

Extracting general-purpose features from LIDAR data

Yangming Li and Edwin B. Olson

University of Michigan

Department of Electrical Engineering and Computer Science

Ann Arbor, MI 48109

Email: ymli@umich.edu, ebolson@umich.edu

http://april.eecs.umich.edu

Abstract—The detection of features from Light Detection and Ranging (LIDAR) data is a fundamental component of feature-based mapping and SLAM systems. Existing detectors tend to exploit characteristics of specific environments: corners and lines from indoor (rectilinear) environments, and trees from outdoor environments. While these detectors work well in their intended environments, their performance in different environments can be very poor.

We describe a general purpose feature detector for LIDAR data that is applicable to virtually any environment. Our methods adapt classic feature detection methods from the image processing literature, specifically the multi-scale Kanade-Tomasi corner detector. Our resulting method is capable of identifying stable features at a variety of spatial scales and produces uncertainty estimates for use in a state estimation algorithm. We present results on standard datasets, including Victoria Park and Intel Research Center (both 2D), and the MIT DARPA Urban Challenge dataset (3D).

Index Terms—Robot navigation, SLAM, LIDARs, Feature detection, Corner Detector

I. INTRODUCTION

The Simultaneous Localization and Mapping (SLAM) problem is at the heart of many robot applications, and many approaches use laser range finders (LIDARs) due to their ability to accurately measure both bearing and range to objects around the robot. In the SLAM context, there are two basic approaches to mapping with LIDARs: feature extraction and scan matching. The first method extracts features (also called landmarks) from the LIDAR data; these features are added to the state vector and loops are closed using data association algorithms like Joint Compatibility Branch and Bound (JCBB) [1]. The features used often depend on the environment: in indoor settings, lines, corners and curves have been used [2], [3], [4], [5], [6]. Outdoors, the hand-written tree detector originally developed for the Victoria Park dataset [7], has been used almost universally (see [8], [9], [10] for representative examples). Naturally, tree detectors work poorly in offices, and corner detectors work poorly in forests. The lack of a general-purpose feature detector that works well in varied environments has been an impediment to robust feature-based systems.

The alternative LIDAR approach, scan matching, directly matches point clouds. This approach dispenses entirely with features and leads to map constraints that directly relate two poses. Scan matching systems are much more adaptable:

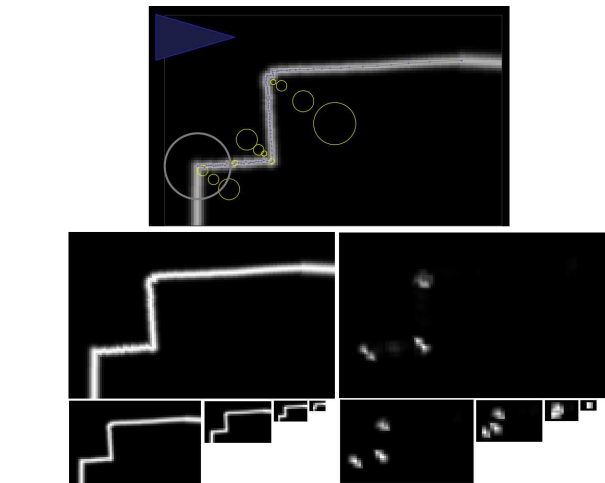


Fig. 1. Multi-scale feature extraction from LIDAR data. Our method rasterizes LIDAR data and applies the Kanade-Tomasi corner detector to identify stable and repeatable features. Top: the input image with overlaid local maxima (prior to additional filtering). Circles indicate features, with the radius equal to scale of the feature. Left: image pyramid of input. Right: Corner response pyramid, where local maxima indicate a feature.

their performance does not depend on the world containing straight lines, corners, or trees. But scan matching has a major disadvantage: it tends to create dense pose graphs that significantly increase the computational cost of computing a posterior map. For example, suppose that a particular object is visible from a large number of poses. In a scan matching approach, this will lead to constraints between each pair of poses: the graph becomes fully connected and has $O(N^2)$ edges. In contrast, a feature based approach would have an edge from each pose to the landmark: just $O(N)$ edges.

