

Indoor Calibration using Segment Chains

Jamil Drareni, Renaud Keriven, Renaud Marlet

▶ **To cite this version:**

Jamil Drareni, Renaud Keriven, Renaud Marlet. Indoor Calibration using Segment Chains. 33rd annual pattern recognition symposium of the German Association for Pattern Recognition (DAGM), Aug 2011, Frankfurt/Main, Germany. pp.71-80, 10.1007/978-3-642-23123-0_8 . hal-00825881

HAL Id: hal-00825881
https://enpc.hal.science/hal-00825881

Submitted on 24 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Indoor Calibration using Segment Chains

Jamil Draréni, Renaud Keriven, and Renaud Marlet

IMAGINE, LIGM, Université Paris-Est, FRANCE.

Abstract. In this paper, we present a new method for line segments matching for indoor reconstruction. Instead of matching individual segments via a descriptor like most methods do, we match segment chains that have a distinctive topology using a dynamic programing formulation. Our method relies solely on the geometric layout of the segment chain and not on photometric or color profiles. Our tests showed that the presented method is robust and manages to produce calibration information even under a drastic change of viewpoint.

1 Introduction

Many tasks in computer vision[1] , such as structure from motion, expect a set of features matched across images to register cameras in a common coordinate system. For decades, corner detectors such as Harris and KLT detectors represented the *de facto* features in computer vision literature. Recently, a new breed of features appeared in the literature. Pioneered by the seminal work of David Lowe [9], this new generation of feature detectors brought two major capabilities that lacked in the previous generation: geometric invariance and a descriptor. The first aspect is often achieved by using a scale-space framework [8] and data normalization, whereas the addiction of a feature descriptor yield a better matching repeatability.

However, in low texture scenes such as man-made environments (Fig.1), feature point perform poorly. In such scenes, 2D features run short in favor of line segments and yet, they have not been used extensively as features for image registration because of the difficulty of matching them. The later stems from the fact that, segments vicinity are textureless and usually not distinctive which precludes histogram-based (gradient, color, ...) description. Even if a segment vicinity shows some variety, defining a neighbourhood zone is not tractable because a segment does not bear a natural scale, as it's the case with feature points. Last but not least, segment's endpoints are seldom accurate thus, using them as distinctive features is often error-prone.

In [3] authors used Kalman filters to track line segments. Their formulation is akin to features tracking in video sequences and assumes short baselines.

Schmid and Zisserman [10] used epipolar lines bundles to constrain putative matches followed by an intensity similarity check (using SSD along the segments)

[1] This work was partially supported by the Fondation d'entreprise EADS, contract no 3610.



Fig. 1. A pair of indoor images used in this paper.

to identify correct matches. Though fast and accurate, their method assumes a known epipolar geometry which makes it unsuitable for camera calibration. In [1], a segments matching method with an application to wide baseline stereo is presented. Here, feature points (SIFT, HOG,...) are extracted and used as anchor points. The actual segments pairing is performed by first, grouping putative matches using their color profiles.Then, anchor-point/segment sideness consistency is exploited to sort out the matches. Obviously, if no or few feature points were detected and paired, the segment matching will rely solely on color profiles which are known to be unstable. The idea of using supporting 2D features was also proposed for segments matching in [5] and more generally for untextured regions matching in [4].

In [11], the authors proposed an approach where segments are assigned a descriptor computed from the layout of neighbouring lines (length ratios, relative angles). As pointed out in [5], line signatures are subject to instabilities because they rely on segments endpoints locations which are known to be inaccurate.

In this paper, we present a new method for segments matching. The proposed method is suitable for camera calibration in an indoor environment and is based on the fact that indoor scenes are often composed of segment chains. While individual segments might not exhibit saliency, we show how segment chains topology have a distinctive layout that can be exploited for matching. In this work, segment chains are extracted and matched using dynamic programming. The result is then used to compute the epipolar geometry induced by the camera motion.

The rest of the paper is organized as follows. Section 2 presents the algorithm used for segments detection. The core of our algorithm is detailed in Section

3 while experimental results are reported in Section 4. Finally, we draw our conclusion and discuss further improvement Section 5.

