2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Quadtree Sampling-based Superpixels for 3D Range Data

Jaehyun Park[1] , Sunglok Choi[1] , and Wonpil Yu[1]

***Abstract*— 3D range sensors are currently being used in various fields. Creating 3D range sensors requires various techniques, such as object detection, tracking, classification, 3D SLAM, etc. For the pre-processing step, superpixels can improve the performance of these techniques. This paper proposes a novel over-segmentation algorithm, known as superpixels, for 3D outdoor urban range data. Superpixels are generated with three steps: boundary extraction using a surface change score and sensor models, initial cluster seeding using a quadtree decomposition, and iterative clustering, which adapts a k-means clustering approach with limited search size in the quadtree dimension. The proposed algorithm produces adaptive superpixel sizes that take into account surface and object border information. This reduces memory size more than regular grid methods and represents small objects well with adaptable pixel sizes. The algorithm is verified using the publicly available Velodyne dataset and the manually annotated ground truth. A comparison with the conventional algorithm is also presented.**

I. INTRODUCTION

With the growth of 3D range sensors (e.g., multi-beam LiDARs, TOF cameras, and depth cameras), a great deal of research has been conducted on indoor and outdoor 3D perception in many fields. In particular, to understand outdoor scenes, the necessity of 3D sensors has been increasing since the success of the DARPA Grand and Urban Challenges as well as the appearance of the Google self driving cars [1]. The 3D sensors provide spatial information, such as 3D point clouds, and camera based sensors also give depth and color information in 2D images [2]. The 3D information helps robots or intelligent vehicles to understand the real world more precisely.

3D perception is fundamental in robotics and autonomous vehicles. Many techniques (e.g., object detection, classification, tracking, etc.) can be used to understand 3D environments. In order to implement these techniques, segmentation is usually used for the pre-processing step, which improves the performance of perception systems [3], [4]. In the field of computer vision, many researchers have attempted to divide meaning segments from image pixels to improve performance [5–7]. The aim of this paper is to generate superpixels from 3D range data for the pre-processing step. Superpixel algorithms are usually applied to 2D images or RGB-D images. Recently, many state-of-the-art superpixel algorithms such as Simple Linear Iterating Clustering (SLIC) [5], Superpixels Extracted via Energy Driven Sampling (SEEDS) [6], and Depth Adaptive Superpixels (DASP) [7] have been proposed for 2D images and RGB-D images. For range images extracted from 3D LiDARs, there are currently no appropriate cases to apply superpixel algorithms. Since outdoor environments are unstructured and more complex than indoor environments, it is difficult to segment objects. Therefore, this paper proposes a novel superpixel algorithm, Quadtree Sampling-based Superpixels (QSS), for a complex urban range scene. The key contributions of our paper are as follows:

1) A superpixel algorithm for an outdoor urban range scene
2) A dynamic cluster seeding method with an adaptive superpixel size using a quadtree decomposition that takes into account surface and object boundary information
3) An iterative clustering approach within dynamic search sizes based on a quadtree dimension of seeds

We evaluate our algorithm compared to other 3D LiDAR based over-segmentation algorithms using standard error metrics. Fig. 1 shows the result of our superpixel algorithm for an outdoor urban range scene and raw point clouds. The size of the superpixels is relative to the surface complexity.

The remainder of this paper is organized as follows. Section II presents an overview of previous works on superpixel algorithms for 2D, RGB-D, and 3D range data. Section III introduces our proposed quadtree sampling-based superpixel algorithm, and Section IV explains our experimental results and verifies our proposed algorithm compared to state-of-the-art algorithms. Finally, we conclude in Section V.

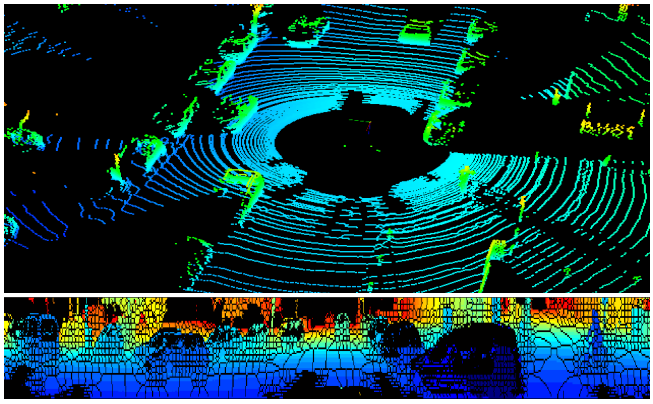


Fig. 1: The proposed superpixel algorithm. **Top row**: input 3D point clouds. **Bottom row**: a superpixel result from the range image colored by distance; black pixels are missing measurements and superpixel boundaries.

