

Results on Range Image Segmentation for Service Robots

Stefan Gächter, Viet Nguyen, and Roland Siegwart
Laboratoire de Systèmes Autonomes (LSA)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland
{stefan.gaechter, viet.nguyen, roland.siegwart}@epfl.ch

Abstract

This paper presents an experimental evaluation of a plane extraction method using various line extraction algorithms. Four different algorithms are chosen, which are well known in mobile robotics and computer vision. Experiments are performed on two sets of 25 range images either obtained by simulation or acquired by a proprietary 3D laser scanner. The performance of the range image segmentation is measured in terms of an average segment classification ratio. Moreover, the speed of the method is measured to conclude on the suitability for service robot applications.

1 Introduction

Range image segmentation is a long standing issue. However, ready made solutions are not available as it is the case for intensity images. One reason may be the diversity in range image acquisition and format making it difficult to develop a generic algorithm. This paper approaches the problem based on a proprietary 3D laser scanner.

Even though integrated range image vision system for service robots has been proposed earlier, see [10] and others, thorough performance analysis has been little addressed for segmentation of range images acquired in real indoor environments. The contribution of this paper is the detailed analysis of one method enabling synergy effects when using current technologies in mobile robotics, where robustness, speed, and scalable algorithm are important aspects.

2 Problem Definition

A 3D range image describes the distance measurements from the sensor to surface points on objects in a scene. In the present case, the points of the range image are specified

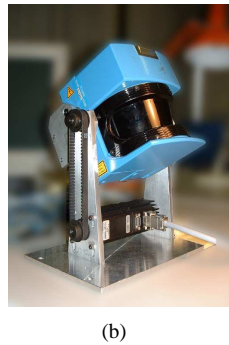
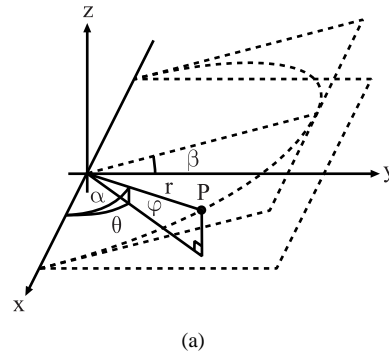


Figure 1. In (a) is depicted the coordinate system of the range image in 3D. In (b) is depicted the 3D range image sensor - a pivoting 2D laser scanner. A measurement point P of the range image is specified in spherical coordinate system (r, θ, φ) . The range image is the composition of range scans at different elevation angles β . The range scans in 2D are specified in the polar coordinate system (r, α) .

in spherical coordinate system (r, θ, φ) as depicted in Figure 1(a). It is common to assume that the noise on range image measurement r follows a Gaussian distribution with zero mean, variance σ_r^2 , and negligible angular uncertainty in θ and φ . In case of line extraction from 2D range scans, more sophisticated error models are discussed in [1]. The latter also addresses systematical errors. Even though more sophisticated error models have been used, the performance gain by their application remains still uncertain.

The problem is, given a noisy range image of unknown objects in an indoor scene, to segment the measured points into planar surface primitives that will be useful for context understanding in mobile robotic applications. Various algorithms attempt to solve this problem. One algorithm - scan line grouping - is discussed in the following sections for range images acquired with a pivoting 2D laser scanner as depicted in Figure 1(b). The algorithm partitions the 2D range scans into straight line segments and merges these into 3D planar surface segments. The straight line segments are described by the line equation in polar form:

$$x \cos(\alpha) + y \sin(\alpha) = r,$$

where $r > 0$ is the perpendicular distance from the origin to the line and $-\pi < \alpha \leq \pi$ is the angle between the x axis and r . The planar surface segments are described by the plane equation in spherical form:

$$x \cos(\varphi) \cos(\theta) + y \cos(\varphi) \sin(\theta) + z \sin(\varphi) = r,$$

where $r > 0$ is the perpendicular distance from the origin to the plane, $-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2}$ is the angle between the xy plane and r , and $-\pi < \theta \leq \pi$ is the angle between the x axis and the projection of r onto the xy plane.

3 Selected Algorithms and Related Work

In case of range images, segmentation is a data-driven process using no application specific knowledge, but using generally applicable knowledge about surfaces. The algorithm discussed here, originally presented in [6], proceeds in two segmentation steps. First, points on a 2D range scan are partitioned into straight line segments. Second, these line segments are merged into 3D planar surfaces. Therefore, line and plane extraction are performed in succession.

