Computing Object-based Saliency in Urban Scenes Using Laser Sensing

Yipu Zhao[1], Mengwen He[1], Huijing Zhao[1], Franck Davoine[2], and Hongbin Zha[1]

***Abstract*— It becomes a well-known technology that a low-level map of complex environment containing 3D laser points can be generated using a robot with laser scanners. Given a cloud of 3D laser points of an urban scene, this paper proposes a method for locating the objects of interest, e.g. traffic signs or road lamps, by computing object-based saliency. Our major contributions are: 1) a method for extracting simple geometric features from laser data is developed, where both range images and 3D laser points are analyzed; 2) an object is modeled as a graph used to describe the composition of geometric features; 3) a graph matching based method is developed to locate the objects of interest on laser data. Experimental results on real laser data depicting urban scenes are presented; efficiency as well as limitations of the method are discussed.**

I. INTRODUCTION

A. Background

In recent years, technologies using laser range sensors such as LiDAR are developing quickly. They can be used for the perception of the environment using a sequence of 2D or 3D laser points (see Fig. 1). Through a bottom-up procedure, laser data can be first processed to find data clusters or segments, i.e., laser points that are most likely to be the measurements of the same objects. These clusters or segments can then be recognized as certain kinds of objects, e.g., planar surfaces [1], straight objects [2] etc. Golovinskiy [3] discusses the processing of airborne laser scan data to classify small objects, such as posts, lights, cars, etc., in an urban environment. The problem is solved by localization, segmentation, representation and classification procedures in a sequential manner. Most of these studies process the entire laser sensing data of scenes in bottom-up ways. As the amount of laser points gets larger, computational cost grows exponentially. However, a robot normally has limited computational resources, while requires real-time processing. Moreover, many tasks concern not all the objects in a scene, but some specific ones, which mean that universally processing the whole data set is a waste of resource. In many applications, it is required to allocate more computational resources on objects of interest.

B. Related Work

1) Saliency: The concept of saliency has been introduced early in the field of computer vision. A salient object is something obvious to the vision system. Biologically inspired methods have been developed to detect salient objects from a visual environment [4], [5]. Many of the visual saliency methods pre-filter a scene by decomposing it into basic features [6] and recombining them to a saliency map, which contains high activation at regions that differ strongly from the surroundings [7]. However, the majority of attention systems do not show real time capability [8] or are tested in controlled indoor environment [9]. Moreover, the methods that are developed for 2D visual image processing may not be directly applied on 3D laser sensing data for the following reasons. First, comparing with visual attention systems, laser sensing data lacks color information. This makes some human-inspired features and cues invalid. New descriptors for learning saliency on laser sensing data should be developed. Second, it is impractical to train classifiers on all possible objects in urban scenes, because sizes, poses and geometric details differ in a single class of objects.

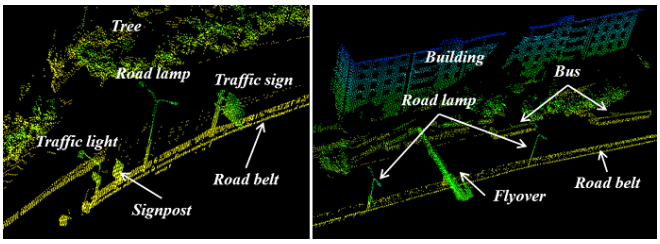


Fig. 1. Geo-referenced laser points[3], providing a low-level geometric representation of the environment. Colors denote for the altitude level. For visualization, the laser points on the road are removed.

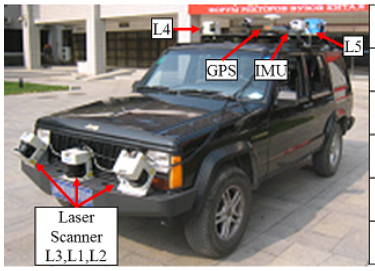
2) Object detection in 3D laser data: There is a huge body of work in the area of object detection in 3D laser data. Most of them follow a bottom-up procedure, where the entire 3D dataset is first segmented into small parts, then classified as objects. The segmentation is formulated as a graph min-cut problem [11] and the entire 3D laser data is considered as a k-nearest neighbors graph. To model complex spatial and relational scenes such as urban environments, advanced graph models such as Associative Markov Networks [12], Markov Random Fields [13] and Conditional Random Fields [14] are also introduced in 3D laser data segmentation. For classification, features such as spin-images [15], tensor voting [16] and point feature histograms [17] are applied. As above methods segment and classify the entire 3D laser data, they meet the problem of great computation when working on large dataset. The computation can be reduced by adding a pre-filter, which only leaves laser points that belong to the task-related objects.

*This work was supported by the NSFC Grants (No.90920304, No.60975061 and No.61161130528).

[1]Y. Zhao, M. He, H. Zhao, and H. Zha are with the State Key Lab of Machine Perception (MOE), School of EECS, Peking University, Beijing, P.R. China zhaoyp at cis.pku.edu.cn

[2]F. Davoine is with the CNRS LIAMA Sino French Laboratory, Beijing, P.R. China. franck.davoine at gmail.com

[3]rendered with PCL (Point Cloud Library) [10]



Laser #	Scan Mode	Major Purpose
L1	Horizontal	SLAM with MODT
L2	Downward	Road geometry
L3	Upward	Objects above the road
L4	Vertical Right	Objects to the right of the car
L5	Vertical Left	Objects to the left of the car

Fig. 2. An intelligent vehicle with multiple single-row laser scanners.