* This work was supported partly by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Institute for Advancement of Technology (KIAT). (Project: 3D Perception and Robot Navigation Technology for Unstructured Environments, M002300090)

[1]Jaehyun Park, Sunglok Choi and Wonpil Yu are with the Intelligent Cognitive Technology Research Dept., ETRI, Daejeon, Korea {jaehyun, sunglok, ywp}@etri.re.kr

II. Related Work

A. 3D Range Sensors

3D range sensors are currently being utilized in many fields. The 3D range sensors that are mainly employed are 3D LiDAR, stereo camera, TOF camera, RGB-D camera, etc. The TOF camera and RGB-D camera have short distance ranges and are overly sensitive to light; thus, these cameras are usually used in indoor environments. Stereo cameras also have the problem of light sensitivity and provide inaccurate depth values. 3D LiDAR sensors, which have long distance ranges and give precise depth values, have been used in many robotic and intelligent vehicle applications. However, 3D LiDARs are currently much more expensive compared to other 3D range sensors. This paper is based on 3D LiDAR, since we target the robots and intelligent vehicles that operate in outdoor urban environments. However, our algorithm could be utilized for other 3D sensors as well. We chose range images to represent the 3D range data, since they enable us to borrow ideas from the vision sector and give structured range data.

B. Previous Work

Superpixel algorithms aim to group pixels in images into perceptually meaningful regions. They have been used in many computer vision fields, since they greatly reduce the complexity of subsequent image processing tasks. The generally desirable properties of superpixels include how well they adhere to image boundaries and their computational complexity reduction as a pre-processing step. Superpixel algorithms can be broadly categorized as either graph-based or gradient ascent methods. Below, we review popular superpixel methods applied to 2D images and 3D range data.

Graph-based superpixel methods consider each pixel as a node in a graph, with edges connecting to neighboring pixels. Edge weights are used to characterize the similarity between the nodes, and superpixel labels are solved for by minimizing a cost function over the graph. The popular graph-based algorithm for 3D range data is an extension of the Felzenszwalb and Huttenlocher (FH) graph-based algorithm that uses a minimum spanning tree to cut the graph edges [8]. The FH algorithm has been utilized in many robotic fields as a pre-processing step, since it has near linear computation time. Jonatan et al. applied the FH algorithm to textured dense 3D point clouds that fused color and laser range data using a Markov random field model [9]. They utilized a weighted combination of Euclidean distance, pixel intensity difference, and the surface normal divergence to cut edges. Other papers have applied the FH algorithm similarly [10], [11]. However, most algorithms have focused on full segmentation over the entire range data rather than superpixels alone.

Gradient ascent-based algorithms iteratively refine the cluster until some convergence criterion starting from initial seeds. In particular, a recently proposed SLIC algorithm that has been utilized in many applications generates superpixels quickly and provides qualitative results. The SLIC algorithm uses a local k-means clustering approach to find superpixels efficiently, clustering pixels in the five-dimensional space of the color and pixel's location. DASP extended this idea for RGB-D images, expanding the clustering space of the added dimensions of depth and point normal angles with a seed point sampling algorithm. Recently, the Voxel Cloud Connectivity Segmentation (VCCS) algorithm was proposed to generate supervoxels in 3D point clouds [12]. VCCS also extended the ideas of SLIC to use colored voxel clouds extracted from RGB-D sensors, expanding the clustering space with the 39 dimensions of color, spatial, and fast point feature histograms. While DASP and VCCS are efficient and give promising results, they focus only on structured indoor scenes with simple objects.

III. Quadtree Sampling-based Superpixels

In this section, we present a Quadtree Sampling-based Superpixels (QSS) algorithm to generate superpixels from range images. The range images are representations of 3D range data from 3D LiDAR, with a particular focus on outdoor environments. QSS uses a variant of k-means clustering with a limited search size similar to SLIC, DASP, and VCCS. Two important distinctions are as follows:

1) The seeding of superpixel clusters is done by a quadtree decomposition-based sampling method, which provides adaptive superpixel sizes and takes into account object surfaces and boundaries. This is more memory efficient than regular grid sampling, and it adheres well to the boundaries of small objects.
2) An adaptive search size is considered by using a dimension of quadtree during iterative clustering. This provides a faster iteration speed than using the fixed search space.