Conceptually, the pose graph resulting from a scan matcher looks like a feature-based graph in which all the features have been marginalized out. This marginalization creates many edges which slows modern SLAM algorithms. In the case of sparse Cholesky factorization, Dellaert showed that the optimal variable reordering is *not* necessarily the one in which features are marginalized out first [11]: the information matrix can often be factored faster when there are landmarks. Similarly, the family of stochastic gradient descent (SGD) algorithms [12], [13] and Gauss-Seidel re-

laxation [14], [15] have runtimes that are directly related to the number of edges. The extra edges also frustrate sparse information-form filters, such as SEIFs [16] and ESEIFs [17], [9].

Feature-based methods have an additional advantage: searching over data associations is computationally less expensive than searching over the space of rigid-body transformations: as the prior uncertainty increases, the computational cost of scan matching grows. While scan matching algorithms with large search windows (i.e., those that are robust to initialization error) can be implemented efficiently [18], the computational complexity of feature-based matching is nearly independent of initialization error. And finally, feature-based methods tend to work better outdoors, because they often reject ground strikes that result when the robot pitches or rolls.

In short, feature-based methods would likely be preferable to scan matching if they were able to offer the same robustness and broad applicability to different environments.

In this paper, we describe a general-purpose feature detection algorithm that generates highly-repeatable and stable features in virtually any environment. Our approach builds upon methods used in image processing, where the need for robust feature detectors has driven the development of a wide variety of approaches. In particular, we show how the Kanade-Tomasi [19], a variant of the Harris corner detector [20] can be applied to LIDAR data.

Our approach can be viewed as an alternative to the recent work of Zlot and Bosse [21], which proposes a number of heuristics for identifying stable keypoints in LIDAR data. Their methods begin with clustering connected components and then either 1) computing the centroids of each segment, 2) computing the curvature of each segment, or 3) iteratively computing a locally-weighted mean position until it converges. Our approach replaces these three mechanisms with a single method. They additionally investigate descriptor algorithms, which significantly simplify data association tasks. These descriptor methods could also be applied to our detector.

The central contributions of this paper are:

- We propose a general-purpose feature detector for 2D and 3D LIDAR data by adapting the Kanade-Tomasi corner detector.
- We show how to avoid false features due to missing data, occlusion, and sensor noise.
- We present experimental evidence that our methods work consistently in varied environments, while two traditional approaches do not.

In the next section, we describe how we convert 2D and 3D LIDAR data into images for feature detection. In Section III, we discuss how uncertainty information—critical for SLAM systems—can be obtained. In Section IV, we present experimental evaluations of our methods versus standard methods.

II. RASTERIZATION OF LIDAR DATA

The core idea of our method is to convert LIDAR data into an image that can then be processed by image processing methods. This process must take into account the fundamental differences between cameras and LIDARs.

A camera image samples the intensity of a scene at (roughly) uniform angular intervals. Individual pixels have no notion of range (and therefore of the shape of the surface they represent), but the intensity of those pixels is assumed to be approximately invariant to viewpoint and/or range. As a consequence, the appearance of a feature is reasonably well described by a set of pixel values.

LIDARs also sample the scene at uniform angular intervals, but each sample corresponds to a range measurement. Critically, unlike cameras, the value of each “range pixel” is profoundly affected by the position and orientation of the sensor. As a result, it becomes non-trivial to determine whether two features encoded as a set of (angle,range) tuples match.

As a consequence of these differences, we have chosen to rasterize LIDAR data by projecting the LIDAR points into a 2D Euclidean space. The resulting image roughly corresponds to viewing the scene from above (see Fig. 2). This choice of representation restores the invariance properties upon which computer vision methods rely, though this choice also creates new challenges. This section will describe how we approach these challenges.

A. 2D Data

2D LIDAR data is rendered into an image by drawing the data using a Gaussian kernel (see Fig. 2) using methods similar to [18]. The variance of this kernel reflects both the range uncertainty *and* the positional uncertainty that arises from sparsely sampling a surface. Nearby points—ostensibly part of the same physical object—are connected with a line, which is also drawn with a Gaussian kernel.

