Line Segment Matching: A Benchmark

Kai Li Jian Yao† Mengsheng Lu Yuan Heng Teng Wu Yinxuan Li
School of Remote Sensing and Information Engineering, Wuhan University
{kaili, †jian.yao}@whu.edu.cn

Abstract

As the vital procedure for exploiting line segments on images for solving computer vision problems, Line Segment Matching (LSM), has been attached with special importance from researchers, and a considerable number of methods have been proposed in recent decades. However, no one has attempted to solve two major problems in this area. The first is how to evaluate different methods in an unbiased way. All proposed methods were evaluated on the images and extracted line segments selected by the authors themselves, making the conclusions based on the somewhat biased experiments less convincing. The second problem is that there is no reliably automatic way to access the correctness of the obtained line segment matches, which can often be up to hundreds in quantity. Checking them one by one through visual inspection is the only reliable, but very tedious and error-prone way. In this paper, we target to solve the two problems. We introduce a benchmark which provides the ground truth matches among 30 pairs of line segment sets extracted from 15 representative image pairs using two state-of-the-art line segment detectors. With the benchmark, we evaluated some of the existing LSM methods.

1. Introduction

As important features in images, especially those photographing man-made scenes, line segments, compared to feature points, are somewhat neglected by researchers. There are relatively few works exploiting line segments for solving computer vision problems, though some successful attempts have been made in exploiting line segments for 3D reconstruction [5, 4, 3, 6, 8], SLAM [10, 11], structure from motion [7, 2, 1], pose estimation [12, 13], etc. A major cause for the less exploitation of line segments lies in the difficulties matching them.

The difficulties result from these factors. First, there is no point-to-point correspondence in corresponding line segments from different images. Due to noise and occlusion, the endpoints of the 2D correspondences of a 3D line segment in different images often do not correspond with each other. This fact is the foremost reason complicating Line Segment Matching (LSM) and its applications. Second, due to the segmentation procedure while detecting line segments, the connectivity of line segments is lost, producing fragmented line segments, making it hard for exploiting the geometric relationship between line segments to match them.

Despite of the difficulties, a growing number of LSM methods have been proposed in recent years. However, no one has attempted to solve two major problems in this area. The first is how to evaluate different methods in an unbiased way. All proposed methods were evaluated on the images and line segments selected by the authors themselves, resulting in the less convincing conclusions. The second problem is that there is no reliably automatic way to access the correctness of the obtained line segment matches. Checking them one by one through visual inspection is the only reliable way, but can be a tedious and error-prone work when there are hundreds of them. For example, in the two images shown in Figure 1. The line segment detector, LSD [34] extracted 1071 and 1016 line segments in the two images, respectively. With the two sets of line segments, the LSM method, LJL [33] found 749 pairs of correspondences. Accessing the correctness of them one by one through visual inspection is definitively an arduous work. We solve the two problems by establishing a benchmark which provides the ground truth matches among 30 pairs of line segment sets extracted from 15 representative image pairs using two state-of-the-art line segment detectors.

The first problem is solved by the benchmark because we unify the experimental images and line segment detectors and limit them on some representative ones. When the ground truth matches between two sets of line segments are known, the second problem, automatically and reliably evaluating the correctness of the obtained line segment matches by certain methods, is solved too. Researchers just need to compare their obtained results with the ground truth, and a detailed and reliable report of their results, with the annotations of the correct and incorrect matches among the obtained results and the matches that have not been found

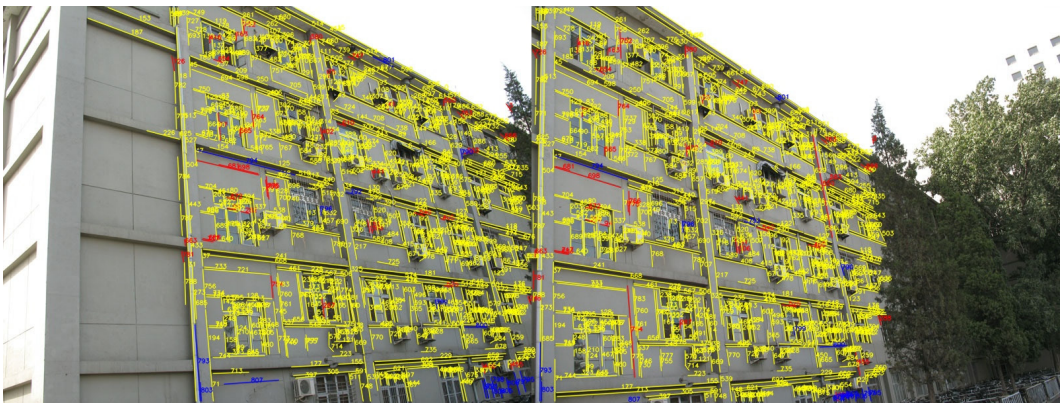

Figure 1. An example of employing the benchmark to evaluate the LSM result obtained by the method, LJL [33]. For each pair of line segment correspondences, the two line segments from the two images are labeled with a same unique number at the middles. The obtained line segment matches are drawn in yellow if they are correct, and in red otherwise. Those matches have not been found by the method, but exist in the ground truth, are drawn in blue.

yet, can be generated automatically. For example, Figure 1 visualizes the evaluation report of the line segment matching result obtained by the method, LJL [33]. It shows, in different colors, that among the 790 pairs of correspondences obtained by LJL, 749 of them are correct (in yellow), 41 are incorrect (in red). There are totally 845 ground truth matches in the two sets of line segments extracted from the two images using LSD, and therefore, 55 matches are missed (drawn in blue in this figure) by the method.