We describe the QSS algorithm below. In III.A, we introduce a quadtree decomposition for seeding superpixel clusters. In III.B, we present how the quadtree decomposition is determined. In III.C, we discuss the distance measure for clustering. Finally, in III.D, we describe the iterative clustering algorithm with adapted superpixel sizes.

A. Quadtree Decomposition

The clustering algorithm begins with initialization, in which a number of cluster seeds that are in the center of the superpixels are selected. The most common seed selection method involves sampling the center position at regular grid to divide the image with a chosen grid size. This method always selects grid size according to images provided by the user. This paper proposes a flexible superpixel size selection method using a quadtree decomposition. Generally, the number of pixels representing objects in range images differs with distance even if the objects are the same size. Close objects are represented with more pixels than distant objects. If an image is divided at a regular grid size to generate superpixels, close objects should be well divided, while distant objects can be joined if the grid size is bigger than the object size. Conversely, if a small grid size is applied for small objects, this can generate superpixels efficiently.

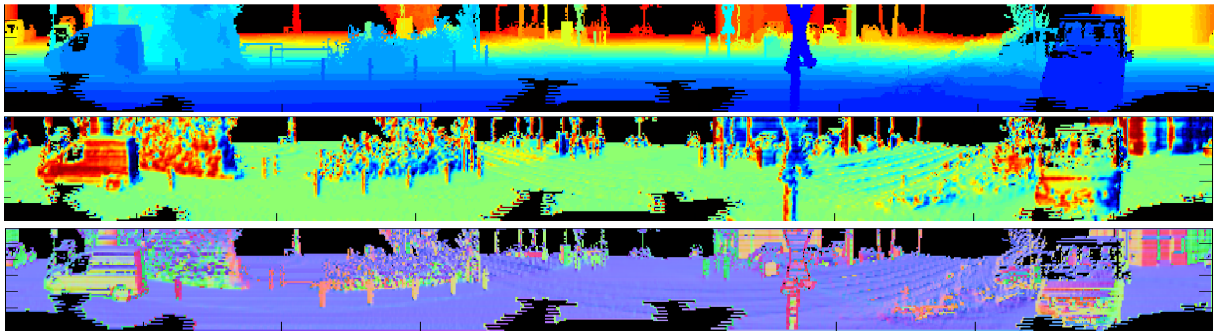


Fig. 2: An input range image and calculated feature images. Top row is an input raw range image colored by distance magnitude. Blue is close, and red is far. Middle row is a computed GD image; colors are mapped to GD angles. Bottom row is a surface normal image, with each normal color coded in RGB color.

However, many superpixels are generated to represent close objects that have many pixels. Therefore, we aim to generate the adaptive number of superpixels that reduces memory to represent an image using superpixels.

This paper uses a quadtree to decompose range images with adaptable cells that are utilized by superpixel seeds. The quadtree is a tree data structure in which one tree node has four children [13], [14]. This is usually utilized representing 2D space in which it is recursively subdivided into four quadrants until a limited size of a region. The important property of the quadtree is that it decomposes space into adaptable cell sizes. We use object boundaries extracted from range images for the criterion of decomposition. Small superpixels will be generated in boundary regions, while large superpixels will be produced in non-boundary regions in which there are flat spaces or large objects. The next section, Section III.B, describes how to extract boundaries from range images.

B. Border Extraction for Quadtree Decomposition

This paper uses object border information for the criterion of a quadtree decomposition. Two methods for border extraction from a range image are proposed: the surface change scoring method according to the range value and gradient direction in the range image and the border extraction method that uses only the range value considering sensor specification.

a) Border extraction using surface change scoring: First, we describe the surface change scoring method. This utilizes the difference in range values and the difference in gradient directions (GDs) from the range pixels. Range values are directly obtained from range images. A GD is a 1D feature in degree within the ranges [-180, 180], which is able to utilize representing surfaces similar to surface normals in 3D point clouds [15]. The gradient of an image is represented as follows:

$$\nabla f = \left[\frac{\delta f}{\delta x}, \ \frac{\delta f}{\delta y} \right] \tag{1}$$

where $\delta \mathrm{f}/\delta \mathrm{x}$ and $\delta \mathrm{f}/\delta \mathrm{y}$ are the gradients in the x and y directions and GD, θ_{GD}, is given by

$$\theta_{GD} = \tan^{-1}\left(\frac{\delta f}{\delta x} / \frac{\delta f}{\delta y} \right) . \tag{2}$$

