

Fusion of Velodyne and camera data for scene parsing

Zhao, Gangqiang; Xiao, Xuhong; Yuan, Junsong

2012

Zhao, G., Xiao, X., & Yuan, J. (2012). Fusion of Velodyne and Camera Data for Scene Parsing. 15th International Conference on Information Fusion (FUSION), pp.1172-1179.

<https://hdl.handle.net/10356/100802>

© 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6289941>].

Downloaded on 20 Mar 2024 16:21:03 SGT

Fusion of Velodyne and Camera Data for Scene Parsing

Gangqiang Zhao¹, Xuhong Xiao², Junsong Yuan¹

¹School of EEE, Nanyang Technological University, Singapore

²DSO National Laboratories, Singapore

Abstract—The fusion of information gathered from multiple sources is essential to build a comprehensive situation picture for autonomous ground vehicles. In this paper, an approach which performs scene classification and data fusion for 3D-LIDAR scanner (Velodyne HDL-64E) and video camera is described. A geometry segmentation algorithm is proposed for detection of obstacles and ground area from data collected by the Velodyne. In the meantime, the corresponding image collected by video camera is classified patch by patch into more detailed categories. The final situation picture is obtained by fusing the classification results of Velodyne data and that of images using the fuzzy logic inference framework. The proposed approach was evaluated with datasets collected by our autonomous ground vehicle testbed in the rural area. The fused results are more reliable and more completable than those provided by individual sensors.

I. INTRODUCTION

Autonomous situation awareness is an important research aspect for robots and unmanned vehicles. Besides whether the terrain is traversable, they also require more specific object category information to carry out their tasks: e.g., approaching a tree, or the water area. For decades, computer vision approaches have been studied to classify scenes from images. In recent years, LIDAR scanners have also been applied to complement the video cameras.

In this work, two sets of sensors are used: the Velodyne HDL-64E 3D-LIDAR scanner [1] and the monocular video camera. The Velodyne scanner provides the 3-dimensional but sparse pointcloud of the surrounding environment. The pointcloud is trustworthy for obstacle detection but lacks information to perform a reliable classification of specific object categories. Besides, the covering range of Velodyne is very limited, therefore, only information of close-by area can be acquired. The images captured by video cameras can cover a much broader area and provide more discriminative information to classify objects into categories. However, due to the lack of depth information, image-based detection of obstacles of various shapes, sizes and orientations remains challenging. Due to the complementary benefits of two sensors, it is possible to acquire more reliable scene classification by fusing the information derived from these two sensors.

Approaches for fusion of LIDAR data and images can be divided into two categories: centralized and decentralized. In centralized approaches, the fusion process occurs at the pixel-level [2] or feature level [3] [4] [5], i.e., features from both LIDAR and video camera are combined in a single vector for posterior classification. Centralized methods can simplify

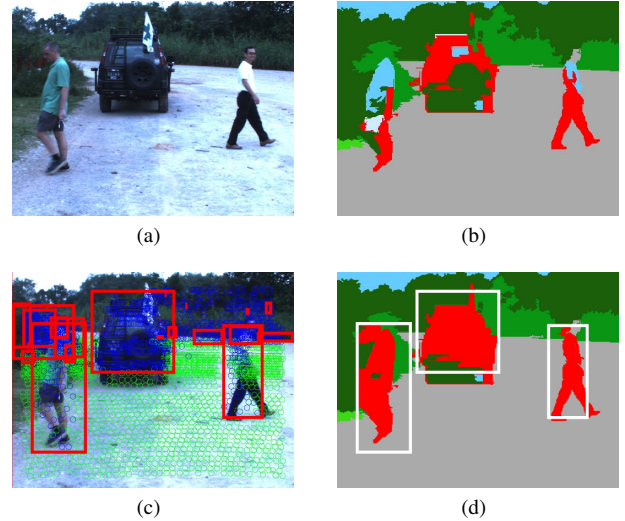


Fig. 1. Illustration of two sensors fusion. (a) shows the original camera image; (b) shows the camera image classification result; (c) shows the detection result of Velodyne scanner which is projected to the image; (d) shows the result of our fusion method. In (d), the red regions are the obstacle areas and each white bounding box localizes one detected obstacle. By fusing results of two sensors, it not only improves the scene classification result but also localizes the obstacles correctly.

the posterior classify process but are difficult to integrate the human knowledge and experience. Furthermore, in the centralized method, only the regions commonly observed by both sensors can be processed. This greatly limits the area they can cover due to the short range of LIDAR sensor. Decentralized approaches separately classify the data for individual sensor first, the classification results are then combined by a fusion method [6] [7]. Generally, these methods require training data to determine the fusion model and the fusion parameters. Besides the two fusion strategies, there are works which try to use them together [8] [9] [10] [11].