The width of the Gaussian kernel is a function of the LIDAR’s range noise σ_s , as well as the positional uncertainty of each point arising from the distance between samples. In other words, even if a sequence of three LIDAR points implies the existence of a corner, the actual position of the corner is masked by the spacing between the points. Assuming that the LIDAR measures ranges at uniform angular spacings, the spatial resolution of a measurement at range r is just $r \sin(\Delta\theta)$, where $\Delta\theta$ is the angular resolution of the sensor (typically 1 degree for a SICK sensor). The width of the Gaussian kernel (which changes for every measurement) reflects the sum of these uncertainties:

$$\sigma^2 \approx \sigma_s^2 + (r \sin(\Delta\theta))^2 \quad (1)$$

The Gaussian kernel has several practical advantages. At short ranges, the range noise of the sensor can cause smooth surfaces to appear rough. This, in turn, causes false feature detections. The Gaussian kernel essentially smooths these surfaces, preventing features from being detected.

There are two other issues that must be addressed:

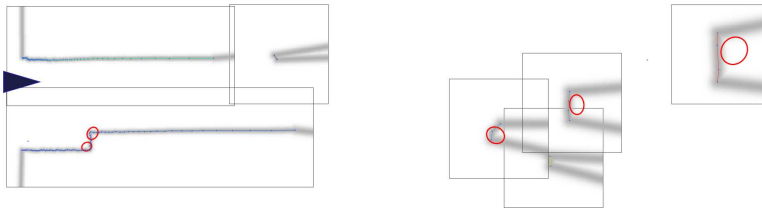


Fig. 2. Intel Research Center rasterization. This indoor dataset has many rectilinear features. Rendering is performed for each contour individually as illustrated by the black rectangular outlines; this dramatically reduces computation time. Ellipses indicate 3σ uncertainty bounds for detected features. At the occlusion boundaries of each contour, the conservatively extrapolated contour is readily visible.

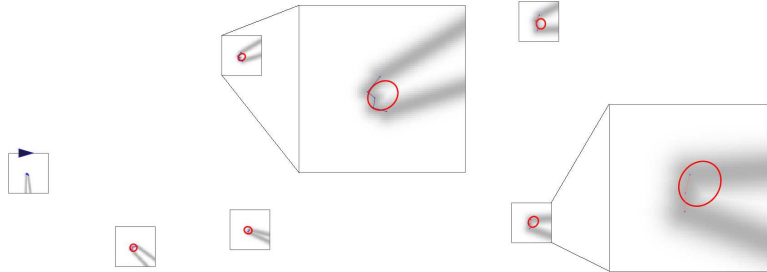


Fig. 3. Victoria Park Rasterization. This figure shows rasterization for a 2D LIDAR scan from an outdoor, tree-filled environment. The same rendering parameters were used for both the Intel and Victoria Park datasets.

- Missing data: often, the full shape of a contour is not visible, which can lead to feature detections at the visibility boundary. In some cases, the *absence* of LIDAR data is proof that a strong feature is present (Fig. 4a), while in other cases, it is possible that there isn't a strong feature at all (Fig. 4b). While we could attempt to explicitly measure and threshold the angle of the hidden corner, this process would add a number of hard-to-tune parameters. (Estimating the angle of the observed contour, for example, is sensitive to noise in the individual range measurements). Our approach is to render the most conservative (i.e., the smoothest) contour consistent with the observed data. This conservative contour is then passed to our system without additional modification.
- Occlusion: A foreground object can occlude portions of a background object (see Fig. 4c) making it appear as though the background object has an abrupt boundary. Our approach is simply to suppress feature detections that are close to these occlusion boundaries.

In the case of many 2D datasets, the majority of the rendered image is empty. It is possible to significantly accelerate the feature detection step by rendering each contour into separate, smaller images. On datasets like Victoria Park, in which there are large amounts of empty space, this technique provided an average speed up of 31.8 times.

Rendering each contour separately also makes it much easier to suppress errant feature detections caused by the conservative surface extrapolations described above; since each contour is rendered separately, there is no possibility that the extrapolated surface will intersect another contour, creating a feature.

Rasterization inevitably introduces additional quantization

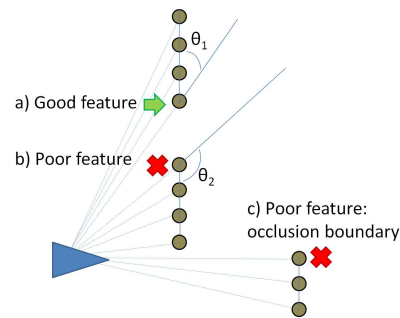


Fig. 4. Feature detection scenarios. The direction from which a surface is viewed is critical to identifying sharp features. In case A, a sharp corner must exist; in contrast, case B may not be a well-defined feature. Our method addresses this issue by rendering the worst-case (most featureless) shape, rather than attempting to threshold the angle of the hidden corner. In case C, a foreground object's shadow can cause a false boundary to appear on a background object; this case is handled explicitly.

noise. However, this quantization noise is modest in comparison to the range noise of sensors, and negligible in comparison to the uncertainty arising from sampling effects. In our experiments, for example, we used an image resolution of 2 cm per pixel.