C. Contribution

In this work, the problem of computing object-based saliency in the laser data is formulated as searching for specific compositions of geometric features. There are three reasons supporting our method: First, low-level geometric features are easy to extract from laser sensing data. Second, many man-made objects in urban environment can be represented using a set of geometric features. They are discriminative to different classes of objects, while, although the objects in a single class differ in their details, many of them can be abstracted into the same composition of low-level geometries. Moreover, extensions to other objects are easy. Being a descriptive representation rather than an exact one, the geometric representations are generated without training on lots of samples.

Below, we emphasize the major contributions of this research. 1) A method for extracting low-level geometric features from laser data is developed, where both range images and 3D laser points are analyzed; 2) an object is modeled as a graph used to describe the composition of geometric features; 3) a graph matching based method is developed to locate the objects of interest on laser data.

The rest of the paper is organized as follows. In section II, a brief introduction is given to our vehicle-borne laser sensing system, as well as an outline to this research. Description of the object-based saliency computation is described in section III and IV. Experimental results and discussions are given in section V, followed by a conclusion and future works in section VI.

II. SYSTEM OUTLINE

In this section, the vehicle-borne laser sensing system is introduced, followed by an outline of the algorithm developments.

A. Vehicle-borne Laser Sensing System

In our previous research [18], a vehicle-borne laser sensing system was developed as shown in Fig. 2. Five single-layer laser scanners are used, collecting range data of surrounding environments at different directions. A GPS/IMU navigation unit is used to provide vehicle pose as the vehicle moves along streets. Calibration is conducted previously to find the geometric parameters from each laser scanner to a vehicle reference frame. In this research, experiments are conducted using the top-right (L4) and top-left (L5) laser scanners, where by integrating calibration parameters and vehicle pose, range measurements can be geo-referenced into a global Cartesian coordinate system. As a result, a 3D representation of the environment during a time internal is generated as shown in Fig. 1.

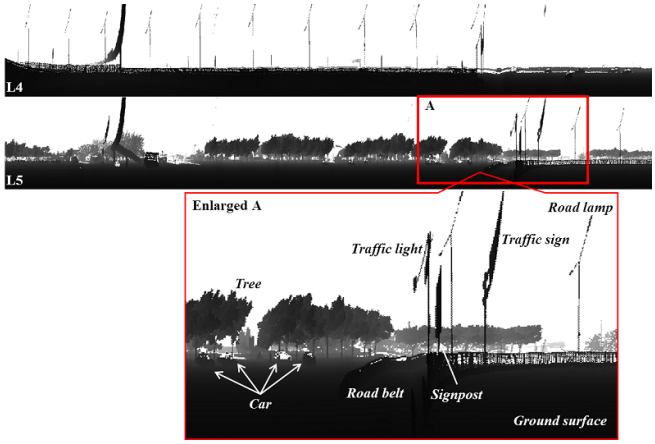


Fig. 3. Range images given by laser scanners L4-L5 as well as an enlarged figure reflecting a complex scene. The horizontal axis is the scan line number ($\propto$ time), and the vertical strips represent the range values of each laser scan.

B. Research Outline

Given a stream of range measurement and a set of predefined objects of interest (OI), the objective is to estimate a series of saliency maps, where the objects of interest are highlighted on their region. In this research, we choose the form of range image as an interface for data representation, while estimations are conducted in both 2D and 3D. The processing flow is depicted in Fig. 4. For the reason of computation cost, the points on the ground surface are removed from the range image through a pre-processing by putting thresholds on their elevation values. Given a range image, four types of geometric features, which are the basic components of many objects, are extracted. Object-based saliency is then computed, where potential objects (PO) are represented with graphs, and are matched with those graphs of objects of interest. This approach will finally return a series of object saliency maps, in which regions of objects that match well with objects of interests are highlighted.

III. GEOMETRIC FEATURE EXTRACTION

In this section, we first define four types of geometric features (called GFs in the rest of the paper) that are used in this work, followed with an introduction of our approach on geometric feature extraction.

A. Geometric Features

Geometric features are the basic components of artificial objects. For example, a traffic sign in most countries may be a vertical line with a triangular or rectangular plane on the top; a car is a combination of several planes [19]; the main parts of a street light are a long vertical line and some short horizontal lines and planes above it. Four GFs are selected in this research to represent objects. They are vertical planes and lines, horizontal planes and lines. Certainly several details of object shape and contour cannot be represented by

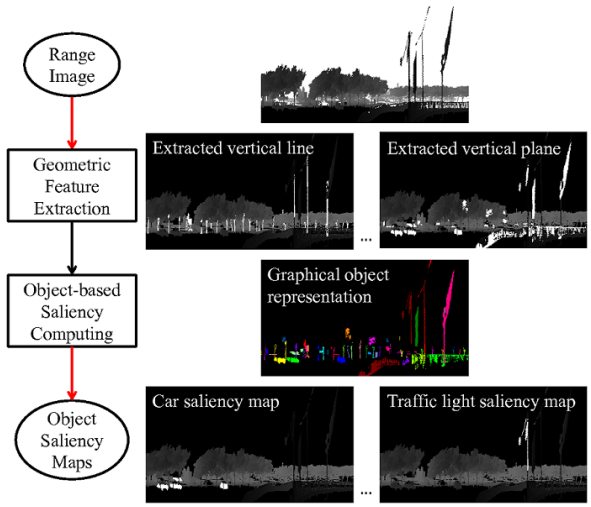


Fig. 4. System flowchart.

such limited types of GFs. But the objective of our approach is to compute object-based saliency, instead of finding an exact representation of objects. Such features are useful in grasping the major geometric primitives of many human-made objects.