In this research, we propose a new way to fuse the results of two sensors by employing the fuzzy logic inference [12]. Fig. 1 illustrates the fusion process. The fuzzy logic is preferable for our application due to their advantages. First, the fuzzy logic is built on top of the knowledge and experience of experts. Therefore, it can employ not only results from LIDAR and video camera data but also the *a priori* knowledge. Second, the fuzzy logic can model nonlinear functions of arbitrary

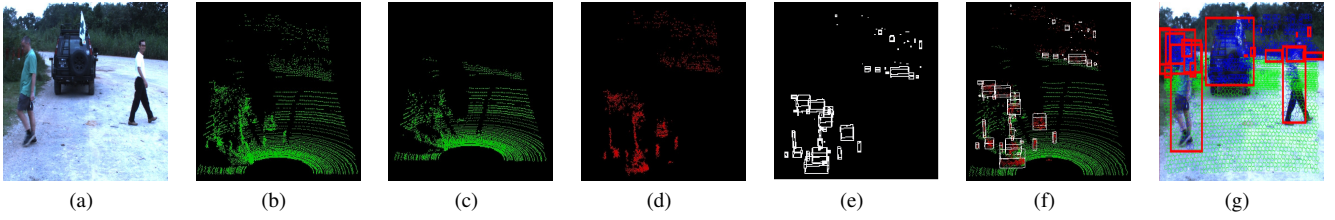


Fig. 2. Illustration of obstacle and ground classification using Velodyne scanner. (a) is the camera image; (b) is the 3D pointcloud of Velodyne scanner; (c) is the ground points; (d) is the above-ground points; (e) shows the detected bounding boxes of the candidate obstacles; (f) shows the pointcloud and the detected results; (g) shows the detected results which are projected to the camera image. Each bounding box represents one candidate obstacle in (e), (f) and (g).

complexity. This is important as scene classification is not a trivial problem. Third, the fuzzy logic can tolerate imprecise results of two sensors. Moreover, the fuzzy logic is a flexible fusion framework and the results of more sensors can be easily integrated to the system. To the best of our knowledge, the proposed approach is the first systematic fuzzy logic inference based fusion work for scene understanding by fusing the results of the Velodyne 3D-LIDAR scanner and the monocular video camera. We test the proposed approach on datasets collected by our autonomous ground vehicle testbed and the results validate the robustness and effectiveness of our method.

II. CLASSIFICATION MODULE FOR INDIVIDUAL SENSORS

As a decentralized fusion method, a geometry segmentation algorithm is proposed to detect obstacle and ground from Velodyne data for this work. In the meantime, one algorithm, which combines both bottom-up and top-down analyses, is designed to classify image patches into multiple categories. In this section, we first describe the two detection algorithms separately and then summarize their pros and cons.

A. Obstacles and ground classification using Velodyne scanner

As mentioned earlier, due to the sparseness of pointcloud, we detect only traversability of the terrain (i.e., classifying the pointcloud into ground and candidate obstacles) from the Velodyne data. To achieve it, we first voxelize the pointcloud \mathcal{P} . Then we separate the ground points using a RANSAC plane fitting algorithm [13]. After that, all the above-ground points are obtained and the candidate obstacles are localized by partitioning the above-ground points using 3D adjacency. Fig. 2 illustrates the result of each step. To speed up the process, we first build a 3D cubic voxel grid using the pointcloud \mathcal{P} . The pointcloud data are stored in cubic voxels for efficient retrieval and the grid resolution is denoted as res and set to be 0.1 meter. By voxelizing, the spatial neighborhood relationships of the 3D points are modelled explicitly.

The second step separates the points into two categories: ground and non-ground. Points are considered in batches, defined by their membership in a single cubic voxel in space. A voxel is considered to contain ground data if the voxel is a member of the lowest (in elevation) set of adjacent non-empty voxels in a vertical column (i.e. not part of an overhang). All 3D points stored in that set of voxels are fitted to a plane using

the RANSAC algorithm and the inliers points are the ground points. All of the voxels that contain ground points are called ground voxel set \mathcal{G} . Other voxels are called the above-ground voxel set \mathcal{U} . One above-ground voxel $V_{i,j,k} \in \mathcal{U}$ may contain a number of above-ground points or be an empty voxel, where i, j and k denotes the indexes of the 3D voxel grid.

The third step detects the possible obstacles by clustering the above-ground voxels according to 3D adjacency [14]. Each obstacle is represented by a voxel cluster. Denote all the voxel clusters as \mathcal{C} and the voxels in one voxel cluster $O \in \mathcal{C}$ should meet the following 3D adjacency criterion:

$$\begin{aligned} & \forall V_{i,j,k} \in O \\ & \exists V_{i',j',k'} \in O \\ & \wedge (|i - i'| < 2 \vee |j - j'| < 2 \vee |k - k'| < 2) \end{aligned}, \quad (1)$$

A summary of our detection algorithm is given in Alg. 1. The detected results are projected to the image as shown in Fig.2(g). Each bounding box localizes one candidate obstacle. The green circles represent the projection of ground points and the blue circles represent the projection of above-ground points.