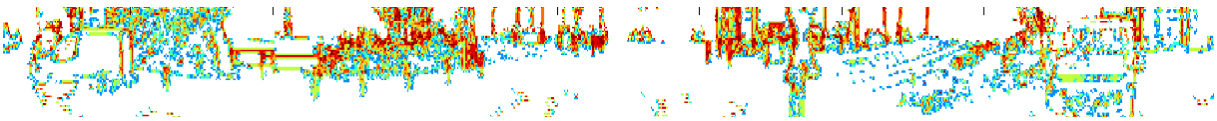
We simply applied a [-1, 0, 1] filter mask on the range image except at the start and end of the row and column, where we applied a [-1, 1] filter mask. The [-1, 1] filter mask shifts the image by half a pixel; however, the [-1, 0, 1] filter mask does not shift the image. The middle row of Fig. 2 shows a GD image from a raw range image in the top row of Fig. 2. The GD image shows a similar specification compared to the surface normals shown in the bottom row of Fig. 2; surface normals are color coded in RGB color space (e.g., flat ground surfaces have the same specification between the GD image and the surface normal image.). However, the GD image from the outdoor range image has a lot of noise, and it affects our ability to find boundaries directly. We applied a bilateral smoothing filter to the GD image to remove noise. The middle row of Fig. 2 is the result of GD after a bilateral smoothing filter.

A score based on range and GD differences is represented as follows:

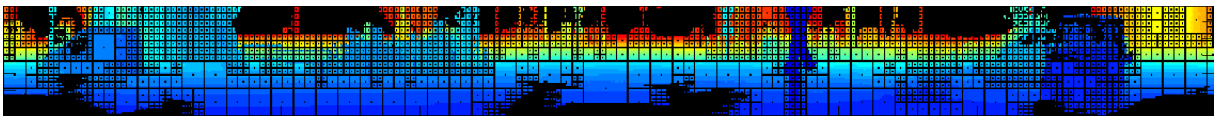
$$Score = \frac{N[(|d_r| < T_r) \ \&\& \ (|d_{GD}| < T_{GD})]}{N_r} \tag{3}$$

where N_r is the number of range pixels in 3 x 3 neighbors, $N[\cdot]$ is the number of sufficient criteria $[\cdot]$, and d_r and d_{GD} are differences in ranges and GD spaces, respectively. T_r and T_{GD} are thresholds that determine how each difference influences a score. Lower thresholds give a high score, which is sensitive to surface changes. In our implementation, we selected $T_r = 5$ in meters and $T_{GD} = 0.5$ in radians. A high score indicates borders of objects. The result of the surface change scored image is shown in Fig. 3 (a), and the quadtree decomposition result using the surface change score is depicted in Fig. 3 (b).

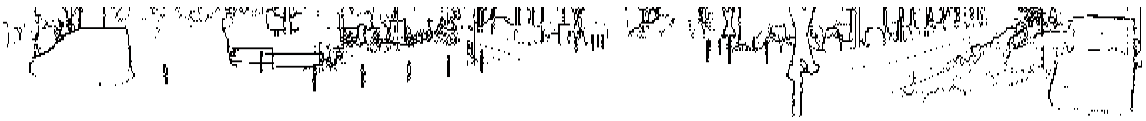
b) Border extraction using sensor specification: Typically, borders appear as non-continuous traversals from foreground to background. There are two kinds of borders in range images: object borders and shadow borders [16]. In this paper, we are interested in object borders, since each



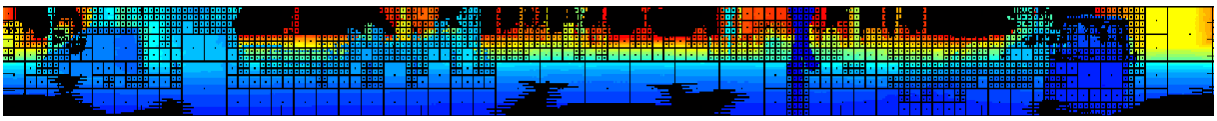
(a) Result of border extraction using surface change scoring. Blue is a low score, and red is a high score.



(b) Result of a quadtree decomposition and seeding clusters center using a result of (a). Black dots are cluster's center position, and squares are size of each cluster.