This benchmark can save a lot of time for algorithm developers in accessing the correctness of their obtained matches. They can hence focus only on improving their algorithms. Also, the benchmark can be used as the standard for evaluating different LSM methods.

The remaining parts of the paper are organized as follows. Section 2 presents a review of LSM methods. An introduction of the benchmark is presented in Section 3. A performance evaluation of some LSM methods is given in Section 4, followed by some discussions related the benchmark presented in Section 5. Finally, the conclusions are drawn in Section 6.

2. A Review of LSM Methods

We present in this section a review of LSM methods in the literature. We classify all the existing LSM methods into four categories according to the ways in which line segments are matched.

Individual line segment description based methods These category of methods match line segments through exploiting the photometric information associated with individual line segments, such as intensity [14, 16, 15], gradient [17, 18, 19, 20, 21], and color [22] in the local regions around them and generate a description vector for each of them. To match line segments by describing the local regions around them is not as easy as that for feature points because there is no point-to-point correspondence between corresponding line segments. The local regions of two corresponding line segments do not always overlap greatly with each other, leading to sometimes the great differences of the description vectors. Mean-Standard deviation Line Descriptor (MSLD) [17] tried to solve this problem by using the statistical values, mean and standard deviation, of gradients of pixels in the vicinity of a line segment since they are less sensitive to noises and image transformations. The statistical values for corresponding line segments are similar with each other even the endpoints of the two line segments do not precisely correspond. The later gradient-based methods [18, 19, 20, 21] more or less inherited this idea to make their line segment descriptors more robust. Another problem for this category of methods is that, like many local region descriptors for feature points, how to deal with the possible scale change between two images to be matched. As we know, the sizes of corresponding regions for two corresponding line segments from two images with scale change are different. Zhang and Koch [18, 19] proposed to solve this problem by detecting and describing line segments from image sequences generated by scaling down the two original images. If the image sequences contain enough images, there must be two images in which two corresponding line segments are described using the regions with the same sizes. In [20], the problem is solved by using nearly parallel line segments to extract the scale through exploiting the line segments detected on the lowest scale. The method is less robust in dealing with scale change than the image sequence based methods, but it is much faster since it involves only a single line segment detection and description for each image.

Line-point invariant based methods This category of methods leverage point matches for line segment matching [23, 24, 25, 26, 27]. The common idea of them is first to find

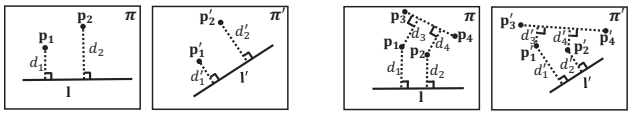

Figure 2. Two kinds of line-point invariants between coplanar points and line segments.

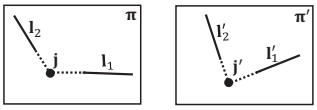

Figure 3. An illustration of the basic idea of the category of LSM methods which match line segments in pairs.

point matches using the existing point matching methods, and then to exploit invariants between coplanar points and line segment(s) under certain image transformations to evaluate the line segments from two images. The pairs of line segments which meet the invariants are regarded to be matches. There are two kinds of invariants between coplanar point and line segment(s): affine invariant and projective invariant. Figure 2 illustrates the two invariants. In Figure 2(a), ($\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{l}$) and ($\mathbf{p}'_1$, $\mathbf{p}'_2$, $\mathbf{l}'$) are the images of coplanar 3D entities projecting into two image planes. The distances of $\mathbf{p}_1$ and $\mathbf{p}_2$ to $\mathbf{l}$ are d_1 and d_2, respectively, and d_3 and d_4 for $\mathbf{p}'_1$ and $\mathbf{p}'_2$ to $\mathbf{l}'$. When the projection from 3D space to 2D images is approximately affine, there exists the relationship: $\frac{d_1}{d_2} = \frac{d'_1}{d'_2}$. This relationship is the so-called affine invariant. Please refer [24] to know how this relationship holds. The projective invariant establishes the relationship between the 2D images of two coplanar 3D points and two coplanar 3D line segments projecting into two image planes. It was initialized in [23], and was promoted in [25] to make it more practical for line segment matching. In [25], the authors used the line segment linking two 3D points as a second 3D line segment. Therefore, the projective invariant turns out to establish the relationship between the 2D images of four coplanar 3D points and one 3D line segment projecting into two image planes. Figure 2(b) gives an illustration of this invariant. Supposed the distances between 2D points and line segments are denoted as $d_{1\sim4}$ and $d'_{1\sim4}$, as shown in Figure 2(b), then there exists the relationship: $\frac{d_1}{d_3} : \frac{d_2}{d_4} = \frac{d'_1}{d'_3} : \frac{d'_2}{d'_4}$. Please refer [25] to get more details about this invariant.