Algorithm 1 The algorithm for ground and obstacle detection using Velodyne scanner

input : point cloud \mathcal{P} .

output : ground voxel set \mathcal{G} , above-ground voxel set \mathcal{U} , voxel clusters \mathcal{C}

- 1 voxelgrid \leftarrow FillVoxelGrid(\mathcal{P} , res) ;
 - 2 voxel set \mathcal{G} and voxel set $\mathcal{U} \leftarrow$ ExtractGround(voxelgrid);
 - 3 voxel clusters $\mathcal{C} \leftarrow$ ClusterVoxelSet(voxel set \mathcal{U}) ;
-

B. Image classification module

Contrary to Velodyne information processing which concerns whether the terrain is traversable, we intend to identify more specific categories of objects from the images. From the rural image that we have collected, we have identified seven possible categories, which include ground, water, high vegetation (highVeg), grass and obstacles, etc. As a matter of fact, obstacles can be further divided into human, car and building etc. But this detailed division requires sufficient training data for each specific class. Viewing that there are many types of possible known or unknown obstacles, we simply classify all of them into obstacle. For a particular

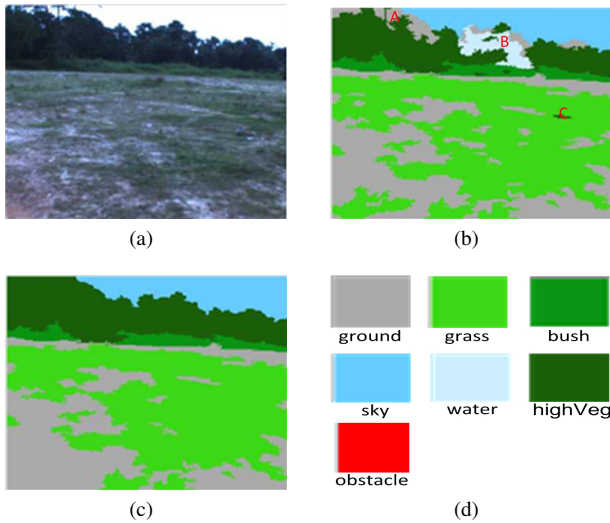


Fig. 3. Illustration of image classification. (a) is the original image; (b) is the classification result of the bottom-up phase; (c) is the final classification result after top-down contextual analysis; (d) shows the color of each category.

task, specific models can be trained for individual interesting classes too. The classification of images is realized by two steps: bottom-up classification of local image patches and top-down contextual analysis to further resolve uncertainties in the bottom-up classification.

During bottom-up classification phase, an image is first over-segmented into small image patches [15]. From each patch, 131 features are extracted, including 24 features from color histograms and 107 features corresponding to different texture descriptors. 36 of them are derived from anisotropic Gauss filtered images, 12 from Gabor filtered images, and 59 Local Binary Patterns [16]. An MLP (multilayer Perception) classifier is trained to classify the local image patches into object categories [17]. Fig. 3(b) is an example of bottom-up classification result, where patches of original image in Fig. 3(a) are classified into different categories.

Sometimes, errors will occur in the bottom-up classification. For instance, in Fig. 3(b), some image patches of “sky” (area A) are wrongly classified into “ground”, some part of “highVeg” (area B) is classified into “water”, and a part of “grass” (area C) is classified as “highVeg”. Some errors in bottom-up classification can be further corrected by a top-down contextual analysis process. This is because only local features of the image patches are considered during the bottom-up classification phase. It is possible that local patches of different object categories may look similar, leading to uncertainties in the bottom-up classification. However, when looking at an image patch from its surrounding context, e.g., the categories of its neighbors, the uncertainty can be resolved. For example, “ground” cannot be above “highVeg” in the image if it is taken from a moving vehicle. This property has been well recognized and employed in several computer vision systems [18]. However, most of them either treat

contextual information equally with local, low-level features or mix the contextual information with low-level features in one classifier. Our work is different from them in that we model the contextual relations independent of the bottom-up classification process, allowing the contextual analysis result to feedback to the bottom-up classification module so as to update the final classification result.

To acquire the top-down contextual relation module, the connected image patches classified into the sample category by the bottom-up classification process are first grouped into bigger components, each component corresponds to a connected area. Then, the existence of neighboring categories of a component is derived from three directions: top, down and sides (both left and right sides). They form the contextual information of the component. From the training sets, a Bayesian network is learned to represent the relations between the category of the component and its neighboring categories.

The top-down process works as follows. Contextual information of an image component is first acquired based on bottom-up classification result as described above. This contextual information is then passed to the Bayesian network as evidence. The probability of the node “category” will be updated through Bayesian network inference. This updated probability is fused with the bottom-up classification confidence via multiplication. As shown in Fig. 3(c), the classification errors in area A, B and C are corrected after contextual analysis.

