



D-LSD: A Distorted Line Segment Detector for Calibrated Images

David Zuñiga-Noël, Francisco-Angel Moreno^(✉), and Javier Gonzalez-Jimenez

Machine Perception and Intelligent Robotics Group (MAPIR), Department of System Engineering and Automation, University of Malaga, Malaga, Spain
{dzuniga,famoreno,javiergonzalez}@uma.es

Abstract. In this paper, we present an algorithm for the detection of line segments directly on the original, distorted images captured by calibrated wide-angle, fisheye and omnidirectional cameras. Distorted line segments are detected as convex polygonal chains of connected straight lines and then validated as the projection of 3D lines. This last validation step is our main contribution, which is formulated in a generic way in order to allow the detection of line segments from calibrated central projection vision systems and without requiring the rectification of the whole image. We evaluate our method with real images from a publicly available dataset and compare it with state-of-the-art alternatives, achieving comparable line detection performance without requiring image rectification. Additionally, we provide an open source reference implementation.

1 Introduction

Typical computer vision techniques rely on point image features to perform, for example, visual SLAM [19], camera motion estimation [20], place recognition [3] or object recognition [8], among many others. However, higher level geometric primitives, such as lines, are gaining growing importance in recent years, as they provide a set of advantages, specially in terms of reliability. These include their higher robustness against illumination changes in the image, as well as their natural presence in human-made environments [25]. Not only that, unlike point features, lines can be found even in images from low-textured environments, which is known to be one of the main issues of geometric computer vision techniques. This problem can be partially mitigated by employing fish-eye or omnidirectional cameras, which provide wider Field-of-View (FoV), hence covering larger areas of the environment and consequently reducing the probability of capturing low-texture areas, but it still remains an issue.

The benefits that line features provide have been exploited recently for the development of new systems employing them either as their main source of information or at least complementary to point features [5, 6, 14, 15, 17]. Nevertheless, detecting and managing line features is more costly than using their point counterpart, specially in wide FoV cameras where distortion is significantly larger than in standard perspective cameras, causing that straight lines in the environment are no longer projected as straight 2D lines on the images. To avoid

this, the images need to be rectified [11] in a process that presents two main issues: (i) it is computationally expensive, and (ii) the maximum effective FoV after rectification is reduced, in practice, to about 130° [7] instead of the original $\sim 180^\circ$ FoV for fish-eye and up to 360° for omnidirectional cameras. Our proposal provides a solution to these two problems.

In this paper, we present D-LSD (Distorted Line Segment Detector), the first line segment detector that enables direct detection of distorted lines from non-rectified images. We formulate the distorted line detection problem under the mild assumption of a calibrated central projection system, resulting in a very efficient line detector capable of handling from pinhole to omnidirectional camera projection models. We evaluate our approach with publicly available real images and compare the execution time and detection rates against state-of-the-art straight line detection methods on rectified images.

2 Related Work

The most popular line segment detector is LSD [16], by Grompone von Gioi *et al.* The LSD algorithm works by grouping pixels with the same gradient direction and validating lines as rare events in the *a contrario* model, according to the Helmholtz principle [10]. This method gained popularity as one of the first line detectors capable of operating in real-time.

Alternatively, in [1], Akinlar and Topal propose to detect lines from a continuous, 1 px-wide chain of edge pixels by least squares line fitting. These high quality edges are detected from their proposed Edge Drawing [23] algorithm. This approach, termed *EDLines*, results in a very efficient line detection algorithm (about one order of magnitude faster than LSD). A similar approach, named CannyLines, is proposed in [18] by Lu *et al.* where lines are detected directly on the edge map extracted by their parameter-free Canny operator.

Following a different approach, Cho *et al.* propose in [9] a new line segment detector exploiting the properties of digitalized lines. In this work, anchor points (*i.e.* peaks in the gradient map) are connected horizontally or vertically only, creating what the authors call *linelets*. Then these linelets are grouped into line segments according to some rules derived from the digitalization properties and validated from a probabilistic perspective. Recently, Zhang *et al.* proposed AG3line [24], which detects line segments by actively grouping anchor points. Finally, the detected lines are validated according to the density of anchors and the alignment of the gradient magnitudes.

Unlike our proposal, all these methods operate only in rectified images and cannot find straight lines in distorted images. In this work, we rely on the efficient Edge Drawing [23] algorithm for continuous edge segmentation and exploit the fact that distorted lines are locally straight to fit 2D line segments, similar to the EDLines [1] approach. Then, connected 2D line segments forming a smooth convex curve are grouped and validated as being the projection of a 3D line, according to the calibrated intrinsic parameters of the camera.