Line segment pair based methods This category of methods match line segments in pairs by exploiting the intersecting junctions of adjacent line segments. The basic idea of this category of methods is illustrated in Figure 3. If the intersecting junction of line segments $\mathbf{l}_1$ and $\mathbf{l}_2$ from one image plane, $\mathbf{j}$ is successfully matched with $\mathbf{j}'$, the intersecting junction of line segments $\mathbf{l}'_1$ and $\mathbf{l}'_2$ from the other plane, it is then easy to determine the corresponding relationship between the two pairs line segments forming the two junctions. With this idea, Shahri *et al*. [31] match the junctions with the help of point matches obtained by SIFT, which provides both the global constraint, epipolar line constraint, and local constraint, local homography constraint, to guide the matching of pairs of line segments. The methods proposed by Kim *et al*. [29], based on their previous work presented in [30], has no requirement of the assistance from point matches obtained beforehand. It first normalizes the local patch formed by two adjacent line segments and their intersecting junction into a canonical frame and then evaluate the similarity of each two pairs of line segments from two images by calculating the NCC (Normalized Cross Correlation) score between their canonical frames. This method tends to produce false matches in regions with repetitive textures because the canonical frames of pairs of line segments in these regions are too similar to be distinctively distinguished by their NCC scores. Besides, the method does not deal with the line segments apart from others greatly that they can not be matched in pairs. The method proposed by Li *et al*. [33], based on their work in [32], avoids the disadvantages of the above two methods. It inherits the method proposed in [29] by describing the junctions formed by two adjacent line segments to get point matches. However, instead of normalized the local patches around the junctions, a more robust SIFT-like descriptor was proposed to describe the junctions. Besides, for those left line segments which are separated with others and can not be used to form junctions, the method evaluates them by local planar homographies estimated from neighboring junction and line segment correspondences identified before.

Line segment cluster based methods This category contains only one method reported in [28]. In this method, the extracted line segments lying adjacently are first clustered into groups, called line signatures, based on their saliency values, *i.e*., the average gradient magnitude of all pixels in a line segment. This grouping strategy enables a line signature in one image will appear in the other image in a high possibility after transformations. To evaluate the similarity of two line signatures from two images, the authors exploit the fact that the relative positions of line segments in a local region will be fairly stable under various image transformations to calculate a description about the line segment distribution for each line signature. Refer to Figure 4(a), in this line signature, based on the line segment with the highest saliency, the center line segment $\mathbf{l}_0$, the method calculates the relative positions of all the other line segments, $\mathbf{l}_{1\sim5}$. The number of other line segments except the center line segment in a line signature is called the rank of the line signature. It is five for the line signature shown in Figure 4(a). The position of a line segment relative to another is represented by a series of angles and length ratios. Refer to Figure 4(b), where the position of $\mathbf{l}_1$ relative to $\mathbf{l}_0$ is deno-

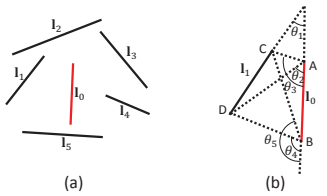

Figure 4. An illustration of the ideas of the LSM method proposed in [28].

ted as the vector: $D_1 = (\theta_{1\sim5}, r_{1\sim5})$, where $r_1 = \frac{|CD|}{|AB|}$, $r_2 = \frac{|AC|}{|AB|}$, $r_3 = \frac{|AD|}{|AB|}$, $r_4 = \frac{|BC|}{|AB|}$ and $r_5 = \frac{|BD|}{|AB|}$. This description vector can incorporate other information, such as saliency value ratio of $\mathbf{l}_1$ and $\mathbf{l}_0$, to strengthen the discriminative power. In this way, the spatial distribution of all other line segments in this signature is denoted as: $\mathcal{D} = \{D_i\}_{i=1}^5$.

Suppose $\mathcal{D} = \{D_i\}_{i=1}^m$ and $\mathcal{D}' = \{D'_j\}_{j=1}^m$ are the descriptions of two line signatures, $\mathcal{L}$ and $\mathcal{L}'$, from two images, where m is the rank of line signatures. Since there are multiple line segments in $\mathcal{L}$ and $\mathcal{L}'$, and the corresponding relationship between individual line segments are unknown, there are various mapping ways between individual line segments in $\mathcal{L}$ and $\mathcal{L}'$. The only correct mapping way, $\mathcal{M}$, indicates the true similarity of the two line signatures and minimizes the distance:

$$d = \sum_{(i,j)\in\mathcal{M}} |D_i - D'_j|, \tag{1}$$

where $|D_i - D'_j|$ refers the distance between the i-th vector in $\mathcal{D}$ and the j-th vector in $\mathcal{D}'$. The authors proposed to use a cookbook strategy to find the best mapping way. For more details about the method, please refer the paper [28].