3.1 Line Extraction

This section briefly presents four line extraction algorithms on 2D range scans. A more detailed discussion on line extraction algorithms and their implementation can be found in [11]. The selection is based on the popularity in both mobile robotics and computer vision.

Recursive-Line-Fitting (RLF). Initially, a set s_0 consists of N_0 measurement points. A line is fitted to the set. The point P_s with maximum distance to the line is detected. If the distance exceeds an inlier threshold, then the set is split up at the point P_s into two subsets s_1 and s_2 of size N_1 and N_2 respectively. The splitting is repeated for each set s_i until the maximum distance is less than the threshold for all sets.

Iterative-End-Point-Fit (IEPF). The procedure is the same as the recursive-line-fitting algorithm, except that the fitted line is constructed simply by connecting the end points in each set.

Hough-Transform (HT). Each of the N_0 measurement points in the initial set s_0 is transformed to the line parameter space. Measurement points that belong to the maximum crossing point in the parameter space which exceeds an accumulation threshold and that are smaller than an inlier threshold form the subset s_i . The subset s_i is removed from the initial set and the procedure is repeated until the maximum crossing point is less than the accumulation threshold.

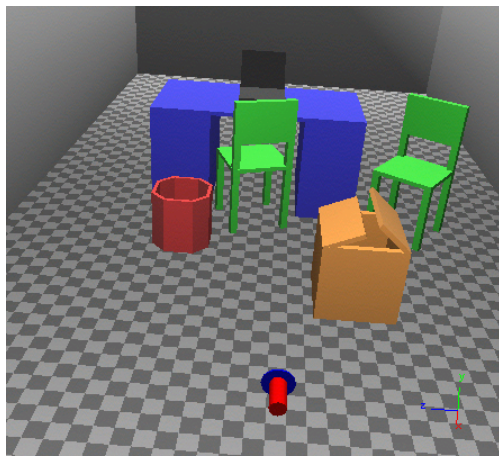
Incremental Algorithm (IA). A subset s_i is created out of the initial set s_0 consisting of two measurement points in series. The line parameters for the subset are computed. The next point in series of the initial set is added to the subset, if the recomputed line parameters satisfy the line conditions. Otherwise, the procedure is repeated until all points are assigned to a subset. The incremental process can be speeded up by adding few points in a series instead of one.

4 Experimental Comparison

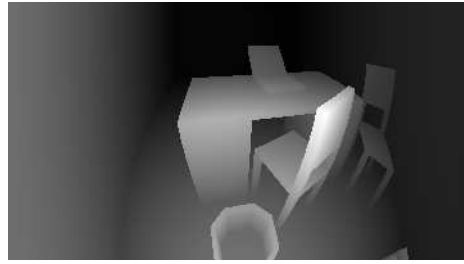
4.1 Plane Extraction

A large number of works are devoted to the range image segmentation problem, but unlike in the previous section, only one plane extraction algorithm is presented here. This plane extraction algorithm is an extension of line extraction algorithms that are extensively used in mobile robotics for localization and mapping. The experimental comparison will reveal the potential of these algorithms for plane extraction and, thus, possible synergy effects.

The plane extraction algorithm under consideration is based on region growing where the primitives are straight line segments instead of individual measurement points. This algorithm, presented in [6], gained some popularity in the mobile robotics field [10, 9] due to its simplicity and speed. Similar algorithms for planar surfaces have been developed [3] or have been extended to curved surfaces [7, 8, 4].



(a)

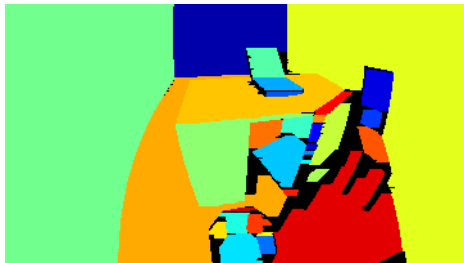


(b)



(c)

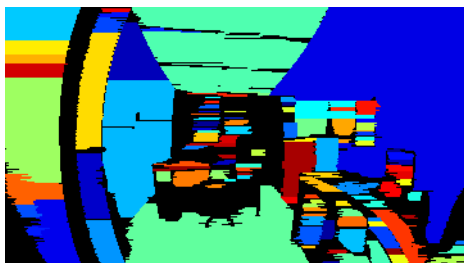
Figure 2. In (a) is depicted the virtual office scene from where the simulated range images have been taken. The real office scene from where the real range images have been taken is not depicted here. The measurement points in spherical coordinate system are projected on a 2D view plane, where the range values are represented proportionally by gray levels. The projections of the simulated and real range image are depicted in (b) and (c) respectively.