B. 3D Data

We also tested our method on 3D Velodyne data from the MIT Urban Challenge Dataset [22]. Perhaps counter-intuitively, 3D data is much easier to process than 2D data, since it is often possible to see *over* obstacles, reducing the severity of the missing data problem.

However, we cannot render images based according to just the (x,y) location of the LIDAR samples the way we do with 2D data, since the Velodyne sensor obtains samples almost everywhere. Our approach, instead, is to render each pixel

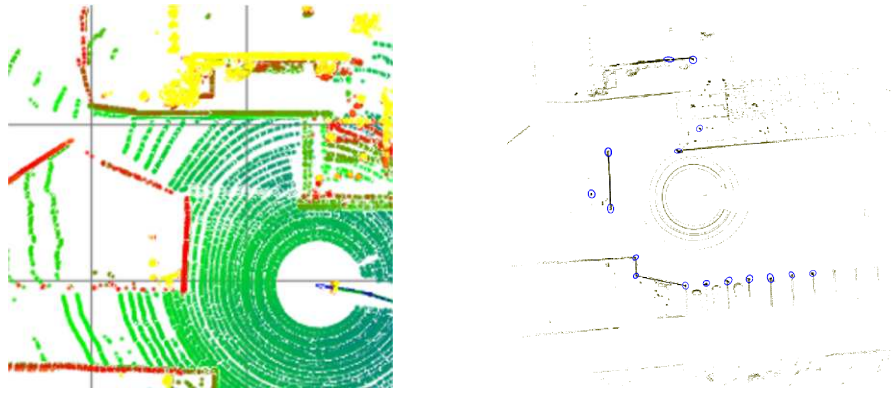


Fig. 5. 3D Scan Rasterization. Left: a Velodyne scan with points colored according to Z height. Right: rasterized image with superimposed extracted features and corresponding uncertainties. 3D LIDAR data was rasterized by considering the range of Z values in each cell of a polar grid.

according to the range of heights collected for that pixel (see Fig. 5). In other words, if three LIDAR samples are collected in the area corresponding to a single (x,y) pixel with $z = 1, 1.5,$ and 2.5 , the maximum difference in height is 1.5 . This procedure effectively measures the visible height of the objects around it and is invariant to viewpoint. However, this procedure requires a fair number of LIDAR returns for each pixel, which necessitates a coarser spatial resolution. We use a polar grid with dimensions of 1 degree by 0.15 m.

Note that we assume that the robot can measure its pitch and roll with respect to the gravity vector with reasonable accuracy; the cost of such a sensor is inconsequential in comparison to that of any laser scanner. When projecting points, the pose of the vehicle is taken into account; as a consequence, the resulting images are invariant to roll and pitch.

III. FEATURE DETECTION

Once an image has been produced using the methods in the previous section, the next task is to identify stable and repeatable features. The Kanade-Tomasi corner detector [19] is virtually identical to the Harris corner detector [20], with the exception that the minimum eigenvalue of the structure tensor is computed exactly, rather than approximated. We achieved noticeably better results from the Kanade-Tomasi detector.

Importantly, the KT corner detector is rotationally invariant when the image has been convolved with a circular filter. Our images naturally meet this condition, since they are constructed using Gaussian kernels.

Our system detects features at a variety of scales so that we can exploit features that are both physically small and large. To do this, we perform KT corner detection on each level of a power-of-two image pyramid, extracting corners wherever local maxima occur.

This feature-detection scheme is very similar to that used by the SURF detector [23]. Our implementation down-samples images using a $\sigma = 1.0$ Gaussian kernel of width 5. Corners are additionally subjected to a threshold of 0.2. This design parameter is fairly robust: the system works well on a variety of datasets over a range of values.