C. Summarization of two methods

By analysing the results, it can be seen that both methods have their own advantages and disadvantages. The laser scanner based method can separate the ground and above-ground points robustly and accurately. It can also segment the obstacles if they are not adjacent to other obstacles. However, the laser scanner can only obtain a sparse pointcloud and it has no information about the water, sky and the areas out of range of the sensor. Besides, the detected obstacles include many tree and bush areas, which will increase the possibility of the vehicle deviating from the road region. As for the camera based method, it can classify the image into seven categories and detect the trees and bushes with high accuracy. However, due to the diversity of the obstacles, some obstacle regions may be classified as wrong categories. The complementary performance of two methods shows the possibility to boost the scene classification and obstacle detection by combining them.

III. FUZZY LOGIC BASED FUSION

Both the results of laser scanner and the results of camera image have their own pros and cons. To parse the scene correctly, the primary work of fusion is to categorize the detected candidate obstacles by Velodyne scanner. The scene classification results are then improved based on the categorization. As a good way to utilize the *a priori* knowledge and experience of human experts [12], we propose to use the fuzzy inference method to fuse the results of two sensors.

A. fuzzification of the fusion

The inputs to the fuzzy fusion module are five related attributes of each candidate obstacle: the size of candidate obstacle (*size*), the image classification result (*class*), the spatial context (*s-context*), the temporal context (*t-context*) and the absolute height (*height*) of the candidate obstacles. The output result classification (r_c) is the detection result for the candidate obstacles. Each input and out parameter is defined as a fuzzy variable.

To employ the *a priori* knowledge, all the associated fuzzy variables are first fuzzified into linguistic labels. The input variable *size* is simply expressed using three linguistic labels SMA (small), MID (middle) and LAR (large) within the universe of discourse (0, 100) percents. The candidate obstacle size is defined as the percent of all image pixels which are inside the candidate obstacle bounding box. The variable *class* is expressed using three linguistic labels GRE (greenery), MID (middle) and OBS (obstacle) within the universe of discourse (0, 100) percents. The classification is measured by the percent of greenery pixels among all the pixels inside the candidate obstacle bounding box. All the detected grass, bush and highVeg pixels by image classification method are considered as greenery pixels. When most of inside pixels belong to the greenery category, the candidate obstacle is probably the tree or the bush, and vice versa.

The spatial context *s-context* is expressed using two linguistic labels GRE (greenery) and OBS (obstacle) within the range(0, 8). It is obtained from the classification results of eight pixels around the candidate obstacle bounding box. Four of them are the corners of the box and the other four are the middle point of each edge of the bounding box. If one pixel is classified as ground, *s-context* is added by one. The temporal context is expressed using two linguistic labels LOW (low) and HIG (high) within the range (0, 1). The temporal context describes the existence possibility of current obstacle in the previous frame. By checking the neighborhood of current position in the previous frame, if there is one obstacle with similar size and classification as current one, the temporal context is HIG. Otherwise, the temporal context is LOW. The *height* is the absolute height of the candidate obstacle obtained by the scanner directly. It is expressed using three linguistic labels LOW (low), MID (middle) and HIG (high) within the range (0, 10) meters. If the obstacle is very high (i.e. > 4m), it is more likely to be a tree rather than a car.

The output result score (r_c) is simply expressed using three linguistic labels GRE (greenery), MID (middle) and OBS (obstacle) within the universe of discourse (0, 1). All the membership functions of input and output variables are illustrated in Fig. 4.

B. knowledge rules of scene classification

Based on the human knowledge and experience, a vehicle is required to move on the ground and avoid all the obstacles simultaneously. To detect the categorization of each candidate obstacle, both the detection results of scanner and the camera are used. Besides, the spatial and temporal context of the