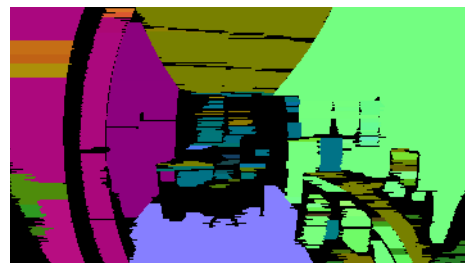
(a)



(b)



(c)



(d)

Figure 3. In the upper row, images (a) and (b), are depicted the segmentation results of the simulated range image given in Figure 2(b). In the lower row, images (c) and (d), are depicted the segmentation results of the real range image given in Figure 2(c). The images in the left column show the segmented regions, where the images in the right column show the corresponding orientations.

Scan-Line-Grouping (SLG). The algorithm assumes that all measurement points on a straight 3D line segment belong to the same planar surface. Therefore, each range scan is divided into straight line segments and their neighborhood relationship is established. Out of these segments, potential *seed regions* consisting of three neighboring line segments are created and rated by an *optimality measure* in the range of $[0, 1]$, where 0 indicates the worst and 1 the best possible seed region. A seed region is an initial subset that originates a plane. The plane parameters for a subset are computed by least-square fit. A neighboring line segment is added to the subset, if the distances between its two end points and the plane are within a threshold. The procedure is repeated until no more neighboring line segments can be added, at which time a new subset is started using the next best available seed region. The region growing is repeated until no seed region remains.

In the current implementation, the algorithm is modified to take into account the spherical coordinate system [2]. Moreover, the current implementation is without any pre- and post-treatment.

4.2 Experimental Setup

The performance of the range image segmentation algorithm is evaluated as described in [5]. However, different sets of range images are used to account for the particularity of the range image sensor. The range images used by [5] and others are acquired by sensors with a small field-of-view. The images have high density and are almost uniformly sampled on a grid. In contrast, the present sensor has a large field-of-view. The images have low density and are uniformly sampled in the angle, see Figure 2(b) and (c). The sets of used range images are based on simulation and real experiments. The ground truth was created for the simulated range images.

Range Image Sensor. The 3D range image sensor, see Figure 1(b), is a custom setup based on a 2D laser scanner, SICK LMS200. In the current configuration, the sensor has a maximum measurement range of 8m, a range resolution of 10mm, a systematic error range of ± 15 mm, and a statistical error standard deviation of 5mm. These values have been validated by [13] for most measurement conditions of varying reflectivity and incidence angle. The sensor has a horizontal scan angle of 180° with angular resolution of $\Delta\alpha = 0.5^\circ$. The 2D laser scanner is mounted on a pivoting support. The support is driven by a step motor via a belt transmission. A similar design has been used by [12] and others. In the current configuration, the sensor has a vertical scan angle of 90° with angular resolution of $\Delta\beta = 0.45^\circ$. The minimum elevation angle is $\beta_{min} = -45^\circ$. Hence, the sensor has a field of view $90^\circ \times 180^\circ$ in vertical and hori-