Note that in this work, we do not consider the nature objects such as trees and bushes, which are difficult to be represented using a limited number of geometric primitives. Modeling and discovering vegetation will be studied in our future work.

B. Related Work on Geometric Feature Extraction

Before introducing the GF extraction method used in our approach, we give a brief review to previous works. Several methods to extract GFs (e.g. line, plane) from 3D point clouds have been proposed, relying for example on Hough transform [20] or RANSAC [21] as parameter estimators. Both of them have been proven to be reliable even in noisy environments. However, none of them fits the condition for our objective. First, the high memory requirement of Hough transform makes it hard to be applied on many robots that have limited memory. Second, RANSAC based approaches may be too time consuming for our objective to pre-filter the laser range data before saliency computation.

Region grow method has been widely applied in shape detection [22]. Generally, region growing starts with selecting some seeds in the input data. A target shape is extracted based on the seeds. Then the seeds keep growing into neighboring data as long as the target shape fit well with it. The two advantages of region growing make it suitable in our approach. First, it fits well with the 2D data representation (i.e. range image) used here. The easy-acquired 2D neighbors in range image can be taken as a strong prior for finding neighbors in corresponding 3D laser data. So the computation cost of finding a region's neighboring data is greatly reduced when works in range image. Second, region growing method does not require high memory requirement. Only local data are stored in memory when growing.

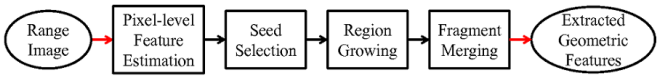


Fig. 5. The flowchart of geometric feature extraction method.

Region growing for GF extraction introduces the following two problems: seed selection and objective function to guide growing. Below we will address these details.

C. Geometric Feature Extraction

The flowchart of GF extraction method is shown in Fig. 5. Details of the four steps—pixel-level feature estimation, seed selection, region growing and fragment merging are given below.

1) Pixel-level features: Several kinds of features are extracted from the range image after removing the ground pixels. They are listed in Table I. Before estimating these features, principle component analysis [23] is performed in a small distance (e.g. 0.5m) surrounding each valid pixel (i.e. laser point). For each laser point, the estimated vector with the largest eigenvalue is defined as the principal direction, and the third one, with smallest eigenvalue, is defined as the normal.

The distributions of normal and principal directions in the 3D neighborhood (e.g. less than 0.7m) of each valid pixel are then considered. For each valid pixel, the distribution of normals is represented by two vectors. The first one is the mode of the distribution, i.e. the modal vector $\vec{MN}$, which is the most likely to be present in the 3D space neighborhood. The second one is the deviation vector $\vec{DN}$, reflecting the standard deviation of surrounding normals in the neighborhood. Similarity, the distribution of principal directions for each valid pixel is represented by the modal vector $\vec{MP}$ and the deviation vector $\vec{DP}$. Moreover, the spatial range of neighborhood in the range image (e.g. 5*5 pixels) of each valid pixel is used. For each valid pixel, the horizontal range RH and vertical range RV are estimated with the 2D range image neighborhood of the pixel.

TABLE I

Description	Count
$\vec{MN}$. modal vector of normal direction distribution, in 0.7m	3
$\vec{DN}$. deviation vector of normal direction distribution, in 0.7m	3
$\vec{MP}$. modal vector of principal direction distribution, in 0.7m	3
$\vec{DP}$. deviation vector of principal direction distribution, in 0.7m	3
RH. horizontal range of laser points, in 5*5 pixels	1
RV. vertical range of laser points, in 5*5 pixels	1
total	14

2) Seed selection: With the pixel-level features defined previously, seeds of each GF type are selected based on a decision tree, which is generated empirically. For example, the points on a plane tend to have a low deviation in the distribution of normals. Meanwhile, the points on a line are expected to have a low deviation in the distribution of principal directions. In vertical and horizontal planes, the modal vector of normal distribution varies. Also in vertical and horizontal lines, the modal vector of principal direction

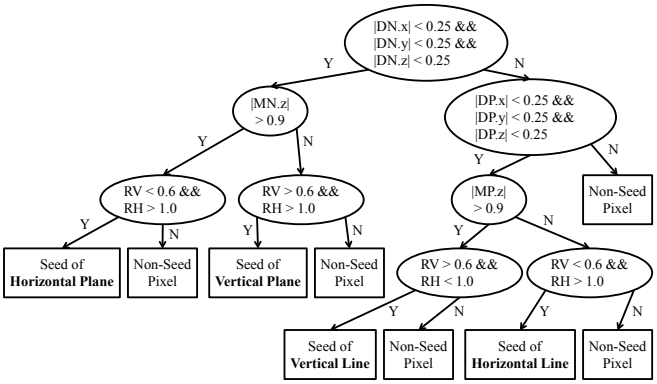


Fig. 6. The decision tree used in seed selection.

distribution varies. Moreover, for points on a vertical geometric feature, the vertical ranges are quite large. Only the points of a vertical line have small horizontal ranges. Based on these rules, a decision tree (see in Fig. 6) is built to select seeds.

3) Region growing: Region growing is conducted for each GF type, started from the seeds. Criterion to allow new pixels joining a region is designed for each GF type, and in order to avoid over growing, strict thresholds are used.

The order of GF types for region growing influences the results. If the line-type GFs are grown first, some planes could be extracted as a series of lines. If the plane-type GFs are grown first, some lines could be grown as parts of planes. According to experiments, the following order yield good results: vertical plane, vertical line, horizontal plane and then horizontal line. When the growing of one GF type ends, the corresponding pixels are disabled to avoid them be associated to other GF types in future growing.