3. Benchmark Dataset

After a comprehensive consideration of factors that affect the performance of LSM methods, namely the transformations between images, the types of scenes captured, the methods used for extracting line segments and the richness of straight line features, we finally selected the 15 pairs of images shown in Figure 5 and two state-of-the-art line segment detection methods, LSD [34] and EDLines [35], for establishing the benchmark. Thus, the benchmark provides the ground truth matches among 30 pairs of line segment sets. All these images were once used in the literature [18, 25, 36, 31, 33] and contains all common image transformations, namely, illumination change, rotation change, JPEG compression, image blur, scale change and viewpoint change. For some of them, we selected multiple pairs of images. Specially, for viewpoint change, we selected image pairs both with short baselines and wide baselines. Besides, the benchmark contains several image pairs characterized by the special scenes captured: namely, poorly-textured scene and occlusion scene. Also, both images capturing planar scenes and non-planar scenes are included to avoid the benchmark being biased towards only planar scenes.

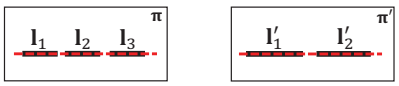

Figure 6. An illustration of multiple-to-multiple corresponding relationship between line segments from two images.

The only reliable method for finding the ground truth of line segment matches between two real images is through one-by-one visual inspection. In [18, 20], the authors evaluated their LSM results automatically on images related by a global homography by projecting the two endpoints of each line segment to the other image using the global homography and checking whether the identified correspondence lies in the very close region of the projected line segment. There are three problems in this strategy. The first is obvious that it can not be applied to images that are not related by a global homography. The second is that this strategy may give wrong judgments when several line segments crowd in a small region, which often appears in richly textured regions. Third, this strategy can only judge whether a match is a possibly correct one but can not find the ground truth line segment matches in the two images. These factors make visual inspection the only reliable way to set up such a benchmark. But the global homography still played a role in establishing the benchmark, at least for some image pairs. It was used to locate the rough places where correspondence(s) of a line segment lie, which facilitated making the benchmark.

To increase the reliability of the benchmark and limit false judgments as much as possible, three individuals worked on a same image for two times and generated six versions of the ground truth, and the final version is their optimal combination.

Unlike point matching where only one-to-one correspondence is allowed for two sets of points, in LSM, multiple-to-multiple correspondence is allowed, *i.e.*, a line segment from one image can be the correspondence of several line segments, or as one of the correspondences of a line segment from the other image. This is because a long line segment can be fragmented into several parts when detecting them. Figure 6 gives an illustration of this situation:a long segment in the first image plane is fragmented into three short line segments, $\mathbf{l}_{1\sim3}$. Meanwhile, its correspondence in other image plane is fragmented into two small line segments, $\mathbf{l}'_{1\sim2}$. The corresponding relationship established between any one in $\mathbf{l}_{1\sim3}$ and any one in $\mathbf{l}'_{1\sim2}$ is correct. Therefore, the ground truth matches for each two line segment sets extracted in a image pair using a line segment detector form such a set: $\mathcal{G} = \{(\mathcal{M}_v, \mathcal{M}'_v)\}_{v=1}^V$, where V denotes the number of elements in $\mathcal{G}$. $\mathcal{M}_v$ and $\mathcal{M}'_v$ are the v-th

(a). Illumination change. (b). Illumination change. (c). Rotation change.
(d). Rotation change. (e). JPEG compression. (f). Image blur.
(g). Poorly-textured scene. (h). Poorly-textured scene. (i). Occlusion.
(j). Scale change. (k). Scale and rotation change. (l). Viewpoint change.
(m). Viewpoint change & poorly-textured scene. (n). Viewpoint change (wide-baseline). (o). Viewpoint change (wide-baseline).

Figure 5. The 15 image pairs, (a)-(o), used in the benchmark dataset.

	LSD		EDLines			LSD		EDLines			LSD		EDLines	
	(N_1, N_2)	G	(N_1, N_2)	G		(N_1, N_2)	G	(N_1, N_2)	G		(N_1, N_2)	G	(N_1, N_2)	G
(a)	(971, 504)	412	(944, 448)	353	(f)	(1712, 450)	364	(1615, 248)	167	(k)	(1334, 569)	179	(1264, 441)	144
(b)	(572, 275)	224	(481, 235)	172	(g)	(102, 82)	58	(71, 67)	43	(l)	(1071, 1016)	811	(1008, 962)	732
(c)	(537, 556)	402	(407, 403)	306	(h)	(101, 98)	59	(72, 82)	44	(m)	(197, 366)	106	(127, 314)	71
(d)	(526, 398)	333	(412, 358)	282	(i)	(537, 368)	177	(389, 287)	120	(n)	(1007, 999)	424	(996, 899)	399
(e)	(590, 1083)	360	(569, 896)	322	(j)	(374, 681)	70	(354, 589)	59	(o)	(508, 553)	274	(375, 454)	322

Table 1. Some essential information about the 15 image pairs used in the benchmark. N_1 and N_2 denote the numbers of line segments detected in a pair of images using a certain line segment detector, and G denotes the number of ground truth matches in (N_1, N_2).

pair of line segment sets with one or multiple elements from the corresponding images. The total number of matches in $\mathcal{G}$ hence equals $\sum_{v=1}^{V} min(size(\mathcal{M}_v), size(\mathcal{M}'_v))$, where $size(\cdot)$ denotes the number of elements in a set. Table 1 presents the essential information of the benchmark.