While we want to detect features at multiple scales, we do not want to match these features in a scale *invariant* manner: unlike cameras, LIDARs directly observe the scale of the objects in the environment. Thus, unlike camera-based methods, the scale at which we detect an object is useful in data association.

The positional uncertainty of features is of critical importance to SLAM applications. Fortunately, it has been shown that the covariance matrix of a Harris Corner is simply equal to the inverse of the structure tensor [24]. Our use of variable-sized Gaussian kernels when rasterizing the LIDAR data encodes the spatial uncertainty in the image, and this is reflected in the structure tensor. All that remains to be done is to scale the covariance matrix according to the square of the resolution of the image (in meters per pixel).

The covariance estimates produced by our system can be seen in Fig. 2 and Fig. 3 for 2D data, and in Fig. 5 for 3D data. The ellipses correspond to 3σ confidence intervals. The fact that principled covariance estimates can be easily derived is one of the strengths of our method.

IV. RESULTS

In a SLAM context, repeatability (the consistency with which a given landmark is observed) is critical. Each re-observation of a landmark creates a “loop closure”, improving the quality of the posterior map.

We measured the repeatability of our feature detector using two well-known datasets: the Intel Research Center dataset (indoor and rectilinear) and the Victoria park (outdoor with clutter). For these datasets, we used posterior position estimates produced by conventional SLAM methods; this “ground truth” allowed us to test whether a particular landmark *should* have been observed given the location of the robot. When evaluating whether a landmark was correctly observed, we used a simple nearest-neighbor gating rule: if a feature was observed within a distance d_1 of an existing landmark, the two were associated. If the feature was more than d_2 away from the nearest landmark, a new landmark was created. Features between d_1 and d_2 were discarded. On the Intel data set, we used $d_1=0.1$ m, and $d_2=0.3$ m, and on the Victoria Park dataset, we used $d_1=1.0$ m, $d_2=3.0$ m.

	Corner Detector		Tree Detector		Proposed Method	
	Victoria	Intel	Victoria	Intel	Victoria	Intel
Feature Detections	409 -	1585	31607	195 -	52545 +	6932 +
Number of instantiated landmarks	43 -	276	748	75 -	1939 +	1294 +
Number of re-observed landmarks	18 -	159	325	12 -	1089 +	777 +
Number of re-observations	201 -	807	24954	25 -	36281 +	4336 +

Fig. 6. Feature detection performance. Three detectors ran on two different datasets: Intel Research Center (indoor) and Victoria Park (outdoor). New features were instantiated when a new detection was far away from any previous landmarks. A larger number of loop closures (re-observations) generally leads to better maps. For each dataset, the best performing method is marked with a “+”, and the worst performing method is marked with a “-”. Our method outperforms the other two methods even in the environments for which the specialized methods were designed.

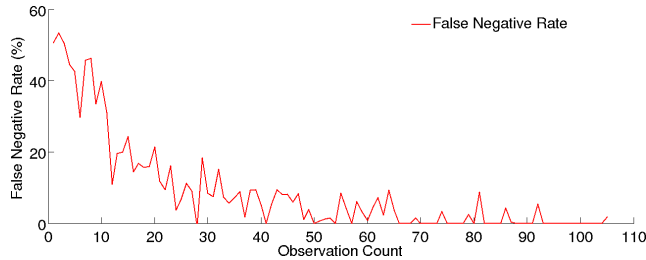


Fig. 8. False Negative Rate. The plot shows the frequency with which a landmark is not correctly detected as a function of how often the landmark is detected. It illustrates that our method achieves very low false-negative rates. There is also a clear trend: the more often a landmark is observed, the lower the false positive rate. The very low false negative rates achieved would allow systems to make use of negative information.

In addition to our proposed method, we provide two comparison methods (both of which used the same data association procedure):

- Corner detector: Line segments are extracted from nearby points using an agglomerative method. If the endpoints of two lines lie within 1.2 m of each other, and the angle of between the two lines is between 75 and 105 degrees, a corner is reported. The particular method is adapted from [5].
- Tree detector: The standard method of tracking features in Victoria park is using a hand-written tree detector with hand-tuned parameters [7]. We used this detector with no additional modifications.

As shown in Fig. 7, the performance of the proposed method is generally as good or better than the other detectors. In contrast, while the performance of the tree detector is good in the Victoria Park dataset, it is very poor indoors (where it is in the “wrong” environment). Similarly, the performance of the corner detector is good in the Intel dataset and poor in Victoria Park. Our general-purpose method performs well in both environments.