4) Fragment merging: Though the over-growing problem is solved using a tight fitting condition, fragments exist in the growing results. To merge these fragments, an iteratively merging process is applied. For every two regions, a merging cost is estimated. If the cost is lower than a threshold (e.g. 1.0), they are merged into a new region, the parameters of which are also estimated. By keep executing the cost estimation and merging steps, the fragments are combined into complete regions.

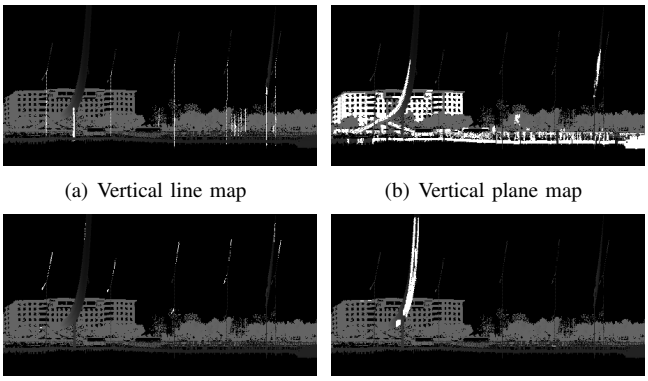
The merging cost C of two regions r_i and r_j is defined below, where n_i, p_i are the normal and principal direction of r_i, n_j, p_j are those of r_j, $dist_{i,j}$ is the Euclidean distance between r_i and r_j, and α is a manually set constant (e.g. 60):

$$C(r_i, r_j) = e^{dist_{i,j} + \alpha * |\tan(ang_{i,j})|} + g_{i,j} - 1 \quad (1)$$

$$ang_{i,j} = \begin{cases} \arccos(n_i \cdot n_j), \text{if } r_i \text{ is a plane} \\ \arccos(p_i \cdot p_j), \text{if } r_i \text{ is a line} \end{cases}$$

$$g_{i,j} = \begin{cases} 0, \text{if } r_i \text{ and } r_j \text{ have the same GF type} \\ +\infty, \text{else} \end{cases}$$

An example of GF maps is shown in Fig. 7, where the laser points that belong to the corresponding type of GFs are highlighted.



(a) Vertical line map (b) Vertical plane map

(c) Horizontal line map (d) Horizontal plane map

Fig. 7. Examples of geometric feature maps.

IV. OBJECT-BASED SALIENCY COMPUTING

As mentioned previously, the combinations of GFs are useful in grasping the major geometries of many man-made objects. To describe such combinations properly and compactly, we propose a graphical object representation. Then all potential objects (PO) in the range image are represented by graphs. Therefore, the object-based saliency computation is converted to a graph matching problem. The object-based saliency of a PO is decided when the correspondence exists between two graphs: the PO's graph and the graph of an object of interest (OI). In the following parts, we first define the graphical object representation, then give the details of graph generation and graph matching.

A. Graphical Object Representation

Graphical object representation is widely used in indoor environments [24] as well as outdoor scenes [25]. Generally, the geometric primitives (e.g. lines, planes) are considered as nodes, while the topologies of geometric primitives (e.g. neighborhood) are represented by edges. Here we define the graphical object representation in a similar way:

1) The node is related to an extracted GF. Its attributes include the type, size and altitude of the GF.

2) The edge represents that the GFs of the two linked nodes are neighboring (i.e. distance of nearest points less than a threshold θ). Its attributes are the vectors that link the center points of the two GFs.

B. Graph Generation

To generate graphs in a range image which contains lots of GFs and objects, the hierarchical clustering method [26] is applied. Each extracted GF is used to initialize a cluster. By finding and merging with neighboring clusters (i.e. distance of nearest points less than θ), the nearby GFs which mostly belong to the same object are combined into one cluster. However, it is hard to find a proper threshold θ that avoids both over-clustering and under-clustering. Observing that GFs at the ground level have higher density than those with higher elevation, θ is set of 0.1m for the GFs on the ground and linearly increase to 0.5 for the others.

A graph is then built for each cluster. Given a cluster containing several GFs, an example of graph generation is

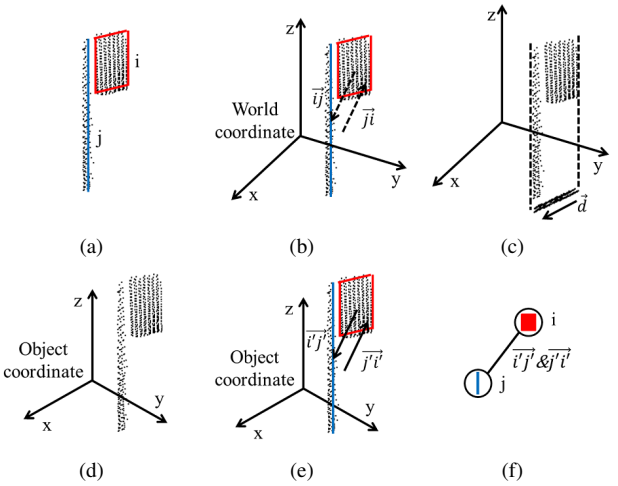


Fig. 8. Examples of graph generation.