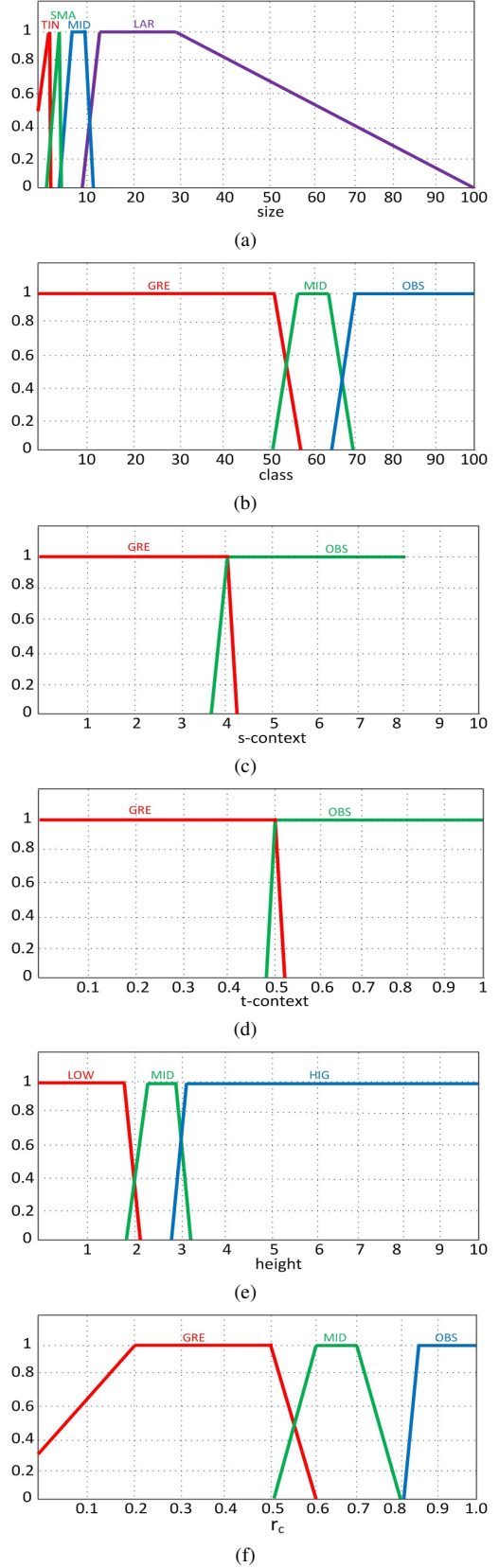


Fig. 4. Illustration of membership function for input and output fuzzy variables. (a) shows the membership function of *size*; (b) shows the membership function of *class*; (c) shows the membership function of *s-context*; (d) shows the membership function of *t-context*; (e) shows the membership function of *height*; (f) shows the membership function of r_c .

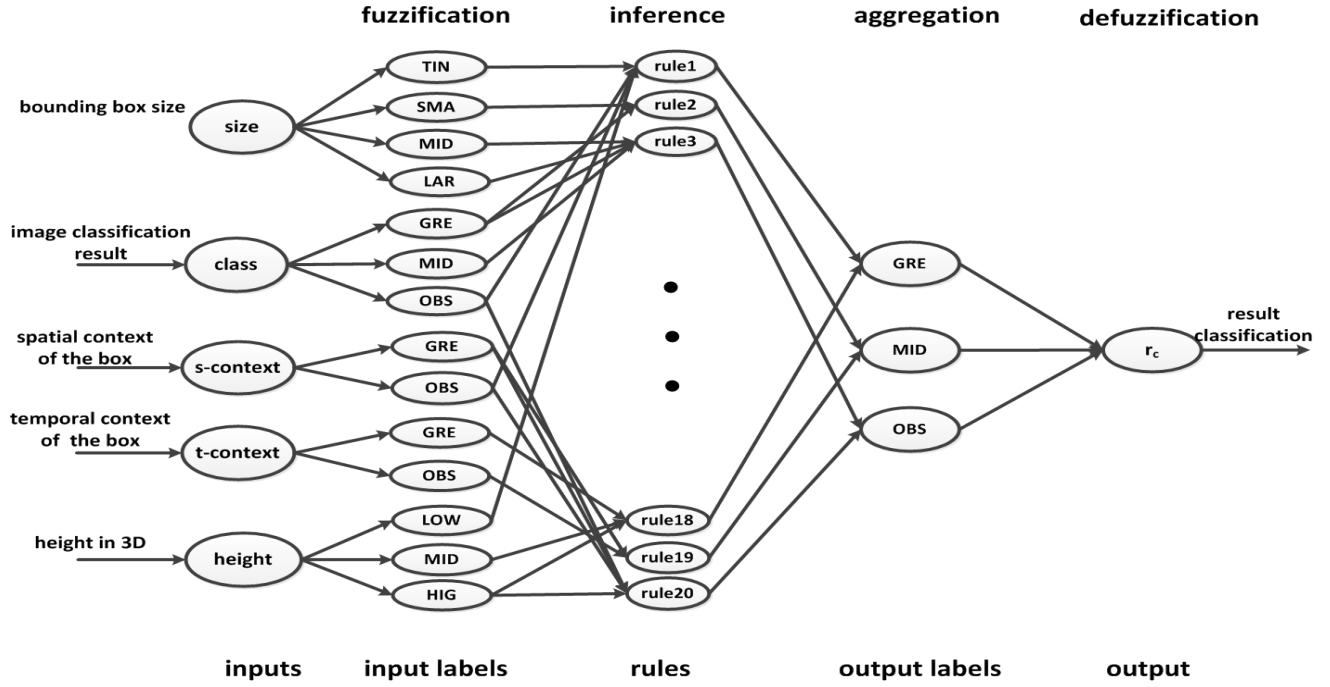


Fig. 5. Information flow of fuzzy reasoning for scene classification.

obstacle is also important knowledge. When the candidate obstacle is surrounded by the ground region, it is probably an obstacle. However, when the candidate obstacle is on the edge of ground region, its categorization highly depends on the image classification result and other information like the height of the obstacle. By analyzing the application scenario of our auto-driving vehicle, the following rules are selected.

The group of rules when the size of object box is large:

- R_1 : if *size* is LAR and *class* is OBS then r_c is OBS;
- R_2 : if *size* is LAR and *class* is MID then r_c is MID;
- R_3 : if *size* is LAR and *class* is GRE then r_c is GRE;
- R_4 : if *size* is LAR and *class* is GRE and *s-context* is GRE then r_c is GRE;
- R_5 : if *size* is LAR and *class* is GRE and *t-context* is GRE then r_c is GRE;
- R_6 : if *size* is LAR and *class* is GRE and *s-context* is OBS then r_c is OBS;

The italic assertion in R_1 to R_6 is the condition part of each rule, which is contributed by the detection result of two sensors. These rules indicate that the size of the obstacle is not the only criterion to decide categorization of the obstacle boxes. The image classification result and the context information are also very important for scene classification.