The false negative detection rate, as shown in Fig. 8, was calculated according to following procedure: for every landmark that is detected within the field of view of a given LIDAR scan, we attempt to match it to previously-known landmarks. If it is successfully associated with an existing landmark, the observation count o for that landmark is incremented. If a landmark *should* have been detected, but was not (or was too far away), the failure count f for that landmark is incremented. We compute the false negative rate as $f/(f+o)$. As shown in Fig. 8, our detector is able to

identify features which are highly stable: they are observed many times with low false negative rates. The figure also shows that some features are detected but are observed less frequently and less consistently; empirically, these seem to correspond to small and difficult-to-observe features. Still, the graph illustrates that, in many cases, the false negative rate is low enough to enable SLAM algorithms to make use of *negative* information—to reject loop closures when landmarks do not appear as expected.

V. CONCLUSION

We have described rasterization methods for 2D and 3D laser scans that allows computer vision methods to be applied to LIDAR data. We demonstrate the general-purpose applicability of the method on benchmark indoor and outdoor datasets, where it matches or exceeds the performance of specialized detectors. The method is also capable of producing uncertainty estimates, a critical factor for SLAM applications. In our future work, we plan to more deeply explore the rasterization issues arising from 3D data and to apply these methods to 3D data collected with a nodding LIDAR (rather than a Velodyne). We also wish to determine whether image-processing type feature descriptors, such as SIFT [25] or SURF [23] could be adapted to LIDAR data. In this case, the fundamental differences between cameras and LIDARs pose significant challenges.

ACKNOWLEDGEMENTS

This work was supported by U.S. DoD grant W56HZV-04-2-0001.

REFERENCES

- [1] J. Neira and J. D. Tardos, “Data association in stochastic mapping using the joint compatibility test,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, December 2001.
- [2] P. Nunez, R. Vazquez-Martin, J. del Toro, A. Bandera, and F. Sandoval, “Feature extraction from laser scan data based on curvature estimation for mobile robotics,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 1167–1172.
- [3] G. A. Borges and M.-J. Aldon, “Line extraction in 2d range images for mobile robotics,” *J. Intell. Robotics Syst.*, vol. 40, no. 3, pp. 267–297, 2004.
- [4] A. Diosi and L. Kleeman, “Uncertainty of line segments extracted from static sick pls laser scans,” in *SICK PLS laser. In Australasian Conference on Robotics and Automation*, 2003.
- [5] E. Olson, “Robust and efficient robotic mapping,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.
- [6] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, “A comparison of line extraction algorithms using 2d range data for indoor mobile robotics,” *Auton. Robots*, vol. 23, no. 2, pp. 97–111, 2007.