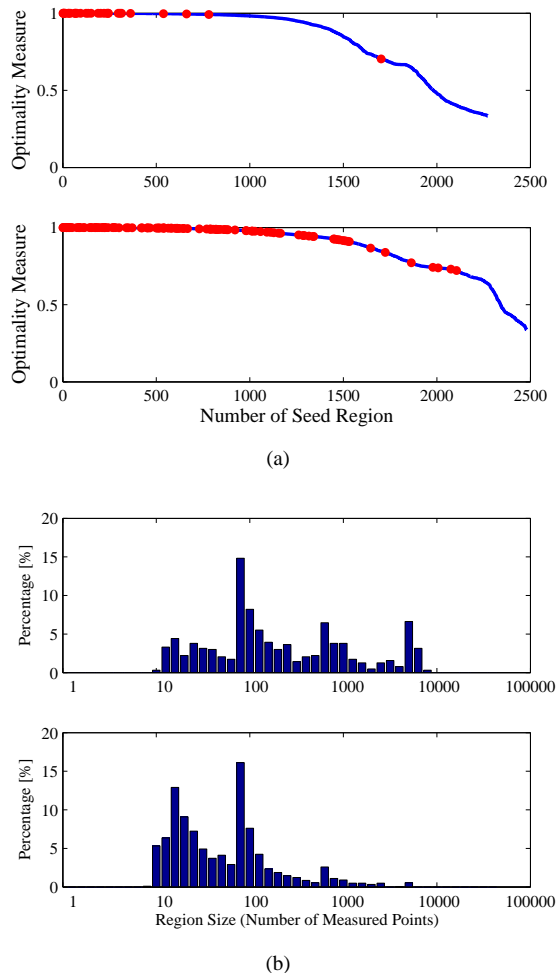
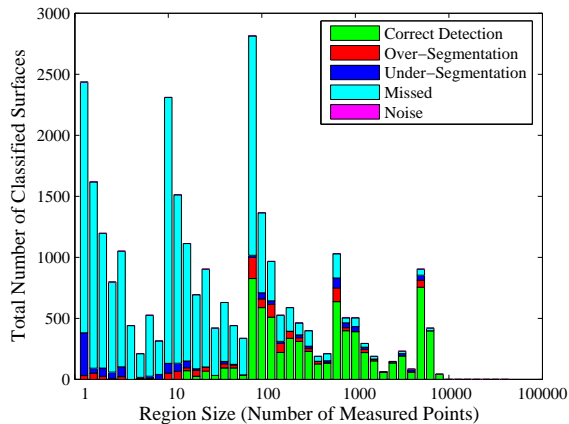
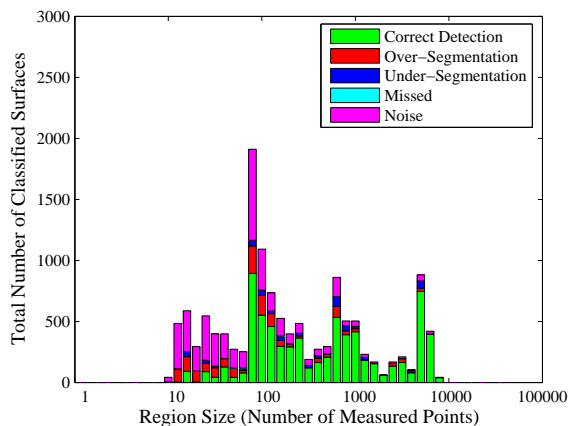


Figure 4. In (a) is depicted the optimality measure in function of the number of seed regions, where the blue line indicates the measure for all initial seed regions and the red dots only the ones resulting in the final segments. The upper and lower graph correspond to the range image depicted in Figure 2(b) and (c) respectively. In (b) are depicted the region size histograms. The upper and lower graph correspond to the set of 25 simulated and real range images respectively. The region size histogram is normalized by the total number of regions.



(a)



(b)

Figure 5. In (a) and (b) are depicted the region size histograms for the planar surfaces in the ground truth and in the segmented range images respectively. The region size is the number of measurement points in a segment. The size distribution for the different classified instances is given for correct detection (green), over-segmentation (red), under-segmentation (blue), missed (cyan), and noise (magenta) instances. The classification is based on a set of 25 simulated range images when using the recursive-line-fitting algorithm for line extraction.

zontal direction and a complete 3D range image consists of $201 \times 361 = 72,561$ measurement points.

The first test set consists of 25 range images of a typical office environment.¹ A pair of images with a different viewpoint each have been taken in 12 different rooms. The rooms are highly structured, but also exhibit large planar surfaces where unobstructed walls and floor are present. The reflectance of the objects varies from opaque to transparent.

The second test set consists of 25 range images of a virtual office environment. Each image has been taken from a different viewpoint. The environment consists of a table, two chairs, a notebook, a dust bin, and a box, see Figure 2(a). The sensor model has the specification as stated above and has been implemented in Webots, a mobile robot simulation software developed by Cyberbotics Ltd. The ground truth for this test set was created in a semi-automatic manner [2].