(c) A border extraction result using sensor specification in III.2.



(d) A quadtree decomposition and seeding of initial seeds based on (c).

Fig. 3: Results of border extraction algorithms and initial cluster seeding using a quadtree decomposition and border information.

border concurrently exists in a range image. The easiest way to find borders in range images is to compare a change in the distance between neighboring range pixels. The criterion is required to determine if it is a border using differences between range values. However, changes are high between neighboring pixels that have far range values. To determine the criterion of the border, we model the range difference between neighboring pixels horizontally and vertically. Generally, range data from 3D LiDARs such as Velodyne sensors is dense horizontally and sparse vertically. The criterion of border, T_{border}, is defined as follows:

$$T_{border}(r_i, r_j) = w_{\{v\},\{h\}} * \min(r_i,\ r_j) \tag{4}$$

where r_i and r_j are the i^{th} and j^{th} range values in a range image, and $w_{\{v\},\{h\}}$is a weight value that has different weights vertically and horizontally. We searched 3 x 3 neighbor pixels and determined if a change of range values between pixels i and j, $d_r(r_i, r_j) := |r_i - r_j|$, was above the criterion T_{border}. If it satisfied the criterion, we marked a pixel with a lower range value as an object border. These extracted borders were utilized as the criterion of the quadtree decomposition. Figs. 3 (c) and (d) show results of range image borders and initial seeds using a quadtree decomposition based on borders.

C. Distance Measure

QSS correspond to clusters in the 3D space, given as

$$F = [r,\ x,\ y]\,, \tag{5}$$

where r is the range value and x, y is the pixel's position. In order to calculate the 3D Euclidean distance in the F space, we must normalize the range proximity and spatial proximity by their respective maximum distance within a cluster. Range distance d_r and spatial distance d_s are normalized by a constant m and quadtree dimensions R_{Dim}. Normalized distance D_F is written as

$$D_F = \sqrt{\frac{w_r d_r^2}{m^2} + \frac{w_s d_s^2}{R_{Dim}^2}}, \tag{6}$$

where w_r and w_s are weights to control the influences of each distance.

D. Clustering

Iterative clustering is processed using assigned seed centers. A clustering process is similar to SLIC, which uses an iterative k-means clustering process with a limited search radius. Each pixel is labeled to the nearest cluster center using (6). The main distinction compared to other algorithms is adaptive search sizes according to superpixel sizes, which are quadtree dimensions. In our implementation, search sizes are 2 Dim($seed_i$) x 2 Dim($seed_i$). Iteration is done iteratively until the cluster centers converge or for a fixed number of iterations. In practice, we found that less than five iterations are sufficient for most range images with large superpixel sizes. Surprisingly, generating over 800 superpixels in 64 x 870 range images, clustering is done within one or two iterations.

IV. Experiments

A. Datasets

This paper uses the publicly available Velodyne datasets [17], [18] to evaluate our algorithm and to compare it with

Fig. 4: Front camera view of experimental data.

state-of-the-art superpixel algorithms. The dataset, recorded with the Velodyne HDL64E-SE in complex urban environments, provides range images in 64 x 870 pixels, where each row corresponds to the measurements of one specific laser beam. Segmenting complex urban scenes is extremely challenging. Since there are many obstacles (e.g., houses, fences, trees, traffic signs, pedestrians, plants, cars, etc.), it is difficult to group pixels into segments. Fig. 4 illustrates the front camera view of the experimental data in scenario 1 of the datasets. However, this dataset does not provide ground-truth labels, just raw Velodyne datasets. To compare our algorithm with the state-of-the art algorithm, we made 10 manually labeled ground-truth images. Examples of ground-truth scenes are shown in Fig. 7. All ground-truth datasets and experimental results are presented on our project site[1].

TABLE I: Paramerts for Each Algorithms

	FH	**nSLIC**	**QSS**
$\mathbf{w_r}$	0.3	2	1
$\mathbf{w_s}$	-	1	2
$\mathbf{w_n}$	150	5	-

B. Error Metrics

We used standard metrics (boundary recall and under-segmentation error) to evaluate the performance of the superpixel algorithms. Boundary recall is the fraction of ground-truth edges that are within a certain distance of a superpixel boundary [19–21]. High boundary recall indicates that the superpixels properly follow the edges of objects in the ground-truth labeling. Boundary recall is formulated as

$$R = \frac{TP}{TP + TN} \tag{7}$$

where TP (true positive) represents the number of ground-truth boundary pixels for which there exists a superpixel boundary in range. FN (false negative) represents the number of ground- truth boundary pixels for which there does not exist a boundary pixel in range.