given in Fig. 8. Two GFs, i and j, are extracted from the laser data in Fig. 8(a), each represented by a node. As i and j are neighbors, their nodes are linked by an edge. Then we estimate a pair of vectors ($\vec{ij}$ and $\vec{ji}$) between their centers (see in Fig. 8(b)). In order to solve the problem caused by different object poses in world coordinate, a local object coordinate is defined for each object, and the attributes of the graph are calculated in the object coordinate. The laser points that belong to i and j are projected into world's x-y plane first. x axis of the object coordinate is set as $\vec{d}$, which is the principal direction of the projected point-set (see in Fig. 8(c)). Also the z axis of the object coordinate is set the same with that of world coordinate. Therefore, an object coordinate is built in Fig. 8(d). The re-estimated pair of vectors ($\vec{i'j'}$ and $\vec{j'i'}$) are shown in Fig. 8(e), while the graph in Fig. 8(f). Note that the object coordinate is not unique since it relies on the principal direction of point-set, which can be $\vec{d}$ as well as $-\vec{d}$. Therefore when generating graphs of sample objects (in IV-C), we re-estimate the edge vectors in two object coordinates, one (called original coordinate) uses $\vec{d}$ as x axis while the other (called mirror coordinate) uses $-\vec{d}$.

When the graphs are generated, the problem of computing object-based saliency can be solved by matching the POs graph with the OIs.

C. Graph Matching

Graphs of objects of interest (GOI) are first learned using manually chosen object samples composed of laser points. The samples are processed in the same way with automatic procedures to generate GOIs. For each class of objects of interest, a pair of graphs is generated, one's edge vectors are re-estimated in original object coordinate while the other's in the mirror coordinate. The laser data as well as graphs of some objects of interest are shown in Fig. 9. Here we only give one graph for each object because the pair of graphs of the same object only vary in edges' attributes.

We suppose at this level that all potential objects (POs) in the range image have been abstracted as graphs (GPOs). For each GPO, we try to match it with all GOIs. Automatically extracted GFs and generated graphs are far beyond accurate

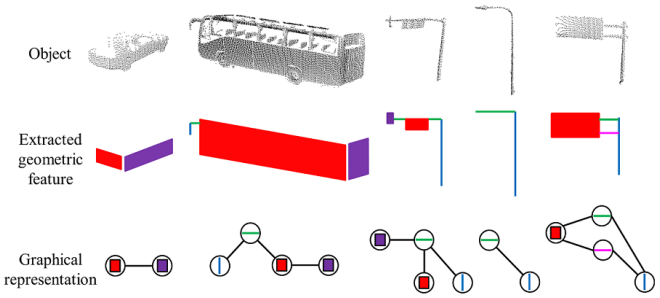


Fig. 9. Example of some samples.

than those in model training. The automatic processes are vulnerable to noise and neighboring objects. Hence the data graphs are often different from the model ones. The main challenge of graph matching is error-tolerance, which means the matching method should work on flawed GPOs as well as perfect GOIs. Therefore we use inexact graph matching [27]. Two graph nodes (or two edges) correspond if their attributes are similar. For example, given a GOI $G_m = (N_m, E_m)$ and a GPO $G_d = (N_d, E_d)$, where N stands for nodes and E stands for edges. First we run inexact matching on the two graphs. Then two best-matched subgraphs $G_{ms} = (N_{ms}, E_{ms})$ and $G_{ds} = (N_{ds}, E_{ds})$ are generated. After that, the matching degree $D(G_m, G_d)$ of G_m and G_d is computed,

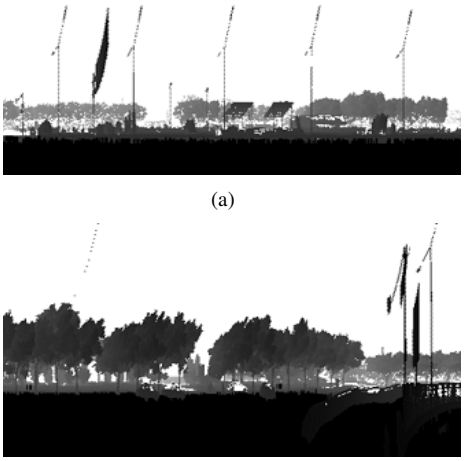
$$D(G_m, G_d) = \max \left(\sum_{k=0}^{card(N_{ms})} S_{N_{ms}^k}, \sum_{k=0}^{card(N_{ds})} S_{N_{ds}^k} \right) / \max \left(\sum_{k=0}^{card(N_m)} S_{N_m^k}, \sum_{k=0}^{card(N_d)} S_{N_d^k} \right) \quad (2)$$

where N^k denotes for the kth node in node set N, and S_n is the area of node n's corresponding GF. If the matching degree $D(G_m, G_d)$ is larger than a threshold (e.g. 0.8), the corresponding object of GPO G_d is considered as an object of interest. Pixels of GFs that belong to the object will be highlighted in the object saliency map M_c, where c is the object of interest consistent with GOI G_m.

V. EXPERIMENTAL RESULTS

To prove the effectiveness of our approach, we performed an experiment that took place in an urban environment. The sensing platform (POSS-v) drives approximately 13km long on the 4th ring road of Beijing. Range images collected by laser scanners L4-L5 are used as input of our approach, which contain 14.3 million laser points after ground removal. Two range image examples of the urban scene can be found in Fig. 10. Here we define 8 classes of objects of interest: car, bus, traffic light, road lamp, signpost, traffic sign, construction and road belt, which are frequently focused on in urban sensing. We manually extracted laser data of the "interesting" objects. With these data, a pair of graphical representations (GOI) is built for each class.