3 Background

The mapping of a 3D point $\mathbf{x} \in \mathbb{R}^3$ to image coordinates $\mathbf{u} \in \Omega \subset \mathbb{R}^2$ is characterized by the projection function $\pi : \mathbb{R}^3 \rightarrow \Omega$. In this work, we consider central projection cameras in which all light-rays pass through a single point in space: the projection (or optical) center of the camera. This includes both dioptric and catadioptric cameras.

In general, lines are projected as conic sections in any radially distorted image. Svoboda and Pajdla [22] showed that lines are imaged as conic sections in catadioptric systems. Accordingly, Barreto [4] showed that this is also true for perspective cameras with radially symmetric distortion following the division model [13]. Here, we only assume that lines are imaged as smooth convex curves, *i.e.* its curvature is either always non-negative or always non-positive, up to a *smoothness* threshold $\theta \in \mathbb{R}^+$ (see Fig. 1). Note that this is a typical assumption used, for example, to find circles [2] or ellipses [21].

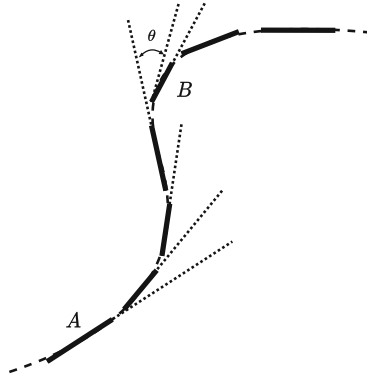


Fig. 1. Curves are approximated in a piecewise linear fashion, *i.e.* as polygonal chains. The polygonal chain starting at segment A breaks both the convexity and smoothness assumptions at B . Therefore, a new polygonal chain candidate starts at B .

Finally, under the central projection assumption, the two endpoints $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$ defining a three-dimensional line, along with the center of projection, define a plane with normal vector $\mathbf{n} \in \mathbb{R}^3$ (see Fig. 2). This fact, along with the inverse of the projection function $\pi^{-1} : \Omega \rightarrow \mathbb{R}^3$ will be used to generate a model of the observed curve. Note that the depth of a 3D point is lost during the projection process, thus π^{-1} can only recover its direction.

4 Algorithm Description

D-LSD operates in 4 main steps as follows. First, a set of continuous edges are extracted from the intensity image. Subsequently, smooth convex curves are

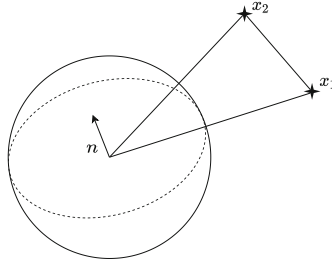


Fig. 2. In a central projection system, a 3D line and the center of projection form a plane. This plane intersects the unit sphere in the dotted circle. The line segment is thus an arc on the unit sphere.

extracted as polygonal chains from connected edges. A model of a 3D line is then extracted by means of a least squares minimization from the pixels on the curve and, finally, such curve is validated as a distorted line according to its reprojection error in the image.

4.1 Edge Detection

We rely on the Edge Drawing (ED) algorithm by Topal and Akinlar [23] for edge detection, since it is a very efficient method to extract connected, 1 px-wide chains of edge pixels. In summary, this algorithm works through these steps:

1. The image gradient magnitude and direction is computed at each pixel from a smoothed version of the input image.
2. Pixels with local maximum gradient are selected. These pixels, known as *anchors*, correspond to edge elements with a very high probability.
3. Finally, anchors are connected using the neighbor's gradient magnitude and direction, following gradient maximum values.

4.2 Polygonal Chain Extraction

Smooth convex curves are then extracted from continuous edge segments in two steps. First, straight lines are extracted from the edge segments, similar to the line segment extraction method EDLines [1]. Essentially, we fit a straight line to the segments under a least squares approach and then pixels are added one by one from the connected edge segment until the error exceeds a given threshold. When no more pixels can be added for that threshold, a new line is initialized and the process is repeated with the remaining pixels.

Finally, once straight lines have been extracted, they are grouped into polygonal chains according to the smoothness and convexity assumptions. Starting from a line segment, consecutive line segments are added to the group as long as the angle between them is below a certain threshold and the whole group shares the same turn direction (refer to Fig. 1). This builds a candidate curve.