2 Segments Extraction

There are numerous ways to extract line segments in images. The classical approach consists of applying the Canny detector [2] to the image to extract edges. These edges are then clipped at high curvature points. Along the curve that connects two high curvature points, a line is fitted by either a least squares method like in [1, 5] or using a robust method such as the M-estimator technique. In this paper, we used the Line Segment Detector (LSD) recently introduced in [6]. We would like to point out that our matching procedure is independent of the underlying line detector and conveniency was the main motivation behind this choice since an implementation of LSD is freely available on the author's web page. In the following we will summarize the LSD detector.

As depicted on Figure.2, line extraction starts by computing the edge map of the image. Then, starting from a random position, line-support regions are formed by grouping adjacent pixels that share the same gradient orientation within a certain tolerance. The formation of the line-supports is done recursively in a flood-fill fashion. Even though this procedure is based on a greedy algorithm, the simple operations behind it makes it very fast. As a result, LSD outputs the coordinates of the each segment endpoints along with the width of the associated line-support.

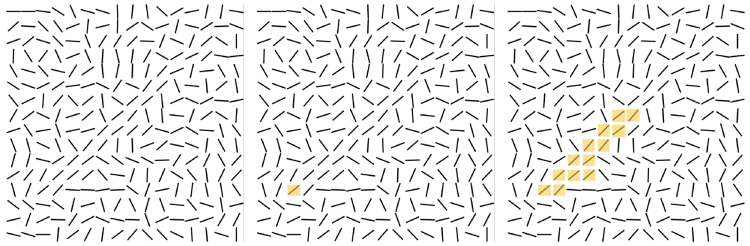


Fig. 2. The Line Segment Detector (LSD). Line support regions are gathered recursively if they fall within a certain orientation tolerance.

Once the segments are extracted, an orientation will be assigned to each of them according to the direction of their supporting gradient. Namely, a segment will be oriented such as the darkest region will always lie on its right side. The purpose of the orientation assignment is to form consistent segments chain as it will be explained later. It could also be used to leverage matching ambiguities when cameras orientations are available[2].

[2] Most modern cameras provide this information in the image header.

2.1 Segments Merging

Segments detected by LSD are often fragmented because of noise. In order to simplify the matching step, we post-process LSD output by merging the segments fragments that share the same orientation (up to a tolerance) and whose endpoints are close. This is illustrated in Fig.3

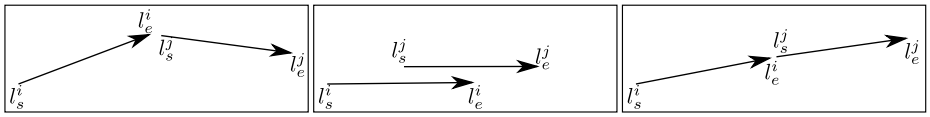


Fig. 3. Segments merging. Segments i and j are identified by their start and endpoint. Left) No merging is done here because angle difference is large. Middle) No merging neither because the distance between endpoints ($||l_e^i - l_s^j||$) is large. Right) Merging is done because the segments are well aligned and close to each other.

The result of this process is a set of neat segments more suitable for our matching procedure. As an exemple of such sets, running the merging procedure on the sequence in Fig.1 is depicted in Fig.4.