Using the method developed in this paper, objects of interest that appear in range image are located and highlighted. The results of input range image Fig. 10(a) are presented in Fig. 11, while those of input range image Fig. 10(b) are



(a)

(b)

Fig. 10. Two range images of the testing scene

presented in Fig. 12. Each resulting figure contains a series of object saliency maps in the left column, where certain classes of "interesting" objects are highlighted. To emphasize the effect of saliency, we can find a detail view of salient objects for each object saliency map in the middle column. 3D views of salient objects are illustrated in the right column, in which we can distinguish the objects of interest easily. As the highlighted regions and the objects of interest match well, the effectiveness of our approach is proved. On the other hand, views of laser points colored on "interesting" object classes are demonstrated in Fig. 13(a) and Fig. 13(b).

We test our approach on the entire data set collected from the 13-km course. The time cost of entire process on a PC equipped with 2.80-Ghz Core i5 and 8 GB of RAM is 18 minutes, which is half of the collecting time. Because most of our algorithms are parallelizable, the speed could be improved after optimization. The statistical results of object-based saliency are presented in Table II. For each class, we present the following: the total number of objects in the data, the total number of objects highlighted as belonging to this class, the number of objects that are highlighted as the class correctly, precision and recall rates of the object-based saliency. Note that the classes in the table do not include the special "other objects" category: of the 466 objects appear in the data, 18 are classified as other objects and would not be highlighted in any saliency maps.

From these results, our approach exhibits good performance on the 8 defined classes. However, there are still some incorrectly highlighted regions. There are mainly two reasons: 1) The effect of geometric feature extraction is limited by input data quality. 2) The implementation of object representation is not robust enough. Therefore, our approach can be improved in future studies.

VI. CONCLUSIONS AND FUTURE WORKS

We present a method for efficiently locating the objects of interest from the laser sensing data of urban scenes by computing object-based saliency. The problem of computing object-based saliency is formulated as searching for specific composition of geometries from the laser data. The input of our approach is a sequence of 3D laser scans of urban scenes, which can be represented in the formats of both range image and 3D point clouds. Through geometry extraction and object-based saliency computing, we generate a sequence of saliency maps, where the objects of interest are highlighted. Experimental results using the data of a large dynamic urban outdoor environment are presented, and performance of the algorithm is evaluated. Because of the low computational cost, our approach has potential in serving as a pre-filter in the on-line sensing tasks such as object detection, mapping and scene semantics extraction. All the laser scan data, training samples and processing results in this research will be opened at **http://poss.pku.edu.cn**.

Future studies will be addressed in extending object classes in saliency computing, and improving robustness in situations where noise, missing data and occlusion appear frequently.

TABLE II

class	Total	Highlighted	Correctly Highlighted	Precision	Recall
car	61	66	56	0.848	0.918
bus	27	22	20	0.909	0.741
traffic light	7	7	6	0.857	0.857
road lamp	210	196	190	0.969	0.904
signpost	13	18	11	0.611	0.846
traffic sign	62	71	56	0.789	0.903
construction	53	43	40	0.930	0.754
road belt	33	33	25	0.758	0.758
all	466	456	404	0.886	0.867

REFERENCES

[1] D. Hahnel, W. Burgard, and S. Thrun, "Learning compact 3D models of indoor and outdoor environments with a mobile robot," *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 15–27, 2003.

[2] P. Althaus and H. Christensen, "Behavior coordination in structured environments," *Advanced Robotics*, vol. 17, no. 7, pp. 657–674, 2003.

[3] A. Golovinskiy, V. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," *Computer Vision, IEEE Int. Conf.*, pp. 2154–2161, 2009.

[4] S. Frintrop, E. Rome, and H. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," *Applied Perception, ACM Trans.*, vol. 7, no. 1, p. 6, 2010.

[5] A. Nuthmann and J. Henderson, "Object-based attentional selection in scene viewing," *Journal of vision*, vol. 10, no. 8, 2010.

[6] J. Wolfe and T. Horowitz, "What attributes guide the deployment of visual attention and how do they do it?," *Nature Reviews Neuroscience*, vol. 5, no. 6, pp. 495–501, 2004.