4.3 Model Generation

Given a set of pixels from a candidate curve, in this step we want to extract a model of its corresponding 3D line in a least squares sense.

As stated in Section 3, the points from a 3D line, along with the camera's center of projection, all lie in a plane. Therefore, in this step, we are interested in estimating the parameters of such plane.

A plane is described by the unit normal vector $\mathbf{n} \in \mathbb{R}^3$ and its distance to the origin $D \in \mathbb{R}$. This way, the point-to-plane distance can be simplified to

$$d_n(\mathbf{x}) = \mathbf{x} \cdot \mathbf{n} + D \quad (1)$$

Setting the center of projection coincident to the origin makes $D = 0$, and then the least squares solution to the plane parameters, in terms of the point-to-plane distance, is given by:

$$\begin{aligned} \mathbf{n}^* &= \arg \min_{\mathbf{n}} \sum_{\mathbf{u} \in \mathcal{C}} \|\pi^{-1}(\mathbf{u}) \cdot \mathbf{n}\|^2 \\ &\text{subject to } \|\mathbf{n}\| = 1 \end{aligned} \quad (2)$$

where π^{-1} is the inverse projection function, for all pixels $\mathbf{u} \in \Omega$ in the segmented curve \mathcal{C} .

Note that (2) is a quadratic system with a quadratic constraint and, thus, it can be expressed compactly in matrix form as:

$$\begin{aligned} \mathbf{n}^* &= \arg \min_{\mathbf{n}} \mathbf{n}^\top \mathbf{M} \mathbf{n} \\ &\text{subject to } \mathbf{n}^\top \mathbf{I}_{3 \times 3} \mathbf{n} = 1 \end{aligned} \quad (3)$$

where M is the symmetric matrix

$$\mathbf{M} = \sum_{\mathbf{u} \in \mathcal{C}} \pi^{-1}(\mathbf{u})(\pi^{-1}(\mathbf{u}))^\top \quad (4)$$

We solve this constrained optimization problem in closed form using the method of Lagrange multipliers as follows. Considering the Lagrangian:

$$\mathcal{L}(\mathbf{n}) = \mathbf{n}^\top \mathbf{M} \mathbf{n} + \lambda(\mathbf{n}^\top \mathbf{I} \mathbf{n}) \quad (5)$$

the necessary conditions for optimality are then:

$$\frac{\partial \mathcal{L}(\mathbf{n})}{\partial \mathbf{n}} = 2\mathbf{n}^\top (\mathbf{M} + \lambda \mathbf{I}) = \mathbf{0}^\top \quad (6)$$

Disregarding the trivial solution, the 3×3 matrix $\mathbf{M} + \lambda \mathbf{I}$ must be singular in order to satisfy (6). Thus, we solve for the values of $\lambda \in \mathbb{R}$ that make

$$\det(\mathbf{M} + \lambda \mathbf{I}) = 0 \quad (7)$$

This is a polynomial expression in λ of degree 3, for which a closed form solution exists. This is the characteristic polynomial, and the roots correspond to the eigenvalues of \mathbf{M} . Finally, the solution to the original problem can be extracted from the kernel of $\mathbf{M} + \lambda \mathbf{I}$.

Up to 3 solutions satisfy equation (6), so we choose as the optimal solution the one with the lowest cost, given by $\mathbf{n}^\top \mathbf{M} \mathbf{n}$.

4.4 Geometric Validation

Finally, once we have a model of the 3D line, we geometrically verify if it corresponds to the observed curve as follows. Given a pixel \mathbf{u} of the curve \mathcal{C} , we get the direction vector of the corresponding light-ray and project it to the plane model of the line estimated as explained above. The projected vector on the plane is finally re-projected to the image using the projection function π . Thus, we say that a curve is the projection of a 3D line, within a certain threshold ρ , if, for all segmented pixels of the curve, their re-projection fall within a ρ -radius circle around the source pixels, measured in image coordinates:

$$\|\mathbf{u} - \pi(\mathbf{x} - (\mathbf{x} \cdot \mathbf{n})\mathbf{n})\| \leq \rho, \quad \forall \mathbf{u} \in \mathcal{C} \quad (8)$$

where $\mathbf{x} = \pi^{-1}(\mathbf{u})$ and \mathbf{n} is the normal vector of the fitted plane. It is worth noticing that, for computational efficiency reasons, only the endpoints of the 2D line segments forming the polygonal chain are verified.