When the size of obstacle is becoming smaller and smaller, the image classification result and the context information will play a more important role for scene classification:

- R_7 : if *size* is MID and *class* is OBS then r_c is OBS;
- R_8 : if *size* is MID and *class* is MID then r_c is MID;
- R_9 : if *size* is MID and *class* is GRE then r_c is MID;
- R_{10} : if *size* is MID and *class* is MID and *s-context* is GRE then r_c is GRE;
- R_{11} : if *size* is MID and *class* is GRE and *s-context* is GRE then r_c is GRE;
- R_{12} : if *size* is MID and *class* is GRE and *t-context* is GRE then r_c is GRE;
- R_{13} : if *size* is MID and *class* is GRE and *height* is MID then r_c is GRE;
- R_{14} : if *size* is SMA and *class* is OBS then r_c is MID;
- R_{15} : if *size* is SMA and *class* is OBS and *s-context* is OBS then r_c is OBS;
- R_{16} : if *size* is SMA and *class* is OBS and *s-context* is GRE then r_c is GRE;
- R_{17} : if *size* is TIN then r_c is MID;

The absolute height of one candidate obstacle is also an important criterion to decide the result. If the obstacle's height is very large (e.g., higher than 4 meters), the obstacle is more likely a tree rather than a car. The height attribute is included in the following rules:

- R_{18} : if *class* is GRE and *height* is MID then r_c is GRE;
- R_{19} : if *class* is MID and *height* is MID then r_c is GRE;
- R_{20} : if *height* is HIG then r_c is GRE;

Although 20 rules do not cover the complete relationships of different attributes, these rules help to integrate the results of two sensors and the human knowledge and experience.

C. fuzzy reasoning

After synthesizing these 20 rules for fusion, their roles are further coordinated through Mamdani's fuzzy reasoning method in this section [19]. The process of Mamdani fuzzy inference involves steps fuzzification, inference, aggregation and defuzzification. The information flow of the fuzzy reasoning is shown in Fig. 5.

Fuzzification converts the input values into a degree via membership functions. The input is always a crisp numerical value and the output is a fuzzy degree of membership in the qualifying linguistic set. The membership functions are illustrated in Fig. 4. After the inputs are fuzzified, the inference of a rule uses the minimal operation to combine different condition assertions for logical operator *and* and generate the output grade for the conclusion assertion. Taking rule R_7 as an example, given a set of inputs *size*^{*} and *class*^{*}, the output grade r_s^* of the label OBS due to this rule can be

inferred as:

$$U_{OBS}^7(r_s^*) = \min(U_{MID}(size^*), U_{OBS}(class^*)) \quad (2)$$

where $U_{MID}(size^*)$ and $U_{OBS}(class^*)$ represent the membership functions of the corresponding labels.

There are two steps involved in the aggregation process: the maximum operation of the output grades of each output label due to several rules, and the generation of the output membership function. The aggregated output grade belonging to one corresponding label (such as label OBS) is calculated as:

$$U_{OBS}(r_s^*) = \max(U_{OBS}^1(r_s^*), U_{OBS}^2(r_s^*), \dots, U_{OBS}^{20}(r_s^*)) \quad (3)$$

The aggregated output membership function $U_O(r_s)$ is obtained by cutting the membership function $U_{OBS}(r_s)$, $U_{MID}(r_s)$ and $U_{GRE}(r_s)$ respectively at the grades $U_{OBS}(r_s)^*$, $U_{MID}(r_s)^*$ and $U_{GRE}(r_s)^*$, and combining them point by point:

$$U_O(r_s) = \max(\min(U_{OBS}(r_s^*), U_{OBS}(r_s)), \min(U_{MID}(r_s^*), U_{MID}(r_s)), \min(U_{GRE}(r_s^*), U_{GRE}(r_s))) \quad (4)$$

After aggregation, the input for the defuzzification process is a fuzzy set and the output is a single number. The defuzzification process finds the center of gravity of the output membership function as the real value of the output variable:

$$r_s^* = \frac{\int U_O(r_s) r_s dr_s}{\int U_O(r_s) dr_s} \quad (5)$$

r_s^* is the final crisp classification score for the candidate box. Based on the classification score, the categories of the candidate obstacles are decided. If the result score of one candidate obstacle is large enough (i.e., $r_c > 0.65$), it is classified as the obstacle. Otherwise, its result depends on the image classification method. After that, we update the categories of the patches inside the obstacle bounding boxes by considering the results of two sensors.

IV. PERFORMANCE EVALUATION

To evaluate our approach, we test it on the datasets collected by an autonomous ground vehicle testbed when driving the vehicle in the rural area. In addition, we compare the result of our fusion approach and that of the image classification approach.