Under-segmentation error measures whether a superpixel overlaps with more than one object. Various models are used to measure this. We used the free parameter model used in [21] expressed as

$$\mathrm{U} = \frac{1}{N}\left[\sum_{S \in GT}\left(\sum_{P:P \cap S \neq \emptyset} min(P_{in},\ P_{out})\right)\right] \tag{8}$$

[1]Homepage:http://sites.google.com/site/qssuperpixel

where N is the number of labeled ground-truth pixels, GT is the number of ground-truth segments, and S the number of output segments of the superpixel algorithm. Ground-truth segments divide a superpixel P into a P_{in} and a P_{out} part.

C. Results

We performed a quantitative comparison of two state-of-the-art superpixel methods: FH graph-based segmentation for 3D LiDAR and SLIC (nSLIC) with points and surface normals. We modified the distance measure of each algorithm suited to 3D LiDAR data. The superpixel results of each algorithm are shown in Fig. 7, and the comparison of the error metrics is presented in Fig. 5. The parameters used in each algorithm are shown in Table I. We explain the comparison algorithms briefly below.

In the FH graph-based algorithm as [8], we used a weight using range values and surface normals as follows:

$$w_{i,j} = w_r d_r + w_n d_n \tag{9}$$

w_{ij} is an edge between node i and node j, d_r is a range difference, and d_n is a normal distance represented by $1 - n_i^T n_j$. w_r and w_n are range and normal weights that are hand-tuned for this implementation as shown in Table 1, respectively. In our results, FH algorithm shows the best results in under-segmentation error and poor results in boundary error. It produces large irregular shaped segments as shown in Fig. 7. Using our dataset, the FH algorithm produced many small segments when more than 1,500 superpixels in 64 x 870 a range image. We do not represent these results in the figure since it is not a suitable result to evaluate data.

nSLIC is expended in the distance measure using point positions and surface normals similar to DASP's distance measure. nSLIC provides compactness and the expected superpixel size. Since there are many small objects in outdoor environments, small objects are merged to one segment. This is an important reason of under-segmentation error.

Our algorithm divided two methods, QSS with a surface change score and with a border model. Fig. 7 shows examples of QSS with a surface score and the parameters represented in Table 1. The methods showed similar performance despite their differences in cluster seeding. The two QSS methods are the best overall performance. They perform the best in boundary recall and second in under-segmentation error. Surprisingly, a clustering process is done within one or two iterations when generating over 800 superpixels in 64 x 870 range images as shown in Fig. 6, since range images have continuous values in most short range different than RGB images. If the size of superpixel increases, the number of iterations is reduced. Our algorithm is more memory efficient than a regular grid sampling method such as SLIC. If a minimum dimension of the quadtree is a 4 x 4 region, our algorithm produced 2,300 superpixels, while 3,700 superpixels generated by SLIC.

V. CONCLUSIONS

This paper presents QSS, a novel over-segmentation algorithm known as superpixels for the outdoor urban range

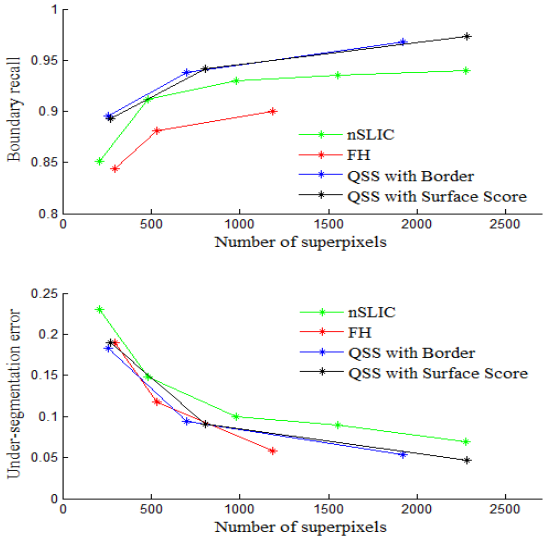


Fig. 5: Boundary recall and under-segmenation error of each algorithm.