Performance Metrics. The performance is measured by comparing the segmentation outcome for simulated range images with the ground truth as described in [5]. Five types of region classification are considered: correct detection, over-segmentation, under-segmentation, missed, and noise. Over-segmentation results in multiple detections of a single surface. Under-segmentation results in insufficient separation of multiple surfaces. A missed classification is used when the segmentation algorithm fails to find a surface, which appears in the image (false negative). A noise classification is used when the segmentation algorithms find a surface, which does not appear in the image (false positive). The formulas for deciding classification are based upon the classification threshold T in the range of (50%, 100%]. The classification threshold measures the congruency between segmented surface and ground truth. The metrics defining each classification are given in [5].

The parameter values for the segmentation algorithm are chosen according to the sensor and environment. The parameters are divided into two types: common parameters and algorithm specific parameters. Common parameters are those shared by all algorithms and for all test sets. These are the minimum number of points per line segment, 9, the minimum physical length of a line segment, 10cm, the standard deviation of range measurement, 1.0cm, the maximum distance from a point to a line where the point is considered inlier to the line, the inlier threshold, 1.0cm, and the maximum distance from an end point of a line segment to the approximated plane that the line segment is considered as part of a plane, the merging threshold, 1.5cm. The algorithm specific parameters are based on the results in [11].

The simulated range images are corrupted with noise following a Gaussian distribution with zero mean and standard

¹The test sets are available upon request from the authors.

deviation $\sigma_r = 1.0\text{cm}$. The standard deviation is chosen larger than the range image sensor's one to simulate texture. Moreover, 25% of jump edges in the range scans are considered as mixed measurement points; neighboring measurement points with range difference greater than 50cm are replaced by their mean value.

The algorithms for line extraction are implemented in C and that for plane extraction in MATLAB. The experiments are performed on a notebook with PentiumM-1.8GHz and 1GB of memory. The values of computation time are measured with the MATLAB profiler.

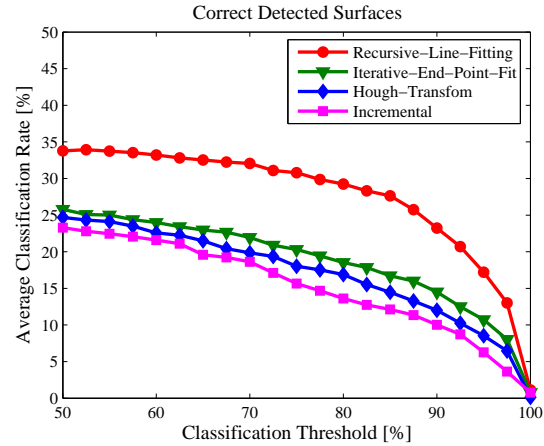
In the following sections, firstly, the segmentation results for simulated and real range images are compared and, secondly, the segmentation results for simulated images are compared with the ground truth.

4.3 Validation of Simulated Range Images

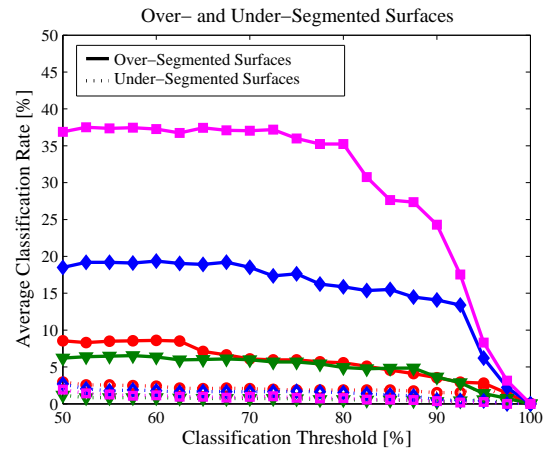
Because no ground truth is available for the real images, the segmentation outcome for simulated range images is compared with the outcome for real range images based on the measures *number of seed regions* and *size of the segmented regions*. This gives a basic idea of how well the simulated images validate the real images.