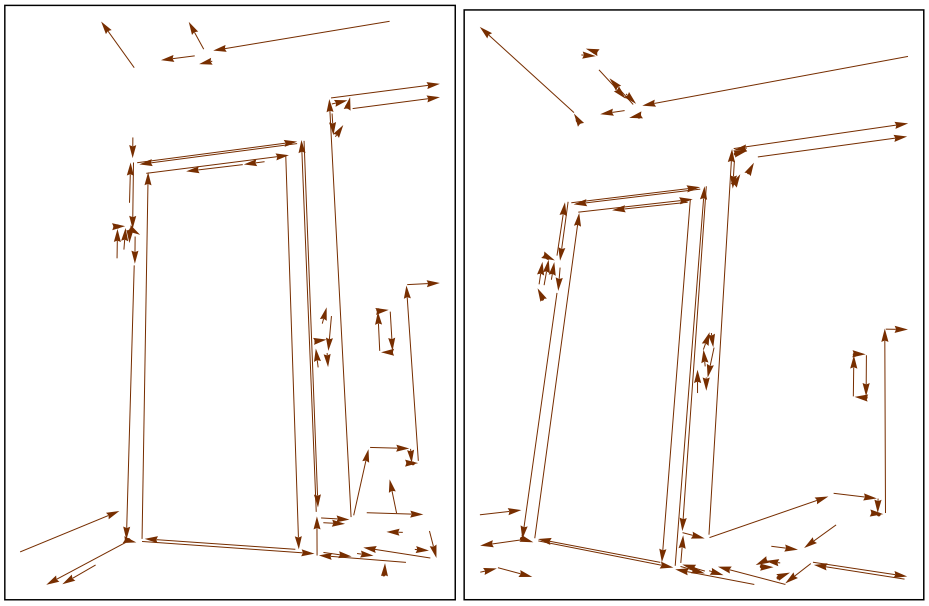


Fig. 4. LSD output after post-processing (see text).

3 Segment Chains Matching

In this section, we will give the details of our segment chain matching method. Starting from an image, we use LSD to extract segments and merge fragments as explained in the previous section. We then, form segments junctions by connecting salient segments in quasi-intersection situation. Such segments are most likely to be 3D-coplanar, thus their intersections are the projection of real 3D points that can be further used for camera calibration. At this stage, segments in T-junction configuration are avoided as they presage occlusions.

Starting from a junction that connects two salient segments, we form an initial 2-segment chain. This chain is recursively grown by adding continuous segments that preserve the overall orientation. Three such chains are depicted in Fig.5. Even though the extraction procedure is greedy, the running time is small thanks to the low cardinality of the segments present in indoor scenes.

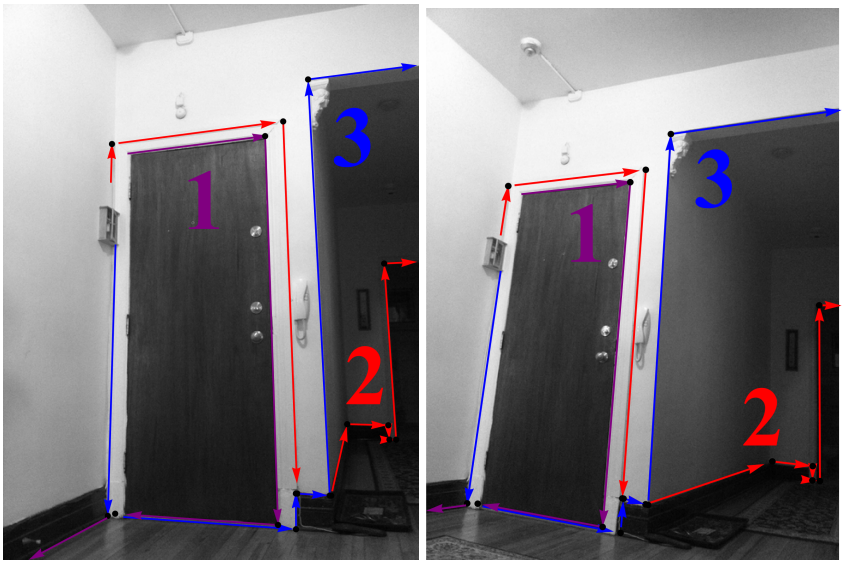


Fig. 5. Three extracted chains. The added segments must preserve the global orientation of the chain (clockwise or counterclockwise) and its continuity.

Once segments chains isolated in both images, matching can take place using a dynamic programing scheme presented in the following.