4. Experimental Results

With the benchmark, we conducted experiments to evaluate the performances of some representative LSM methods. Since the established benchmark provides only the ground truth matches among the line segments extracted by certain line segment detectors (LSD and EDLines), we employed for experiments only those methods which have no special requirements on the extracted line segments. Three methods were selected: the line segment description based method, MSLD [17]; the line-point invariant based method, LPI [24]; and the line segment pair based method, LJL [33]. Each of the three methods achieved the state-of-the-art performance in their own category, and we want to know their comparative performances on the established benchmark. The implementations of the three methods were provided by their authors. We did not change anything but replaced the input images and line segments from the benchmark. We had also got the executable binaries of LS [28], the representative of the line segment cluster based LSM methods, but since we could not replace its input line segments, we could not evaluate it on the benchmark.

4.1. Line Segment Matching Results

We evaluated different LSM methods on our established benchmark by computing the three measures: $recall$ (R),

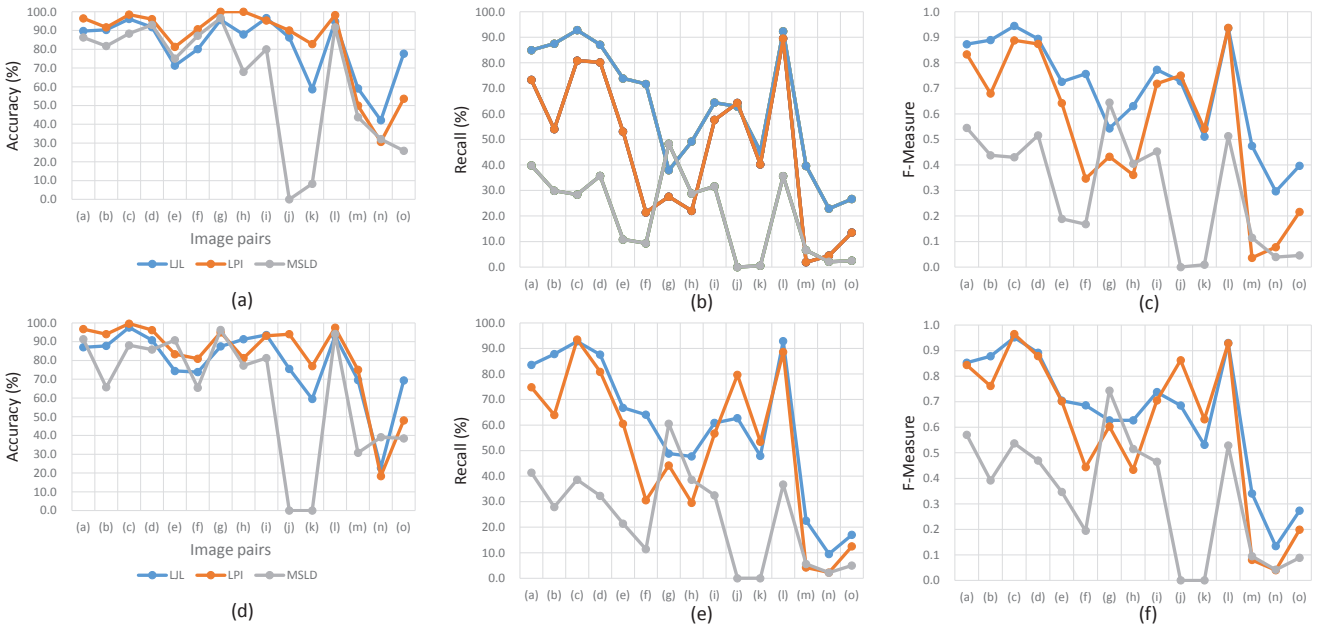

Figure 7. The performance of three LSM methods: LJL [33], LPI [25] and MSLD [17] on the established benchmark. The top row shows the results when using input line segments extracted by LSD [34], while the bottom row shows the results when EDLines [35] was the line segment detector.

i.e., the ratio of the number of correct matches and the ground truth matches; *accuracy* (A), *i.e.*, ratio of correct match and the total obtained matches; and $F - Measure(F) = \frac{2RA}{R+A}$. The experimental results are shown in Figure 7. From this figure, we can conclude how different LSM methods are affected by some factors.

Line segment detectors The experimental results of the three methods using line segments extracted by LSD and EDLines are shown in the first and second rows of Figure 7, respectively. The curves for a certain measure and a certain method w.r.t. the two line segment detectors behave similar for all image pairs. For example, the curves for LPI, drawn in orange, for all the three measures, when using line segments extracted by LSD and EDLines, behave in a similar trend when varying the test image pairs. This is because both LSD and EDLines are the state-of-the-art line segment detectors. They produced similar line segment detection results for a same image, as evidenced by the similarity of the two detectors in the numbers of the extracted line segments and numbers of the ground truth matches for same images shown in Table 1. Thus, with similar line segments as input, it is reasonable that the methods produced similar LSM results. For this reason, as we present how other factors affect the LSM results later, we base only on the results using the line segments detected by LSD, *i.e.*, the diagrams shown in the first row of Figure 7.