The range images to exemplify the segmentation algorithm are depicted in Figure 2(b) and (c). The algorithm for the line extraction is the recursive-line-fitting method. This algorithm has the best performance as it is shown later. The outcome is depicted in Figure 3(a) and (c). The orientations of each segment are depicted color coded in Figure 3(b) and (d). The optimality measure of the corresponding seed regions is depicted in Figure 4(a). As it can be seen, the number of initial seed regions for the simulated and real range image, 2272 and 2480 respectively, are similar, however, the number of seed regions resulting in the final segments, 31 and 114 respectively, is considerably higher for the real image, because the real image is more structured than the simulated one.

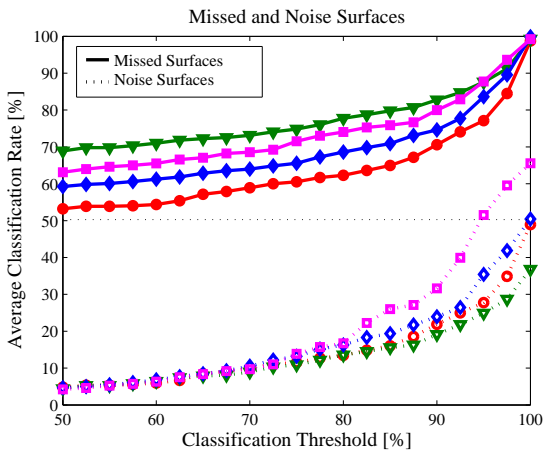
In Figure 4(b) are depicted the region size histograms of the two sets. The region size is the number of measurement points in a segment. The simulated set tends to result in larger segments while the real set in smaller ones. Moreover, the average ratio of discarded to total measurement points is 4.5% and 41.6% for the simulated and real set respectively, which correlates with the total number of regions of 634 and 3083 respectively. It is obvious that the two sets have rather different characteristics for large and small segments. However, the region size histogram has a similar distribution between about 80 and 150 measurement points. It is assumed that at least for this band the correlation between simulated and real image set is strong enough to make a conclusion on the comparison with the ground truth in the following section.



(a)



(b)



(c)

Figure 6. In (a), (b), and (c) are depicted the average classification rates in function of the classification threshold for four different line extraction algorithms. The average is taken over a set of 25 simulated range images.

4.4 Result of Simulated Range Image Segmentation

The average classification rates in function of the classification threshold for the simulated set are depicted in Figure 6. The average classification rate is the mean of the number of classified planar surfaces divided by the total number of planar surfaces in the ground truth. Thus, a perfect range image segmentation would result in a correct detection classification rate of 100%. The classification threshold measures the congruency. A threshold of 100% demands perfect congruence between regions in the ground truth and the corresponding ones in the range images. This is virtually impossible to achieve and, therefore, the number of missed regions increase to 100% with increasing threshold as can be seen in Figure 6(c).

As shown in the Figure 6, the segmentation method based on the recursive-line-fitting algorithm performs best. Generally, this algorithm has better performance in correct detection and low over- and under-segmentation over the whole range. When the iterative-end-point-fit algorithm is used, the correct detection is poorer and the initial rate drops by 10%. This algorithm tends to shorten the line segments. The average outlier ratio is 11.1% compared with 4.3% for the previous algorithm, see Table 1. Thus, the influence of false edge detection is more unlikely, which is an advantage when merging the line segments and a slightly better performance in over- and under-segmentation results. In contrast, the algorithm misses most small planar surfaces, because short line segments are discarded.

In general, the average classification ratio of missed surfaces is high given that a minimum segmentation length is imposed, which has similar or larger size than the smallest structure in the ground truth. It is the same for the minimum number of measurement points per line segment. This is clearly visible in the region size histogram of classified instances depicted in Figure 5, where most of the missed regions are of small size. The histograms are based on the result for the recursive-line-fitting method.

The performance when using the other algorithms differs mainly in the outcome of the over-segmentation, which is considerably higher. The line extraction based on Hough-transform can result in ambiguous line segments, i.e. line segments from the same range scan intersect or overlap. In case of the incremental method, false edge detection is the reason. The edges in the range scan are in general not uniformly sampled in distance, which is a precondition for a good performance of the incremental algorithm. Therefore, edge points are added to a wrong line segment and alter its true pose. The result for both methods is over-segmentation.