5 Experimental Evaluation

We have evaluated D-LSD with real images from the TUM mono dataset [12], comparing it with the fastest state-of-the-art line segment detectors: LSD [16], EDLines [1] and AG3line [24]. For a fair comparison, we have used the original authors' C++ implementations¹. All the experiments were conducted on an Intel Core i7-7700HQ CPU with 16 GB of RAM and on a Linux-based OS.

The TUM monocular dataset is tailored for the evaluation of visual odometry and SLAM systems. In this evaluation, we focus on the specific `wideWhitePaper` calibration sequence, which contains 800 images showing two sheets of white paper on top of a dark table. This sequence is particularly interesting since long and high contrast line segments are observed along the whole sequence from different points of view. A wide-angle camera was used to record this sequence, having $148^\circ \times 122^\circ$ non-rectified field of view. The camera parameters, as provided with the dataset, are calibrated using the pinhole projection with the FoV [11] distortion model. Finally, the camera has a global shutter CMOS sensor with 1280 px \times 1024 px resolution, recording frames at a fixed rate of 10 fps.

In order to test the other methods, we first rectify the raw images using the tools provided with the dataset² in `crop` mode, which keeps as much pixels from the original image as possible but without adding any black borders (see Fig. 3). We set the output image resolution to match the original 1280 px \times 1024 px for a fair comparison between the methods using rectified images and D-LSD. The other evaluated methods are then applied to the rectified images keeping their default parameters. In turn, for D-LSD, we have set the smoothness threshold to $\theta = \frac{\pi}{16}$ and the reprojection error for line validation to $\rho = 4$ px. Additionally, small (less than 25 px) single straight line polygonal chains are omitted.

¹ The reference implementation of D-LSD as well as the others' used for evaluation are publicly available at: https://github.com/dzunigan/line_detection.

² https://github.com/tum-vision/mono_dataset.code.

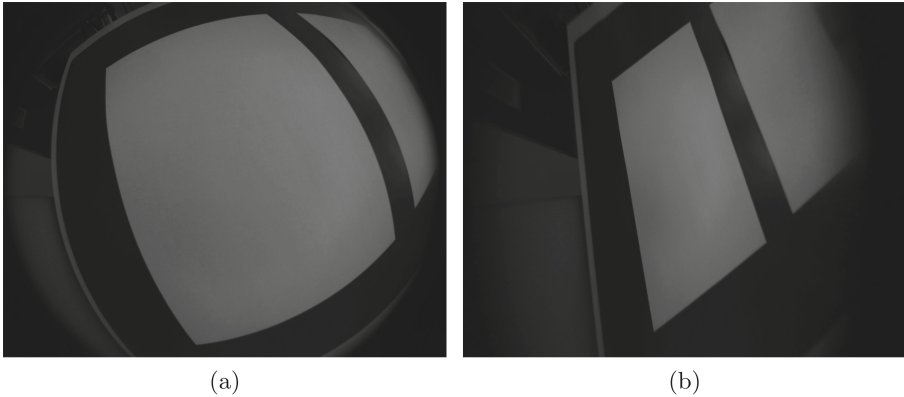


Fig. 3. Original (3a) and rectified (3b) images from the `wide_whitePaper` sequence of the TUM monocular dataset [12].

We have measured the D-LSD performance with the *recall* value, computed as the ratio of the number of line segments found in the distorted image by D-LSD w.r.t. the number of segments in the image, as given by any of the state-of-the-art methods on the rectified image. Thus, a maximum value of 100% would tell us that all the line segments detected in the rectified image have been recovered with D-LSD. A correspondence between rectified and distorted lines is set if at least 90% of the rectified line falls within 4px of a distorted line detected by D-LSD. We omit small line segments from the evaluation since their detection is highly affected by the internal parameters of each algorithm. However, the length in pixels of the same line segment is different in rectified and non-rectified images. Thus, we set the minimum line length threshold in terms of the angle between the light-rays of the endpoints of a line, which does not depend on the actual projection function. In this way, we can discard the same small segments for all the evaluated methods. The length threshold is represented as a percentage of the vertical FoV.

The recall rates for the whole `wide_whitePaper` sequence and for different values of the minimum line length threshold are shown in Fig. 4. The results of the experiment show that most of the long lines (about 21° and above) are detected by D-LSD, while it struggles to detect smaller lines. This is, in part, because of some internal thresholds of D-LSD that are expressed in pixels and thus are not equivalent when using rectified images. The total number of line segments detected by each algorithm along with the average execution time are reported in Table 1. Only line segments with length above 25% minimum length threshold (*i.e.* 61° and above) are taken into account. It is important to highlight that one of the advantages of using D-LSD is that it can detect a larger number of the long lines than the rest of the methods, since it uses the original non-rectified images with wider FoV. Not only that, our proposal is the fastest detector, even

without taking into account the previous rectification step needed by the others (first column in Table 1).