Image transformations Compared with scale change and viewpoint point change with wide-baseline, illumination change ((a) and (b)), rotation change ((c) and (d)), image blur ((e)) and JPEG compression ((f)) and short-baseline viewpoint change ((l)) are easy to be coped. All the three methods, therefore, achieved their own best performances among these image pairs. Comparatively, in view of accuracy, LPI has a slight advantage over the other two methods; LJL is better than MSLD in most cases. As for recall, the advantage of LJL over the other two methods is more remarkably, resulting in its highest F-Measures among the three methods. LJL is overall better than MSLD for both the two measures. For more challenging image transformations, scale change ((j) and (k)) and viewpoint change with wide-baselines ((n) and (o)), none of the three method produced satisfactory results. Though LPI and LJL produced results in (j) with relatively high accuracies, nearly 90%, the recalls of the both methods are only slightly above 60%. Comparatively speaking, LPI is better than LJL for scale change, while LJL outperforms LPI w.r.t. viewpoint point change with wide-baseline. In both situations, MSLD produced the worst results. This is because the fixed size window based line segment descriptor applied in MLSD is essentially unable to deal with both these two extreme situations.

Poorly-textured scene Three image pairs, (g), (h) and (m), are characterized by poorly-textured scenes captured. All the three methods evaluated are supposed to be badly affected by this special type of scenes. The line-point invariant based method, LPI requires point matches to help the matching of line segments. However, point matches are hard to be obtained in this kind of scenes. The good per-

formance of LJL can be guaranteed only when the epipolar geometry between the two images to be matched can successfully be recovered from the putative junction point matches. Otherwise, the two subsequent stages existing in this method aiming to refine the raw result can not reliably proceed. With only small numbers of extracted line segments, it is hard to generate sufficient junctions and accordingly sufficient junction matches. MSLD has no demand for point matches, but the deficiency of signal variances in poorly-textured scenes compromises the discriminative power of the descriptor. From Figure 7(a)-(c), though the three methods have high accuracies in (g) and (h) (except MLSD in (h)), the recalls are very low for all of them. Remarkably, LPI has the highest accuracies in both (g) and (h), up to 100%, but it has the worst recalls. This means LPI produced only small numbers of line segment matches, but all of them are correct. All the three methods have terrible performances in (m). The causes are not only the poor textures of the scene, but also the illumination change and fairly great viewpoint change.

Occlusion The scene captured in the pair of images in (i) is occluded by the shades of twigs, but in different manners in the two images. The detected line segments will be interrupted by the shades, resulting in some fragmented short line segments with their endpoints lying in the shades. Hence, line segment correspondences from the two images do not have even nearly corresponding endpoints, which might cause bad performances of some methods relying on it. Among the three methods employed for comparison, MSLD relies on the most for the approximate correspondence of the endpoints of corresponding line segments for matching them since it needs to generate a description for line segments. The other two methods do not have such requirement. It is thus expected that MSLD will perform the worst in this image pair. The experimental results substantiate the expectation, and MSLD produced the worst results with respect to all the three measures. LJL outperforms LPI slightly on all the three measure, and both the two methods generated results with very high accuracies (over 95%) and relatively low recalls (around 60%).

4.2. Running Time

The running time of LSM methods is closely correlated with the number of input line segments. For the line segment description based method, MSLD, the majority of time is spent on describing each individual line segments from both images. For the line-point invariant based method, LPI, the running time consists of the time costs of point match generation, line-point variant construction and testing. For line segment pair based method, LJL, the time is spent on line-junction-line structure construction, description, matching and refining; and individual line segment grouping and matching. Table 2 shows the elapsed time of these methods on the image pairs (g) and (l) with line segments detected by LSD. The most and least numbers of line segments were detected in these two image pairs by LSD among all the 15 image pairs in the benchmark. From the elapsed time of the three methods on these two extreme cases, we can know the comparative efficiencies of the three methods and how their efficiencies are affected by the numbers of the extracted line segments. As we can see from Table 2, MSLD has an overwhelming advantage in the efficiency both with great and small numbers of line segments. When with a small number of input line segments, LPI took the most time. This is because LPI must spend time generating point matches, which can be a time-consuming task especially when the used images are large. As the number of input line segment increases, the time cost of LJL increases tremendously and is several times than that of LPI. This is because as the number of input line segments grows, the number of the constructed line-junction-line structures increased swiftly, requiring more time to match them to get line segment matches.

	LJL	LPI	MSLD		LJL	LPI	MSLD
(g)	1.1	14.5	0.5	(l)	348.0	94.5	0.3

Table 2. The elapsed time (in seconds) of three LSM methods with input image pairs (g) and (l), shown in Figure 5, and line segments extracted by LSD.