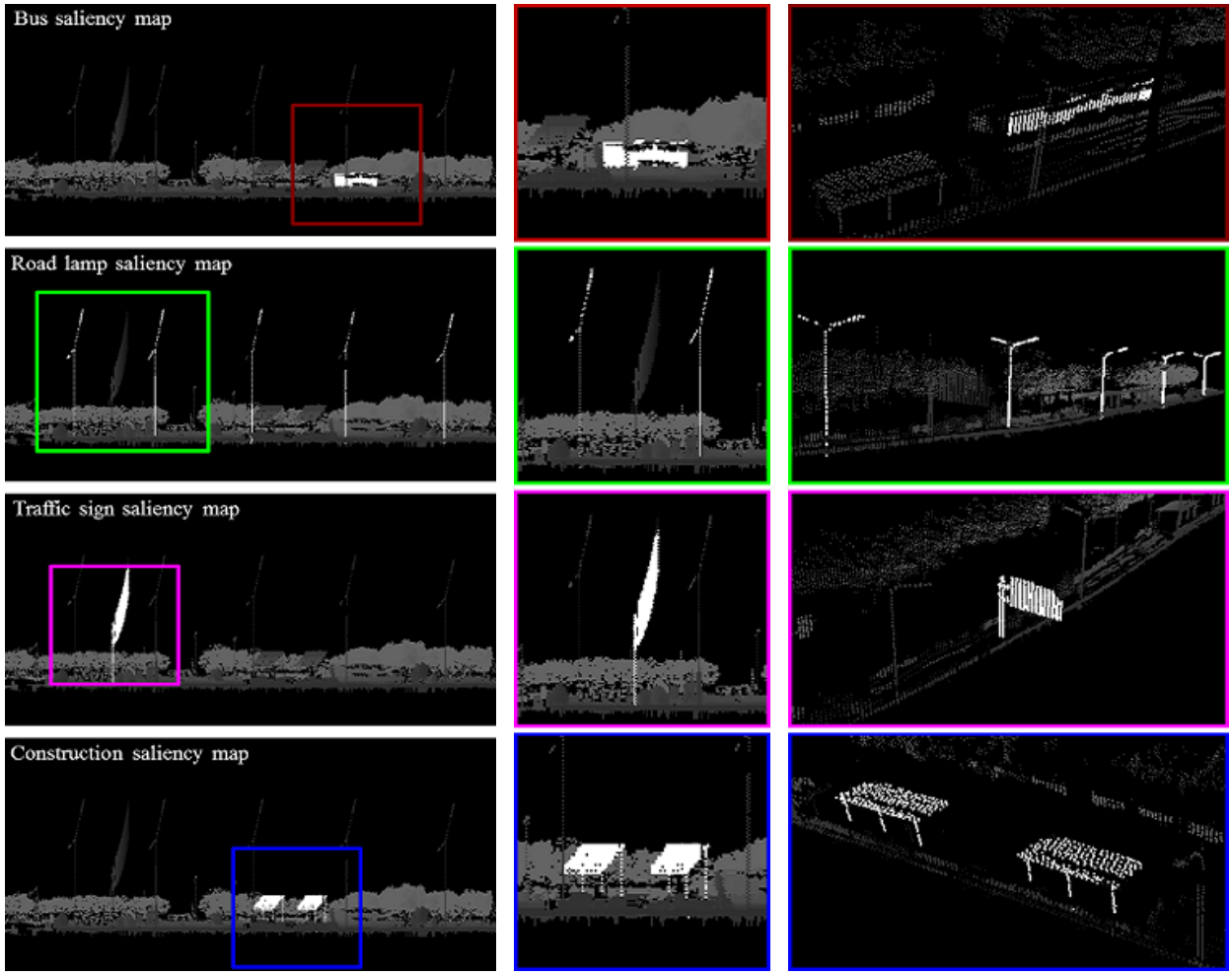
[7] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature reviews neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[8] S. Choi, S. Ban, and M. Lee, "Biologically motivated visual attention system using bottom-up saliency map and top-down inhibition," *Neural Information Processing-Letters and Review*, vol. 2, no. 1, 2004.

[9] N. Murray, M. Vanrell, X. Otazu, and C. Parraga, "Saliency estimation using a non-parametric low-level vision model," *Proc. Computer Vision and Pattern Recognition*, 2011.

[10] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (pcl)," *Robotics and Automation, IEEE Int. Conf.*, 2011.

[11] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," *Computer Vision Workshops, IEEE Int. Conf.*, pp. 39–46, 2009.



(a) Object saliency maps (b) Detail views of salient objects (c) 3D views of salient objects

Fig. 11. Object saliency results of range image Fig. 10(a), demonstrated as multiple maps.

[12] R. Triebel, K. Kersting, and W. Burgard, "Robust 3D scan point classification using associative markov networks," *Robotics and Automation, IEEE Int. Conf.*, pp. 2603–2608, 2006.

[13] D. Anguelov, B. Taskarf, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of markov random fields for segmentation of 3D scan data," *Computer Vision and Pattern Recognition, IEEE Conf.*, vol. 2, pp. 169–176, 2005.

[14] E. Lim and D. Suter, "Conditional random field for 3D point clouds with adaptive data reduction," *Cyberworlds, Int. Conf.*, pp. 404–408, 2007.

[15] A. Johnson, *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[16] C. Tang and G. Medioni, "Curvature-augmented tensor voting for shape inference from noisy 3D data," *Pattern Analysis and Machine Intelligence, IEEE Trans.*, vol. 24, no. 6, pp. 858–864, 2002.

[17] R. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3D registration," *Robotics and Automation, IEEE Int. Conf.*, pp. 3212–3217, 2009.

[18] H. Zhao, L. Xiong, Z. Jiao, J. Cui, H. Zha, and R. Shibasaki, "Sensor alignment towards an omni-directional measurement using an intelligent vehicle," *Intelligent Vehicles Symposium, IEEE Int. Conf.*, pp. 292–298, 2009.

[19] R. Fraile and S. Maybank, "Building 3d models of vehicles for computer vision," *Visual Information and Information Systems*, pp. 699–704, 1999.

[20] K. Khoshelham, "Extending generalized hough transform to detect 3D objects in laser range data," *Transform*, vol. 36, pp. 206–210, 2007.

[21] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[22] A. Leonardis, A. Gupta, and R. Bajcsy, "Segmentation of range images as the search for geometric parametric models," *Computer Vision, Int. Journal*, vol. 14, no. 3, pp. 253–277, 1995.