Table 1 states the computation time per range image for the main processing steps: line extraction, neighborhood relation compilation, seed region computation, and plane

Table 1. Computation Time per Range Image

	RLF	IEPF	HT	IA
Avg Outlier Ratio	4.3%	11.1%	10.5%	13.0%
Line Extraction	0.62s	0.79s	22.04s	0.85s
Neighborhood Rel.	0.23s	0.18s	0.20s	0.17s
Seed Regions	0.44s	0.26s	0.40s	0.22s
Plane Merging	3.86s	3.27s	3.77s	2.17s
Total	5.16s	4.50s	26.42s	3.41s

merging. Moreover, the average number of extracted segments and outlier ratio is given. The average is taken over a set of 25 simulated range images with size of $201 \times 361 = 72'561$ measurement points. The plane merging step is the most demanding. The average computation time for line extraction may vary - the Hough-transform based algorithm is by far the slowest - the average time for the other processing steps is similar among the different methods. The total average computation time for the best case is about 3.41s for the incremental based method. Over-segmentation and high outlier ratio may have a positive impact. The computation time, when using the set of real range images, roughly doubles for the line extraction step and roughly halves in the plane merging step. The computation for the other steps remains similar. Thus, the total average computation time for real images does not change significantly, from 5.16s to 4.01s, when using the recursive-line-fitting algorithm.

5 Conclusion

This paper has presented an experimental evaluation of a plane extraction method using four different line extraction algorithms. Overall, the range image segmentation based on the recursive-line-fitting algorithm has best performance. The range image segmentation based on the iterative-end-point-fit algorithm may perform better with real images, because the line segments depend less on the accuracy of the detected end points. In general, the quality of range image segmentation is strongly related to the performance of the line extraction method. The right choice may differ along with the application and implementation details.

The paper has also pointed out the difficulty to simulate range images. Even though indoor environments feature mainly planar surfaces, they are highly structured and the surface reflectance varies strongly. The resulting images are much more cluttered and noisy than can be achieved by a simple simulation. It needs better modeling to test the segmentation method soundly. Still, the simulation had the advantage of controllability and it was possible to conclude on the basic performance of each line extraction method.

The range image segmentation, as presented here, is useful for applications where time is critical and the field of view can be limited.

Acknowledgment

This work is supported by the Swiss National Science Foundation through the National Center of Competence in Research on Interactive Multimodal Information Management (IM2).

References

- [1] A. Diosi and L. Kleeman. Uncertainty of line segments extracted from static SICK PLS laser. In *Australasian Conference on Robotics and Automation*, December 2003.
- [2] S. Gächter. Results on range image segmentation for service robots. Technical Report EPFL-LSA-2005-01, Laboratoire de Systèmes Autonomes, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, 2005.
- [3] M. Haindl and P. Žid. Fast segmentation of range images. In A. Del Bimbo, editor, *Image Analysis and Processing*, Lecture Notes in Computer Science 1310, pages 295–302. Springer-Verlag, Berlin, 1997.
- [4] M. Haindl and P. Žid. Range image segmentation by curve grouping. In A. K. Jain, S. Venkatesh, and B. C. Lovell, editors, *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume 2, pages 985–987. IEEE Press, 1998.
- [5] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.
- [6] X. Jiang and H. Bunke. Fast segmentation of range images into planar regions by scan line grouping. *Machine Vision and Applications*, 7(2):115–122, 1994.
- [7] X. Jiang, H. Bunke, and U. Meier. High-level feature based range image segmentation. *Image and Vision Computing*, 18(10):817–822, July 2000.
- [8] I. Khalifa, M. Moussa, and M. Kamel. Range image segmentation using local approximation of scan lines with application to CAD model acquisition. *Machine Vision and Applications*, 13(5–6):263–274, March 2003.
- [9] P. C. Leger. Fast planar segmentation of range data for mobile robots. United States Patent, 5'978'504, November 1999.
- [10] E. Natonek. Fast range image segmentation for servicing robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 406–411, May 1998.
- [11] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart. A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2005.
- [12] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg. Aufbau eines 3D-Laserscanners für autonome mobile Roboter. Technical Report GMD-Report 126, GMD - Forschungszentrum Informationstechnik, Germany, 2001.
- [13] C. Ye and J. Borenstein. Characterization of a 2-D laser scanner for mobile robot obstacle negotiation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2512–2518, May 2002.