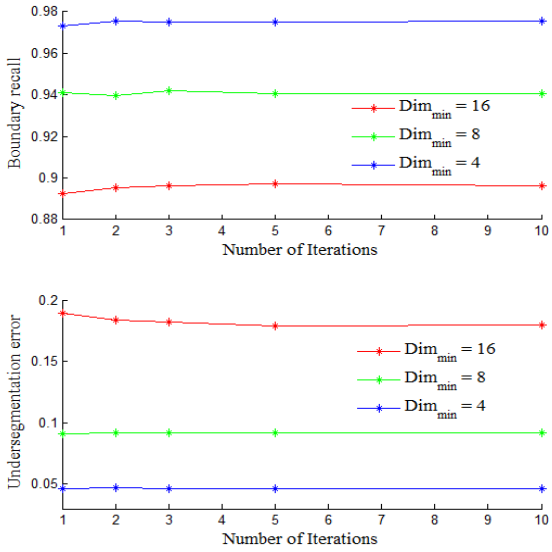


Fig. 6: Convergence of boundary recall and under-segmenation error according to the number of iterations. Dim_{min} means a minimum dimension of the quadtree.

scene. Superpixels are generated with three steps: boundary extraction using a surface change score and sensor models, initial cluster seeding using a quadtree decomposition based on borders, and iterative clustering, which adapts a k-means clustering approach with limited search sizes in the quadtree dimension. The proposed algorithm produces adaptive superpixel sizes that take into account surface and object border information. This reduces the memory size more than regular grid methods and represents small objects well with adaptable pixel sizes. The algorithm is verified using the publicly available Velodyne dataset and the manually annotated ground truth. We used standard metrics (boundary recall and under-segmentation error) to evaluate the performance of the superpixel algorithms. Our algorithm showed better performance than other state-of-the-art algorithms in quality and gives significant results with one or two iterations in our most implementations.

ACKNOWLEDGMENT

The authors would like to thank Dr. Byungjae Park for his helpful comments on this paper.

REFERENCES

[1] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[2] J. Jeong, H. Shin, J. Chang, E.-G. L. S. M. C. K.-J. Yoon, and J. il Cho, "High-quality stereo depth map generation using infrared pattern projection," *ETRI Journal*, vol. 35, no. 6, pp. 1011–1020, December 2013.

[3] D. Wang, I. Posner, and P. Newman, "What could move? finding cars, pedestrians and bicyclists in 3d laser data," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.

[4] D. Held, J. Levinson, and S. Thrun, "Precision tracking with sparse 3d and dense color 2d data," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[5] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[6] M. V. den Bergh, X. Boix, G. Roig, B. de Capitani, and L. J. V. Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *ECCV (7)*, 2012.

[7] D. Weikersdorfer, D. Gossow, and M. Beetz, "Depth-adaptive superpixels," in *21st International Conference on Pattern Recognition*, 2012.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, 2004.

[9] J. R. Schoenberg, A. Nathan, and M. E. Campbell, "Segmentation of dense range information in complex urban scenes." in *Proc. International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010.

[10] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3D laser point clouds," in *Proc. International Conference on Intelligent Robots and Systems (IROS)*, October 2010.

[11] X. Zhu, H. Zhao, Y. Liu, Y. Zhao, and H. Zha, "Segmentation and classification of range image from an intelligent vehicle in urban environment," in *Proc. International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[12] J. Papon, A. Abramov, M. Schoeler, and F. Wrgtter, "Voxel cloud connectivity segmentation - supervoxels for point clouds," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013.

[13] R. A. Finkel and J. L. Bentley, "Quad trees: A data structure for retrieval on composite keys," *Acta Inf.*, vol. 4, pp. 1–9, 1974.

[14] H. Samet and R. E. Webber, "Storing a collection of polygons using quadtrees," *ACM Trans. Graph.*, vol. 4, no. 3, pp. 182–222, July 1985.

[15] B. Enjarini and A. Gräser, "Planar segmentation from depth images using gradient of depth feature," in *Proc. International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[16] B. Steder, R. Bogdan, R. Kurt, and K. W. Burgard, "Point feature extraction on 3d range scans taking into account object boundaries," in *Proc. International Conference on Robotics and Automation (ICRA)*, 2011.

[17] F. Moosmann and C. Stiller, "Velodyne SLAM," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, June 2011.

[18] ——, "Joint self-localization and tracking of generic objects in 3d range data," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.

[19] X. Ren and J. Malik, "Learning a classification model for segmentation," in *In Proc. 9th Int. Conf. Computer Vision*, 2003, pp. 10–17.