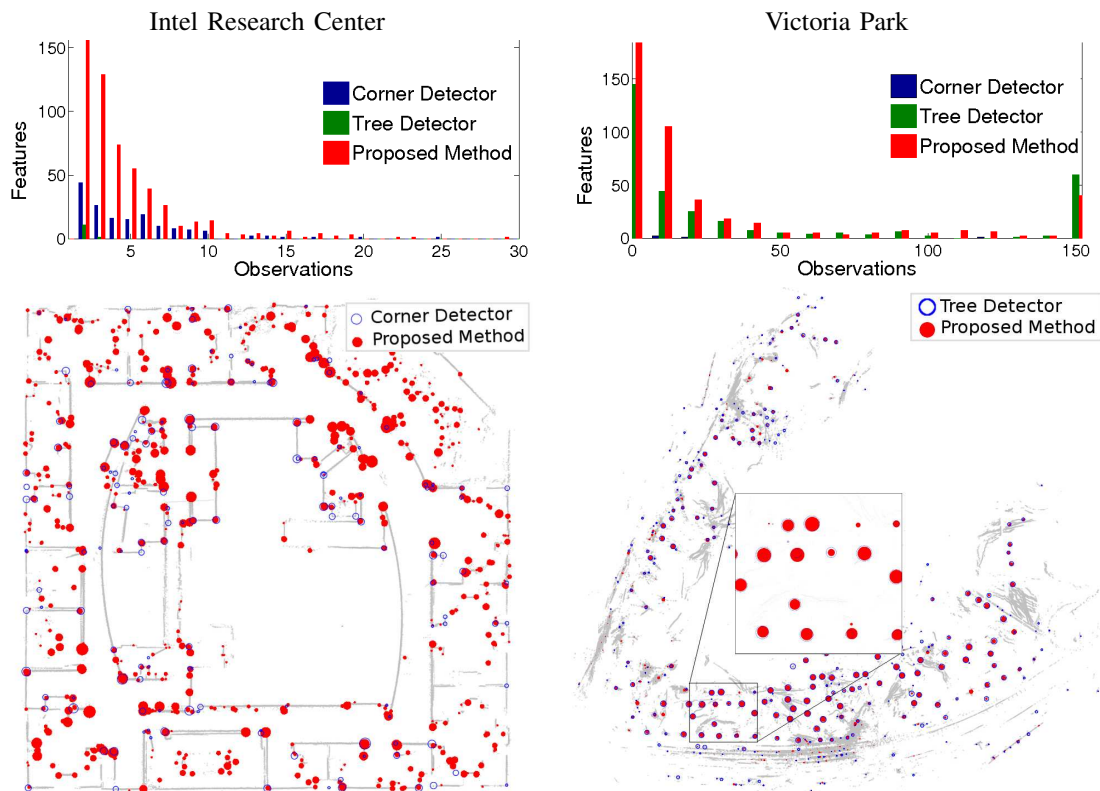


Fig. 7. Repeatability. Top: histograms show the number of features according to the number of observations of each feature. Bottom: the location of detected features is shown; the size of each marker represents the number of re-observations (large dots denote often-reobserved features). The proposed method matches or exceeds the performance of the corner and tree detectors even in the environments for which those detectors were designed. While the performance of the corner and tree detectors is very poor in the “wrong” environment, the proposed method is robust in both environments. Note: in the histogram for Victoria Park, the peak at 150 observations is the result of the fact that it represents *all* values greater than or equal to 150.

[7] J. E. Guivant, F. R. Masson, and E. M. Nebot, “Simultaneous localization and map building using natural features and absolute information,” *Robotics and Autonomous Systems*, vol. 40, no. 2-3, pp. 79–90, 2002.

[8] M. Montemerlo, “FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association,” Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.

[9] M. R. Walter, R. M. Eustice, and J. J. Leonard, “Exactly sparse extended information filters for feature-based SLAM,” *Int. J. Rob. Res.*, vol. 26, no. 4, pp. 335–359, 2007.

[10] Y. Liu and S. Thrun, “Results for outdoor-slam using sparse extended information filters,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002, pp. 1227–1233.

[11] F. Dellaert, “Square root SAM,” in *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, USA, June 2005.

[12] E. Olson, J. Leonard, and S. Teller, “Fast iterative optimization of pose graphs with poor initial estimates,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2262–2269.

[13] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, “A tree parameterization for efficiently computing maximum likelihood maps using gradient descent,” in *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

[14] T. Duckett, S. Marsland, and J. Shapiro, “Learning globally consistent maps by relaxation,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, San Francisco, CA, 2000, pp. 3841–3846.

[15] U. Frese, P. Larsson, and T. Duckett, “A multilevel relaxation algorithm for simultaneous localization and mapping,” *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 196–207, April 2005.

[16] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., April 2003.

[17] R. Eustice, H. Singh, and J. Leonard, “Exactly sparse delayed-state filters,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 2428–2435.

[18] E. Olson, “Real-time correlative scan matching,” in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 4387–4393.

[19] C. Tomasi and T. Kanade, “Detection and tracking of point features,” Tech. Report CMU-CS-91-132, Carnegie Mellon University, Tech. Rep., April 1991.

[20] C. Harris and M. Stephens, “A combined corner and edge detection,” in *Proceedings of The Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[21] R. Zlot and M. Bosse, “Place recognition using keypoint similarities in 2d lidar maps,” in *ISER*, 2008, pp. 363–372.

[22] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams., “A perception driven autonomous urban vehicle,” *Journal of Field Robotics*, September 2008.

[23] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” in *9th European Conference on Computer Vision*, Graz Austria, 2006.

[24] U. Orguner and F. Gustafsson, “Statistical characteristics of harris corner detector,” in *Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th Workshop on*, Aug. 2007, pp. 571–575.

[25] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.