3.1 Dynamic Programing Formulation

Let C and C' be two segments chains to match, given as a list of their junction points. We define the cost of matching two junctions $j_k \in C$ and $j'_l \in C'$ in terms

of their incident angles $(\theta_{j_k}, \theta_{j'_l})$ as:

$$match(j_k, j'_l) = |\theta_{j_k} - \theta_{j'_l}| \tag{1}$$

The cost of skipping a junction j_k is given by the length of its inward segment:

$$skip(j_k) = ||j_k - j_{k-1}|| \tag{2}$$

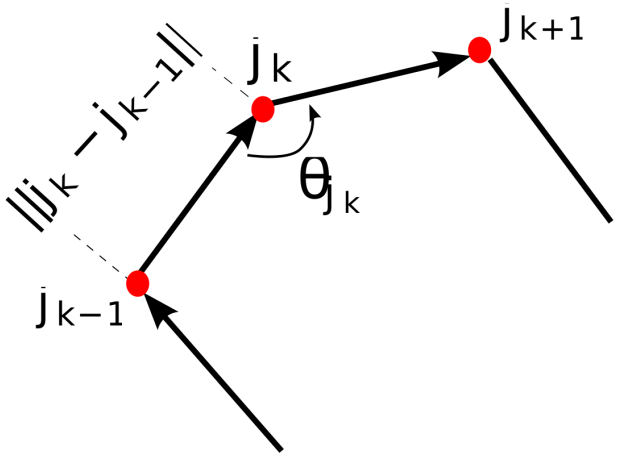


Fig. 6. The incident angle θ_{j_k} of a junction j_k and the length of its inward segment.

See Fig.6 for further explanations.

The optimal cost of matching the junction $j_k \in C$ with the junction $j' \in C'$ is formulated recursively as:

$$cost(j_k, j'_l) = match(j_k, j'_l) + \min \begin{cases} cost(j_{k-1}, j'_{k-1}) \\ cost(j_{k-1}, j'_l) + skip(j_k) \\ cost(j_k, j'_{l-1}) + skip(j'_l) \end{cases} \tag{3}$$

Because we do not know in advance the corresponding segment chains from an image to another, we simply cross-match all the chains using our dynamic programing formulation and assign matches using a winner-takes-all scheme. Because the number of chains is low, this procedure is fast. To show the effectiveness of this procedure, the cross-matching costs of the 3 chains depicted in Fig.5 is reported in Fig.7 .

3.2 Camera Calibration

Once segment chains matches have been determined, junctions matches are trivially extracted. However, theses matches are only putative because, unless they

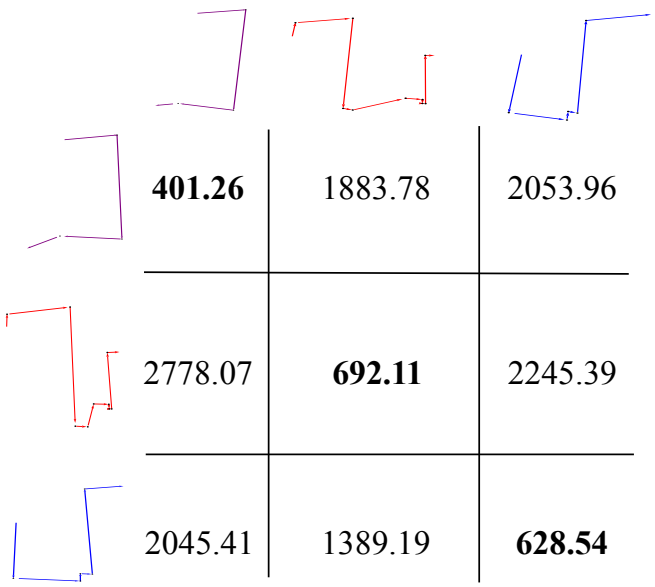


Fig. 7. Matching cost. The cost of cross-matching 3 different chains. We can see that real matches have a very low cost compared to other combinations which precludes a winner-takes-all scheme.

result from coplanar segments, they are not the projection of the same 3D points. To circumvent this, a RANSAC routine is used to randomly select 7-tuples from putative junction matches and robustly estimate the fundamental matrix as explained in [7]. Of course, in the calibrated case, one would prefer estimating the essential matrix in order to extract the rotations and translations of the cameras.

Our experiments showed that indoor scenes exhibits few outliers. Indeed, in such scenes, adjacent segments are likely to be coplanar, thus most junctions turn out to be real features.