A. Dataset

Two datasets are collected by an autonomous ground vehicle testbed while the vehicle is outfitted with a Velodyne 3D-LIDAR scanner, a monocular camera and other sensors. The first dataset corresponds to an open ground in the rural area while the second one corresponds to the road in the rural area. As shown in Table I, the first dataset consists about 440 keyframes and the second one consists about 450 keyframes. The vehicle runs about 3 minutes to collect each dataset.

TABLE I
THE INFORMATION OF TWO DATASETS.

Dataset	Frame No.	Labelled Obstacle No.	Correctly detected No.
Dataset 1	440	170	170
Dataset 2	450	53	53

B. Scene classification and Obstacle detection

To quantify the performance of the proposed approach, we manually labelled the ground truth positions of the instances of obstacles in about 20% of all keyframes. Consider the scenario of autonomous ground vehicles, only the obstacles above the traversable ground area are labelled. Let DR and GT be the discovered obstacle regions and the bounding boxes of ground truth, respectively. The performance is measured by two criteria: $precision = \frac{|GT \cap DR|}{|DR|}$ and $recall = \frac{|GT \cap DR|}{|GT|}$. By combining $precision$ and $recall$, we use a single $F-measure$ as the metric for performance evaluation. $F-measure = \frac{2 \times recall \times precision}{recall + precision}$ is the weighted harmonic mean of $precision$ and $recall$. We evaluate the fusion method on these two datasets. Fig. 7 shows the result of several keyframes. The top panel shows the result of dataset 1 and the bottom panel shows the result of dataset 2. In each panel, the first row shows the original camera image and the second row shows the image classification result. The red color illustrates the region of detected obstacles and other colors have similar meaning as in Fig. 3(d). The third row shows the detected result using scanner pointcloud and the results are projected to the camera image. Each bounding box localizes one candidate obstacle. The green circles represent the projection of ground points and the blue circles represent the projection of above-ground points. There are ground points inside several bounding boxes due to the 3D-2D projection. The fourth row shows the result of our fusion method. Each white bounding box localizes one detected obstacle. In the sequences, the obstacles are subject to variations introduced by moving vehicles and pedestrians, static obstacles, road curvature changes, etc. It is possible that some keyframes contain only one obstacle and some keyframes do not contain any obstacles.

Table I shows the number of total labelled obstacles and the number of correctly detected obstacles for each dataset. It can be seen that the proposed method has a obstacle detection rate close to 100%. These results show that the proposed approach performs well for scene classification and obstacle detection from real-world driving environment.

To further evaluate the proposed method, we compare its result with that of image classification method. As shown in Fig. 6, our proposed fusion approach improves the obstacle classification significantly in terms of $F-measure$ value. This is because the detected results of our method include major parts of or the complete obstacle regions. On the contrary, the image classification method only detects small parts of the obstacle regions due to the diversity of the obstacles. Therefore it obtains a high $precision$ value but with a low $recall$ value. These comparisons clearly demonstrate the advantages of the proposed fusion method.