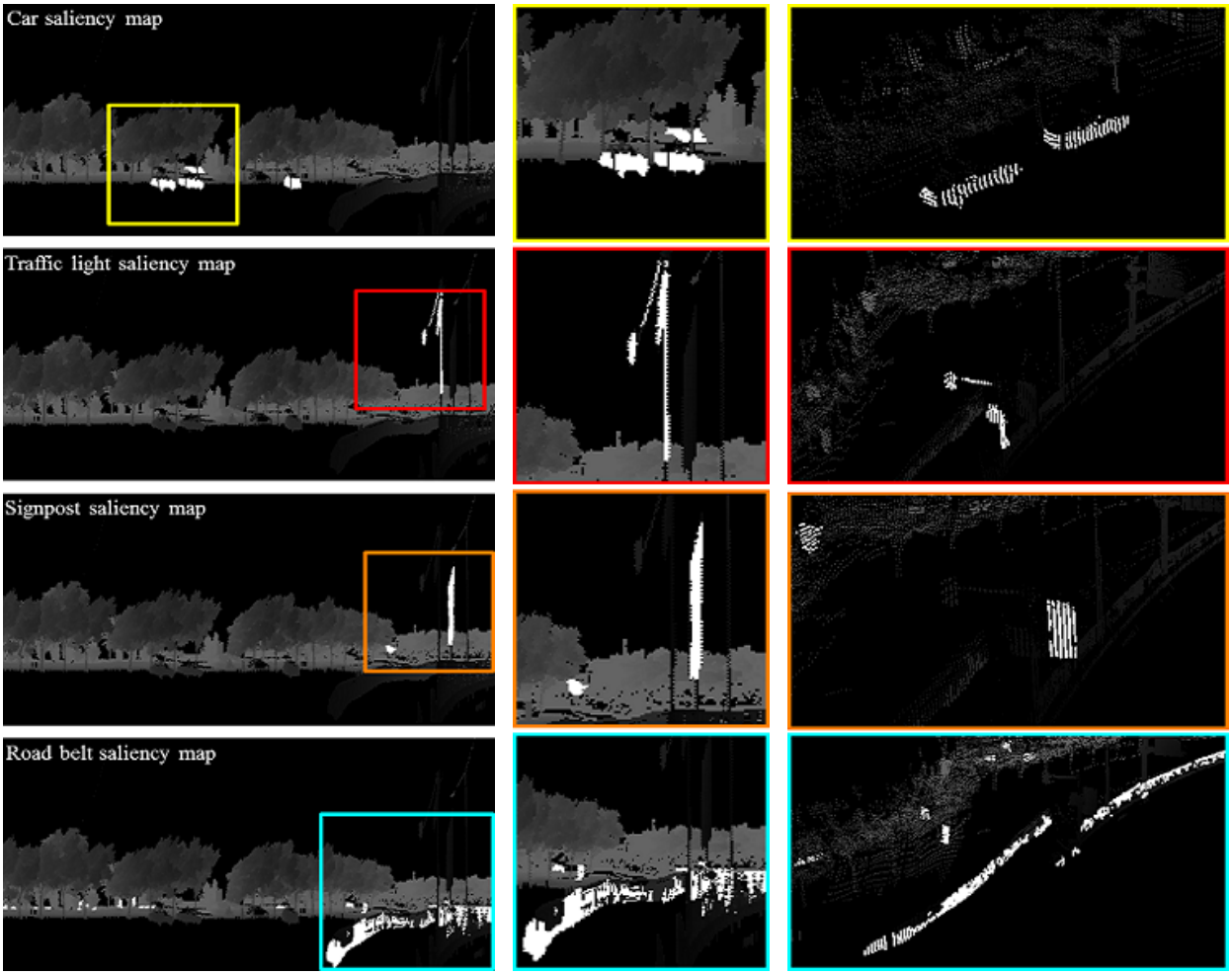
[23] J. Daniels, L. Ha, T. Ochotta, and C. Silva, "Robust smooth feature extraction from point clouds," *Shape Modeling and Applications, IEEE Int. Conf.*, pp. 123–136, 2007.

[24] S. Gächter, A. Harati, and R. Siegwart, "Structure verification toward object classification using a range camera," *Intelligent autonomous systems 10: IAS-10*, p. 356, 2008.

[25] R. Schnabel, R. Wessel, R. Wahl, and R. Klein, "Shape recognition in 3D point-clouds," in *Proc. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, vol. 2, Citeseer, 2008.

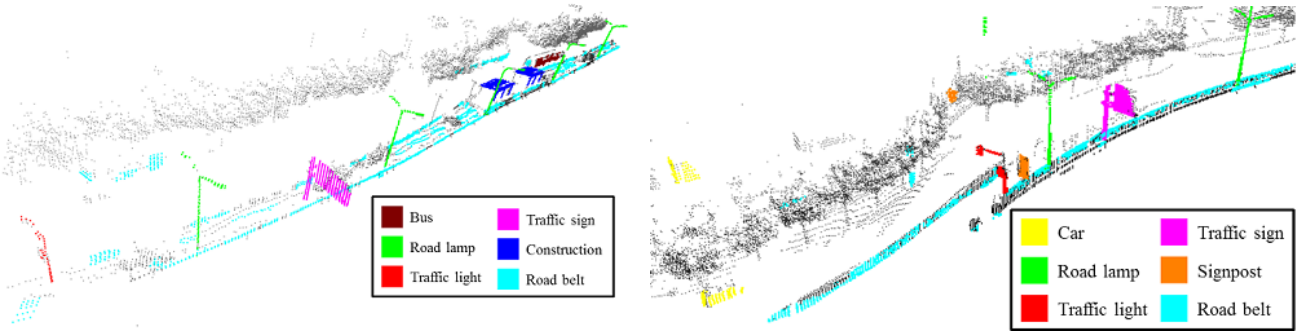
[26] S. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[27] R. Cesar, E. Bengoetxea, I. Bloch, P. Larrañaga, *et al.*, "Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms," *Pattern Recognition*, vol. 38, no. 11, pp. 2099–2113, 2005.



(a) Object saliency maps (b) Detail views of salient objects (c) 3D views of salient objects

Fig. 12. Object saliency results of range image Fig. 10(b), demonstrated as multiple maps.



(a) The annotated laser points of Fig. 10(a) on object saliency results (b) The annotated laser points of Fig. 10(b) on object saliency results

Fig. 13. Object saliency results rendered in 3D view.