4 Experiments

In this section we show the results of our experiments to demonstrate the effectiveness of the proposed method. All images were shot with a hand-held iPhone. The images were downscaled to a resolution of 1200×800. The estimated epipolar geometry from the first sequence (see Fig.1) is depicted in Fig.8. Notice the precision of the epipolar lines despite the camera motion.

We also ran our algorithm on a second sequence (see Fig.9). As it can be seen, this sequence is challenging due to the amount of occlusion and the lack of texture. The resulting epipolar geometry is shown on Fig.10.

The method was implemented on Mathematica and still, the running time for each sequence was under 5 seconds on a 2.13 Ghz computer.



Fig. 8. Estimated epipolar lines on the first sequence along with 2 segment chains (blue and red).

5 Conclusion

In this paper we presented a novel method to match line segments in an indoor environment. Instead of building descriptors per segment, we considered segment chains for their rich geometric topology that makes them very distinctive even in presence of occlusions and severe viewpoint change. We also presented a simple dynamic programing formulation to efficiently match such chains based solely on geometric properties. In fact, our method is completely intensity-blind as no photometric information was used to achieve the matching.

Moreover we think that our method could be successfully used to reconstruct interiors from images and furthermore to produce building outlines.

Finally, we are convinced that our system could benefit from advances in graph theory field to better explore the network of segments and efficiently extract meaningful chains. In fact, topics such as graph partitioning techniques are already in study.

References

1. H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 329–336, Washington, DC, USA, 2005. IEEE Computer Society.



Fig. 9. Second sequence. Top: Original images. Bottom: Detected segments after fragments removal.

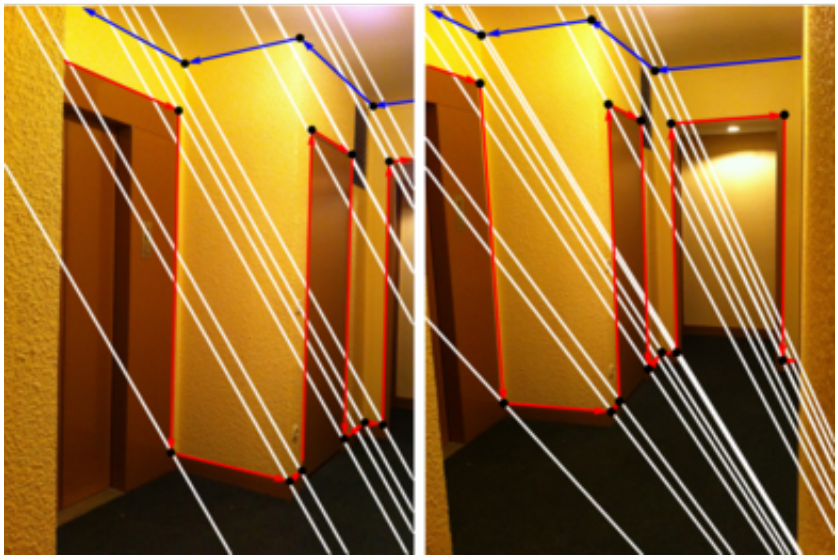


Fig. 10. Estimated epipolar lines and 2 segment chains from the indoor sequence.

2. J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:679–698, November 1986.
3. R. Deriche and O. D. Faugeras. Tracking line segments. In O. D. Faugeras, editor, *ECCV*, volume 427 of *Lecture Notes in Computer Science*, pages 259–268. Springer, 1990.
4. R. Dragon, M. Shoaib, B. Rosenhahn, and J. Ostermann. Nf-features - no-feature-features for representing non-textured regions. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV (2)*, volume 6312 of *Lecture Notes in Computer Science*, pages 128–141. Springer, 2010.
5. B. Fan, F. Wu, and Z. Hu. Line matching leveraged by point correspondences. In *CVPR*, pages 390–397. IEEE, 2010.
6. R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):722–732, 2010.
7. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004.
8. T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, Norwell, MA, USA, 1994.
9. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91, Nov. 2004.
10. C. Schmid and A. Zisserman. Automatic line matching across views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–671, 1997.
11. L. Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *Proc. International Conference on Computer Vision.* IEEE, 2009.