4.3. General Conclusions

Despite that the three methods vary greatly for their performances on the benchmark, we can still reach some general conclusions based on the experimental results. LJL produced generally the best results, with highest recalls and F-Measures in most cases, and higher accuracies for some severe image transformations. LPI produced line segment matches with the highest accuracies for slight and moderate image transformations. MSLD produced the worst results in most cases, especially for severe image transformations. As for efficiency, MSLD has an overwhelming advantage over the other two methods. LJL is more efficient than LPI for small numbers of input line segments, but becomes less efficient when the numbers of input line segments are great.

5. Discussions

We present in this section some discussions about using the benchmark and some explanations for the evaluation results presented above.

5.1. Experimenting with Only Long Line Segments

Short line segments are more sensitive to noises in the images and are hence often unstable and not precisely located. This fact makes short line segments hard to be matched, and cause them bringing inferences for matching

long line segments. Therefore, many LSM methods select only long line segments for matching. The established benchmark does not make such selection and provides the ground truth matches among all detected line segments. The experiments presented in the last section were also based on all detected line segments. We kept all extracted line segments for establishing the benchmark for these reasons. First, different LSM methods vary their limitations on the length of line segments. If we choose a certain length threshold for making the benchmark, it may be favored by some methods but not for others, making the benchmark biased towards certain methods. Second, the ability of dealing with the unstableness of short line segments is also an indicator of the robustness of a LSM method. Though the benchmark is based on all extracted line segments, it can be exploited to evaluate the LSM methods which have limitations on the length of the extracted line segments. Researchers need just remove those ground truth matches containing line segments which violate the length constraint and use the trimmed ground truth for evaluating their results.

5.2. Explanations for the Inconsistent Results

The experimental results reported in this paper for all the three methods (LJL, LPI and MSLD) are quite worser than that presented in the original papers, even with the same image pairs and the same input line segments. This discrepancy might result from that the versions of implementations of the algorithms used by us are different from the authors', or the mistakes made by the authors–also by us, though less likely–when accessing the correctness of the obtained results. Accessing the correctness of hundreds of line segment matches through visual inspection is an error-prone work. The authors, maybe including us though we had tried our best to avoid this problem by accessing a same pair of line segment sets with different individuals for multiple times when we made this benchmark, may inevitably make some false judgments. However, we found, when we conducted experiments, the major reason for the inconsistent results lies in our rigor in accepting a pair line segments as a correct match in the benchmark. That is, some line segment matches which were accepted as correct ones by the authors, were regarded as false ones in the benchmark. This is not because the authors are not sufficiently rigorous in checking their obtained results, but because their ways of accessing the obtained results naturally tends to cause false judgments. The following example shows how this comes.

Suppose an author had processed the two images shown in Figure 8(a) and Figure 8(b) using his method. Then, he visualized the obtained line segment matches to access their correctness through visual inspection. When he came to the match (#10, #23). As almost everybody did, we bet, he regarded this match as a correct one. However, when we

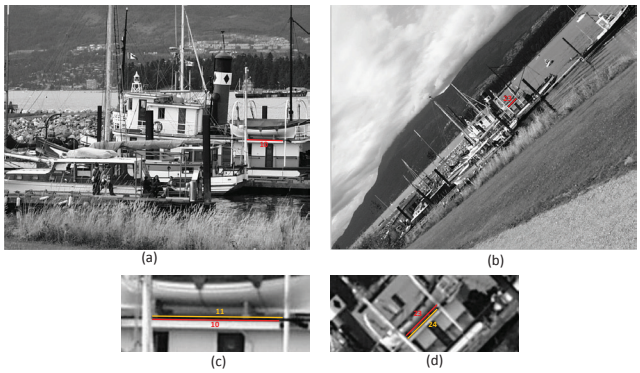

Figure 8. An example showing why some line segment matches that were regarded to be correct by previous papers but were regarded to be false in the established benchmark.

established the ground truth line segment matches between these two images, we would not include this match into the ground truth match set. This is because when we found the ground truth matches between these two images, for each line segment in one image, we found its correspondence in the context of a group of adjacent line segments and pick out the best one. This process of finding matches among two groups of line segments is quite different with the process of checking the correctness of the obtained matches, in which only the obtained line segment correspondences are shown, but the contexts where other neighboring line segments present are neglected. For example, when we tried to find the correspondence of line segment #10 in Figure 8(a), we will zoom in the local region around it, as shown in Figure 8(c). Next, we would zoom in the related region in the other image where the possible correspondence of #10 lies, obtaining the view shown in Figure 8(d). With the context of neighbor line segments, it is easy to determine that the best correspondence for #10 is #24, not #23, and we added (#10, #24) into the ground truth. We found in experiments that such cases are not rare, especially for two images with great scale change, in which the distances between neighbor line segments are narrowed after zooming out.

6. Conclusions

In this paper, we have introduced a benchmark for line segment matching. The benchmark contains the ground truth matches among 30 pairs of line segment sets extracted for 15 representative image pairs using two line segment detectors. The benchmark can solve two major problems in this area: no unbiased standard for evaluating different methods and the difficulty of accessing the correctness of the obtained match automatically and reliably. We experimented several representative line segment matching methods on the benchmark and report the results in this paper. Based on the experimental results, we have analyzed the advantages and disadvantages of these methods.