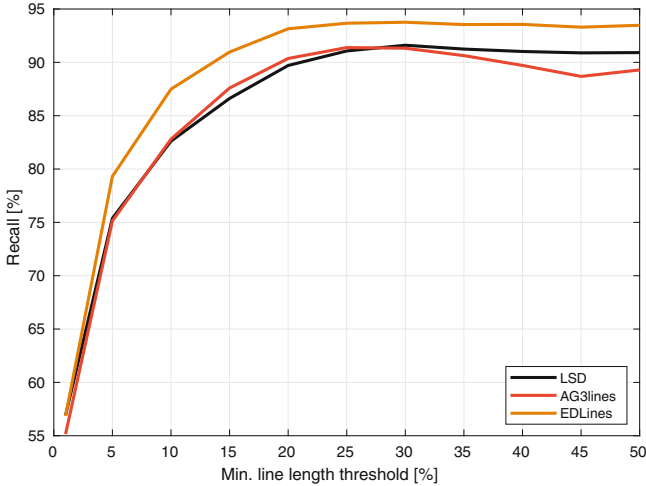


Fig. 4. D-LSD recall ratio compared to state-of-the-art line segment detectors on rectified images from the `wide_whitePaper` sequence.

Table 1. Total number of line segments detected in the `wide_whitePaper` sequence at 25 % length threshold and average execution time (with standard deviation) measured in milliseconds.

	Rectification [12]	LSD [16]	AG3line [24]	EDLines [1]	D-LSD
Total lines	–	6654	5868	6273	7186
Exec. time	7.32 (1.64)	38.8 (2.84)	20.2 (2.70)	12.9 (3.85)	12.4 (3.92)

6 Conclusions

In this work, we have presented a new line segment detector for calibrated images that works directly on the original, non-rectified image. An open source C++ implementation of our algorithm is available at <https://github.com/dzunigan/line.detection>.

Our approach, coined D-LSD, relies on the efficient Edge Drawing [23] algorithm for edge detection. Edge pixels are then grouped into polygonal chains, representing smooth convex curves. Finally, the segmented curves are validated as the projection of a 3D line by fitting a plane that includes the projection center and the light-rays passing through the curve pixels. This plane fitting problem is formulated as a quadratically constrained quadratic optimization problem, which is solved in closed form using the method of Lagrange multipliers. Our method

is general enough to handle any central projection system: from wide-angle to omnidirectional cameras.

We have evaluated the proposed detector with images captured by a wide-angle camera available in the TUM monocular dataset [12], comparing its detection capabilities against state-of-the-art line segment detectors on rectified images. In the experiments, D-LSD shows comparable performance for non-short segments while being the fastest approach and without requiring any previous image rectification. We believe D-LSD could be of much interest for demanding real time applications that rely on line features for further image analysis.

Acknowledgements. This research was funded by the Government of Spain and the European Regional Development's funds (FEDER) under the projects ARPEGGIO (PID2020-117057) and WISER (DPI2017-84827-R).