[20] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *In ECCV*, 2010.

[21] P. Neubert and P. Protzel, "Superpixel benchmark and comparison," in *Proc. of Forum Bildverarbeitung, Regensburg*, 2012.

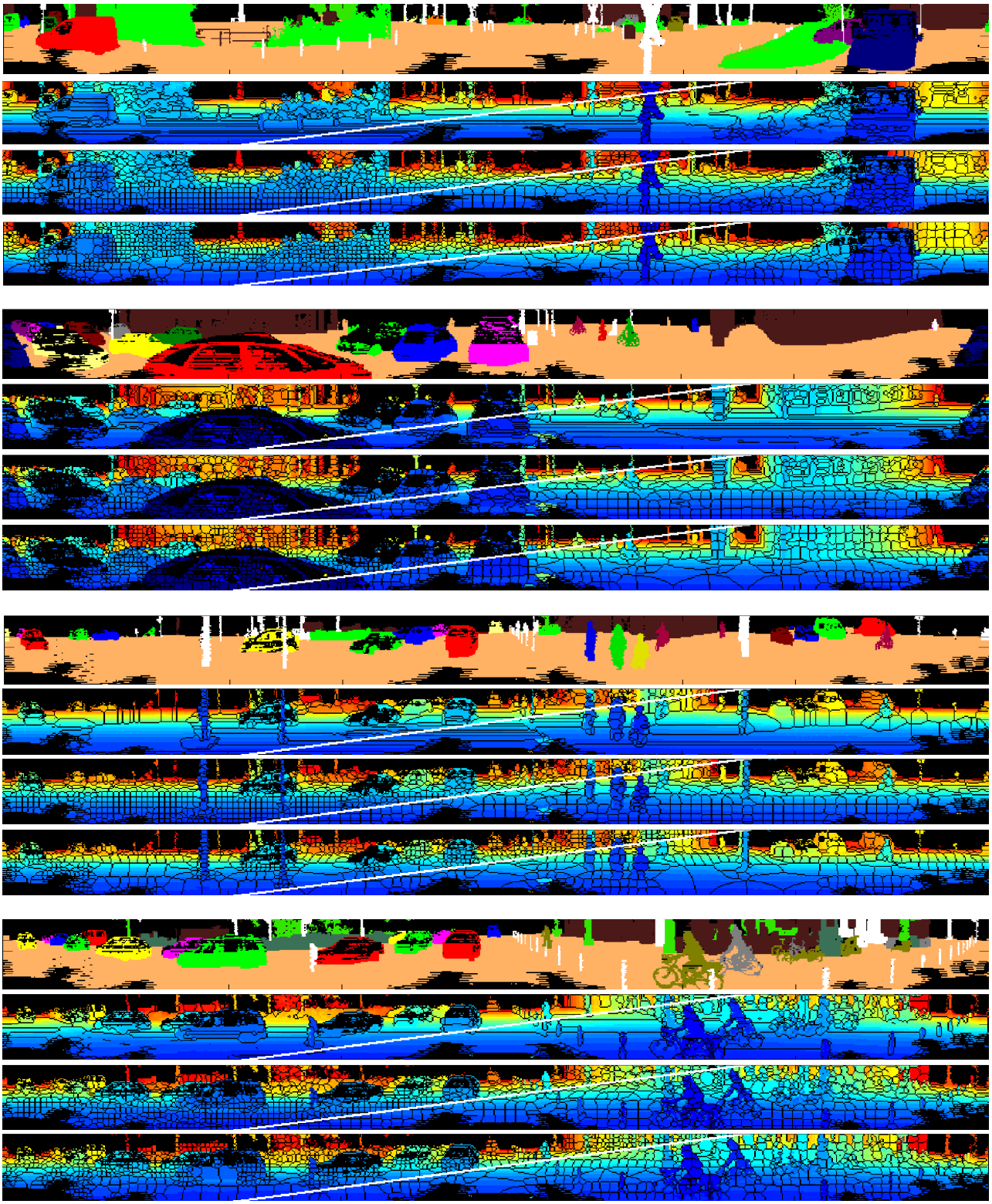


Fig. 7: Examples of superpixels produced by various methods. From top to bottom: ground truth, FH, nSLIC, and QSS. Each image is shown with two different superpixel sizes; the average superpixel size in the upper left is 2000 pixels and is 800 pixels in the lower right. FH generates about 800 pixels in our implementation.