References

[1] M. Chandraker, J. Lim, and D. Kriegman. Moving in stereo: Efficient structure and motion using lines. In *ICCV*, 2009. 1

[2] A. Bartoli and P. Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, 100(3):416-441, 2005. 1

[3] S. Ramalingam and M. Brand. Lifting 3D Manhattan lines from a single image. In *ICCV*, 2013. 1

[4] M. Hofer, M. Maurer, and H. Bischof. Improving sparse 3D models for man-made environments using line-based 3D reconstruction. In *3DV*, 2014. 1

[5] M. Hofer, M. Donoser, and H. Bischof. Semi-global 3D line modeling for incremental structure-from-motion. In *BMVC*, 2014. 1

[6] A. Jain, C. Kurz, T. Thormahlen, and H. P. Seidel. Exploiting global connectivity constraints for reconstruction of 3D line segments from images. In *CVPR*, 2010. 1

[7] G. Schindler, P. Krishnamurthy, and F. Dellaert. Line-based structure from motion for urban environments. In *3DPVT*, 2006. 1

[8] H. Bay, A. Ess, A. Neubeck, and L. Van Gool. 3D from line segments in two poorly-textured, uncalibrated images. In *3DPVT*, 2006. 1

[9] S. Ramalingam, M. Antunes, D. Snow, G. Hee Lee, and S. Pillai. Line-sweep: Cross-Ratio for wide-baseline matching and 3D reconstruction. In *CVPR*, 2015.

[10] B. Micusik and H. Wildenauer. Descriptor free visual indoor localization with line segments. In *CVPR*, 2015. 1

[11] P. Smith, I. D. Reid, and A. J. Davison. Real-time monocular SLAM with straight lines. In *BMVC* 2006. 1

[12] L. Zhang, C. Xu, K. Lee, and R. Koch Robust and efficient pose estimation from line correspondences. In *ACCV* 2012. 1

[13] B. Přibyl, P. Zemčík, and M. Čadík Camera pose estimation from lines using plücker coordinates. In *BMVC* 2015. 1

[14] C. Schmid and A. Zisserman. Automatic line matching across views. In *CVPR*, 1997. 2

[15] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs In *ECCV*, 2002. 2

[16] C. Baillard, C. Schmid, A. Zisserman, and A. Fitzgibbon. Automatic line matching and 3D reconstruction of buildings from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, 1999. 2

[17] Z. Wang, F. Wu, and Z. Hu. MSLD: A robust descriptor for line matching. *Pattern Recognition*, 42(5):941–953, 2009. 2, 5, 6

[18] L. Zhang and R. Koch. Line Matching Using Appearance Similarities and Geometric Constraints. In *ACCV*, 2012. 2, 4

[19] 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*, 24(7):794-805, 2013. 2

[20] B. Verhagen, R. Timofte, and L. Van Gool. Scale-invariant line descriptors for wide baseline matching. In *WACV*, 2014. 2, 4

[21] K. Hirose, and H. Saito. fast line description for line-based SLAM. In *BMVC*, 2012. 2

[22] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *CVPR*, 2005. 2

[23] M. I. Lourakis, S. T. Halkidis, and S. C. Orphanoudakis. Matching disparate views of planar surfaces using projective invariants. *Image and Vision Computing*, 18(9):673–683, 2000. 2, 3

[24] B. Fan, F. Wu, and Z. Hu. Line matching leveraged by point correspondences. In *CVPR*, 2010. 2, 3, 5

[25] B. Fan, F. Wu, and Z. Hu. Robust line matching through line-point invariants. *Pattern Recognition*, 45(2):794–805, 2012. 2, 3, 4, 6

[26] M. Chen, and Z. Shao. Robust affine-invariant line matching for high resolution remote sensing images. *Photogrammetric Engineering & Remote Sensing*, 79(8):753–760, 2013. 2

[27] D. S. Ly, C. Demonceaux, R. Seulin, and Y. Fougerolle. Scale invariant line matching on the sphere. In *ICIP*, 2013. 2

[28] L. Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *ICCV*, 2009. 3, 4, 5

[29] H. Kim, S. Lee, and Y. Lee. Wide-baseline stereo matching based on the line intersection context for real-time workspace modeling. *Journal of the Optical Society of America, Optics, Image Science, and Vision*, 31(2):421–435, 2014. 3

[30] H. Kim and S. Lee. Simultaneous line matching and epipolar geometry estimation based onthe intersection context of coplanar line pairs. *Pattern Recognition Letters*, 33(10):1349–1363, 2012. 3

[31] M. Al-Shahri and A. Yilmaz. Line matching in wide-baseline stereo: A top-down approach. *IEEE Transactions on Image Processing*, 23(9):4199-4210, 2014. 3, 4

[32] K. Li, J. Yao, X .Lu. Robust line matching based on ray-point-ray structure descriptor. In *ACCV Workshop on Robust Local Descriptors for Computer Vision* , 2014. 3

[33] 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. In *Neurocomputing*, 2015. 1, 2, 3, 4, 5, 6

[34] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(4):722–732, 2012. 1, 4, 6

[35] C. Akinlar and C. Topal. EDLines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633–1642, 2011. 4, 6

[36] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1-2), 43-72, 2005. 4