References

1. Akinlar, C., Topal, C.: EDLines: a real-time line segment detector with a false detection control. *Pattern Recogn. Lett.* **32**(13), 1633–1642 (2011). <https://doi.org/10.1016/j.patrec.2011.06.001>
2. Akinlar, C., Topal, C.: EDCircles: a real-time circle detector with a false detection control. *Pattern Recogn.* **46**(3), 725–740 (2013). <https://doi.org/10.1016/j.patcog.2012.09.020>
3. Angeli, A., Filliat, D., Doncieux, S., Meyer, J.A.: Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Trans. Robot.* **24**(5), 1027–1037 (2008). <https://doi.org/10.1109/TRO.2008.2004514>
4. Barreto, J.P.: A unifying geometric representation for central projection systems. *Comput. Vis. Image Underst.* **103**(3), 208–217 (2006). <https://doi.org/10.1016/j.cviu.2006.06.003>
5. Briales, J., Gonzalez-Jimenez, J.: A minimal closed-form solution for the perspective three orthogonal angles (P3oA) problem: application to visual odometry. *J. Math. Imaging Vis.* **55**(3), 266–283 (2015). <https://doi.org/10.1007/s10851-015-0620-x>
6. Briales, J., Gonzalez-Jimenez, J.: A minimal solution for the calibration of a 2D laser-rangefinder and a camera based on scene corners. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1891–1896 (2015). <https://doi.org/10.1109/IROS.2015.7353625>
7. Caruso, D., Engel, J., Cremers, D.: Large-scale direct SLAM for omnidirectional cameras. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 141–148 (2015). <https://doi.org/10.1109/IROS.2015.7353366>
8. Cheng, G., Han, J.: A survey on object detection in optical remote sensing images. *ISPRS J. Photogrammetry Rem. Sens.* **117**, 11–28 (2016). <https://doi.org/10.1016/j.isprsjprs.2016.03.014>
9. Cho, N., Yuille, A., Lee, S.: A novel linelet-based representation for line segment detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(5), 1195–1208 (2018). <https://doi.org/10.1109/TPAMI.2017.2703841>
10. Desolneux, A., Moisan, L., Morel, J.M.: Edge detection by Helmholtz principle. *J. Math. Imaging Vis.* **14**(3), 271–284 (2001)
11. Devernay, F., Faugeras, O.: Straight lines have to be straight. *Mach. Vis. Appl.* **13**, 14–24 (2019). <https://doi.org/10.1007/PL00013269>

12. Engel, J., Usenko, V., Cremers, D.: A photometrically calibrated benchmark for monocular visual odometry. arXiv e-prints p. [arXiv:1607.02555](https://arxiv.org/abs/1607.02555) (2016)
13. Fitzgibbon, A.W.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. 1 (2001). <https://doi.org/10.1109/CVPR.2001.990465>
14. Gomez-Ojeda, R., Briaies, J., Gonzalez-Jimenez, J.: PL-SVO: Semi-direct monocular visual odometry by combining points and line segments. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4211–4216 (2016). <https://doi.org/10.1109/IROS.2016.7759620>
15. Gomez-Ojeda, R., Moreno, F., Zuñiga-Noél, D., Scaramuzza, D., Gonzalez-Jimenez, J.: PL-SLAM: a stereo SLAM system through the combination of points and line segments. *IEEE Trans. Robot.* **35**(3), 734–746 (2019). <https://doi.org/10.1109/TRO.2019.2899783>
16. Grompone von Gioi, R., Jakubowicz, J., Morel, J., Randall, G.: LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(4), 722–732 (2010). <https://doi.org/10.1109/TPAMI.2008.300>
17. Lee, J.H., Zhang, G., Lim, J., Suh, I.H.: Place recognition using straight lines for vision-based SLAM. In: 2013 IEEE International Conference on Robotics and Automation, pp. 3799–3806 (2013). <https://doi.org/10.1109/ICRA.2013.6631111>
18. Lu, X., Yao, J., Li, K., Li, L.: CannyLines: a parameter-free line segment detector. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 507–511 (2015). <https://doi.org/10.1109/ICIP.2015.7350850>
19. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **31**(5), 1147–1163 (2015). <https://doi.org/10.1109/TRO.2015.2463671>
20. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol. 1, pp. 1. IEEE (2004)
21. Pătrăucean, V., Gurdjos, P., Grompone von Gioi, R.: Joint a contrario ellipse and line detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4), 788–802 (2017). <https://doi.org/10.1109/TPAMI.2016.2558150>
22. Svoboda, T., Pajdla, T.: Epipolar geometry for central catadioptric cameras. *Int. J. Comput. Vis.* **49**, 23–37 (2002). <https://doi.org/10.1023/A:1019869530073>
23. Topal, C., Akinlar, C.: Edge drawing: a combined real-time edge and segment detector. *J. Vis. Commun. Image Rep.* **23**(6), 862–872 (2012). <https://doi.org/10.1016/j.jvcir.2012.05.004>
24. Zhang, Y., Wei, D., Li, Y.: AG3line: active grouping and geometry-gradient combined validation for fast line segment extraction. *Pattern Recogn.* **113**, 107834 (2021). <https://doi.org/10.1016/j.patcog.2021.107834>
25. Zuñiga-Noél, D., Jaenal, A., Gomez-Ojeda, R., Gonzalez-Jimenez, J.: The UMA-VI dataset: Visual-inertial odometry in low-textured and dynamic illumination environments. *Int. J. Robot. Res.* **39**(9), 1052–1060 (2020). <https://doi.org/10.1177/0278364920938439>