines | InitialAccuracy (%) | FilteredMatch Lines | FinalAccuracy (%) |
| (a) | 40 | 92.5 | 39 | 94.9 |
| (b) | 48 | 87.5 | 46 | 91.3 |
| (c) | 47 | 89.3 | 46 | 91.3 |
| (d) | 80 | 96.3 | 78 | 98.7 |
| (e) | 82 | 96.3 | 81 | 97.5 |
| (f) | 84 | 58.3 | 50 | 80.0 |
| (g) | 182 | 64.9 | 173 | 68.2 |
| (h) | 95 | 83.2 | 92 | 85.9 |

Table II shows the runtime of initial matching and filtered matching steps in seconds(s). In both steps, initial matching and filtering, execution time increased in proportion to the number of lines in the image. But after filtering it shows enough runtime applicable to real-time applications.

TABLE II

RUNTIME OF MATCHING AND FILTERING STEPS

|  |  |  |
| --- | --- | --- |
| Input Image | Initial Match (s) | Filtered Match (s) |
| (a) | 0.0013 | 0.0341 |
| (b) | 0.0018 | 0.0424 |
| (c) | 0.0017 | 0.0413 |
| (d) | 0.0027 | 0.0708 |
| (e) | 0.0027 | 0.0739 |
| (f) | 0.0137 | 0.0977 |
| (g) | 0.0458 | 0.2403 |
| (h) | 0.0113 | 0.1023 |

**V. Conclusion**

We address the problem of appearance based line matching. Our approach is more accurate than the existing method and minimizes the runtime. First, we used sampling for picking datasets for reducing calculation times. Second, we proposed a filter using topological relations to remove mismatched line pairs from two images that have small view difference. Our proposed method has shown excellent performance in reducing errors caused by existing line feature based matching. Compared to existing methods, it showed more accurate results and enough runtime to work in real time. However, the filtering performance of final line matching accuracy depends on the result of the initial matching. Therefore for the future work, there is a need for improvement in the initial matching for the overall improvement in the filter.

**Acknowledgment**

This research was supported by the CEST (Center for Embedded Software Technology).

**References**

[1] [L. Zhang](http://www.sciencedirect.com/science/article/pii/S1047320313000874%22%20%5Cl%20%22%21) and [R. Koch](http://www.sciencedirect.com/science/article/pii/S1047320313000874#!), “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, pp. 794-805, October 2013.

[2] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, “MonoSLAM: Real-Time Single Camera SLAM,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 29, pp. 1052-1067, April 2007.

[3] H. Bay, V. Ferraris, L. V. Gool, “Wide-baseline stereo matching with line segments,” *IEEE Computer Society Conference on Computer Vision and Pattern* *Recognition*, 2005.

[4] V. Kolmogorov, R. Zabih, “Multi-camera Scene Reconstruction via Graph Cuts,” *European Conference on Computer Vision*, 2002.

[5] C. Schmid, A. Zisserman, “Automatic line matching across views,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[6] Z. Wang, F. Wu, Z. Hu, “MSLD: A robust descriptor for line matching,” *Pattern* *Recognition*, vol 42, pp. 941-953, May 2009.

[7] S. Choi, T. Kim, W. Yu, “Performance Evaluation of RANSAC Family,” *Proceedings of the British Machine Vision Conference*, 2009.