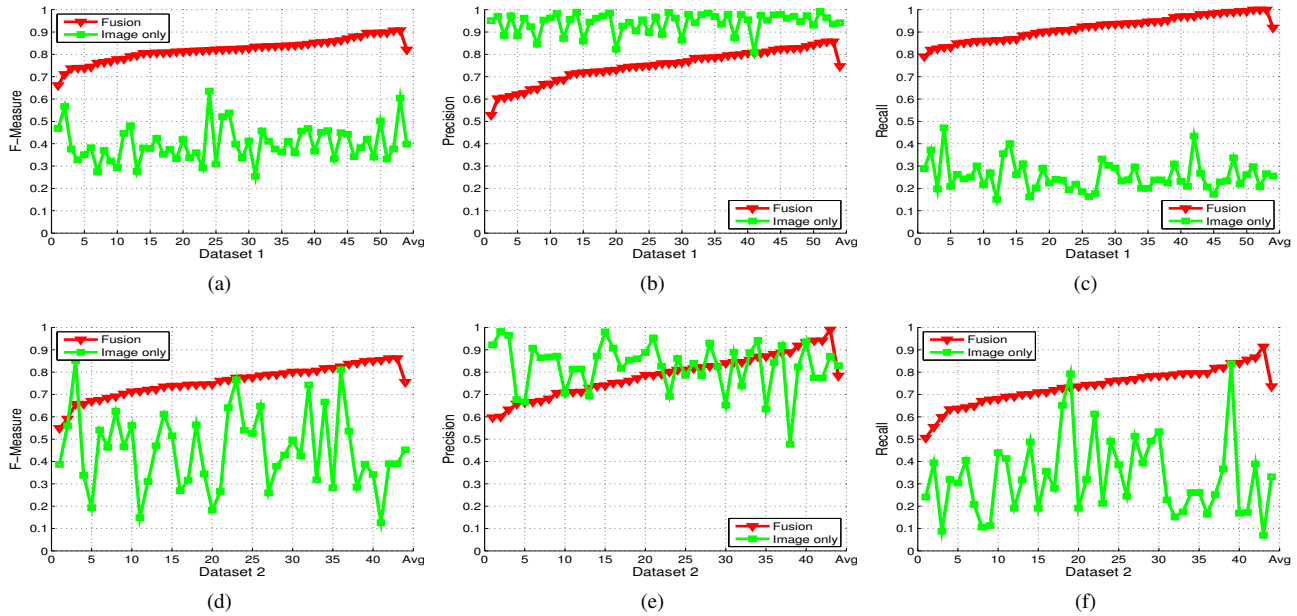


Fig. 6. The performance evaluation of the proposed fusion based scene classification method (Fusion) and the image classification method. (a) shows the F -measure value of dataset 1; (b) shows $precision$ value of dataset 1; (c) shows $recall$ value of dataset 1; (d) shows the F -measure value of dataset 2; (e) shows $precision$ value of dataset 2; (f) shows $recall$ value of dataset 2.

V. CONCLUSIONS

In this paper, we propose a fuzzy inference based fusion method for scene classification using laser scanner and video camera. It can incorporate not only the results of two sensors, but also the human experience and knowledge. The proposed approach was evaluated with datasets collected by an autonomous ground vehicle testbed in the rural area. The fused results are more reliable and more completable than those provided by individual sensors.

ACKNOWLEDGMENT

This work was supported in part by the DSO-NTU project M4060969.040, as well as Nanyang Assistant Professorship to Dr. Junsong Yuan. We thank Jingjing Meng to help proof read the paper.

REFERENCES

- [1] Velodyne Lidar Inc., “Hdl-64e,” in <http://velodynelidar.com/lidar/hdlproducts/hdl64e.aspx>.
- [2] Stephen R. Schnelle and Alex Lipchen Chan, “Enhanced target tracking through infrared-visible image fusion,” in *Proceedings of the 14th International Conference on fusion (FUSION)*, 2011.
- [3] Michael Teutsch, Wolfgang Kruger, and Jürgen Beyerer, “Fusion of region and point-feature detections for measurement reconstruction in multi-target kalman tracking,” in *Proceedings of the 14th International Conference on fusion (FUSION)*, 2011.
- [4] U. Hofmann, André Rieder, and Ernst D. Dickmanns, “Radar and vision data fusion for hybrid adaptive cruise control on highways,” in *Proceedings of the Second International Workshop on Computer Vision Systems*, London, UK, UK, 2001, ICVS ’01, pp. 125–138.
- [5] N. Kaempchen, M. Buehler, and K. Dietmayer, “Feature-level fusion for free-form object tracking using laserscanner and video,” in *Intelligent Vehicles Symposium (IV)*. IEEE, 2005.
- [6] Olivier Aycard and Anne Spalanzani et al., “Grid based fusion tracking,” in *Proceedings of the Intelligent Transportation Systems Conference*, 2006.
- [7] Raphaël Labayrade, Cyril Royere, Dominique Gruyer, and Didier Aubert, “Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner,” *Auton. Robots*, vol. 19, pp. 117–140, September 2005.
- [8] F. Garcia and D. Olmeda, “Hybrid fusion scheme for pedestrian detection based on laser scanner and far infrared camera,” in *Proceedings of the Intelligent Transportation Systems Conference*, 2010.
- [9] Wenyin Tang, K. Z. Mao, Lee Onn Mak, and Gee Wah Ng et. al, “Target classification using knowledge-based probabilistic model,” in *Proceedings of the 14th International Conference on fusion (FUSION)*, 2011.
- [10] Biruk K. Habtemariam, R. Tharmarasa, and T. Kirubarajan et. al, “Multiple detection probabilistic data association filter for multistatic target tracking,” in *Proceedings of the 14th International Conference on fusion (FUSION)*, 2011.
- [11] Sean Martin, “Sequential bayesian inference models for multiple object classification,” in *Proceedings of the 14th International Conference on fusion (FUSION)*, 2011.
- [12] L.A. and Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [13] Martin A. Fischler and Robert C. Bolles, “Readings in computer vision: issues, problems, principles, and paradigms,” chapter Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, pp. 726–740. San Francisco, CA, USA, 1987.
- [14] Bertrand Douillard et al., “A pipeline for the segmentation and classification of 3d point clouds,” in *International Symposium on Experimental Robotics 2010*, 2010.
- [15] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, Sept. 2004.
- [16] J. Zhang, M. Marsza, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *Int. J. Comput. Vision*, vol. 73, no. 2, pp. 213–238, June 2007.
- [17] Lan Du, Lu Ren, David B. Dunson, and Lawrence Carin, “A bayesian model for simultaneous image clustering, annotation and object segmentation,” in *NIPS*, Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta, Eds. 2009, pp. 486–494, Curran Associates, Inc.
- [18] Aude Oliva and Antonio Torralba, “The role of context in object recognition,” *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 52–527, 2007.
- [19] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *Int. J. Hum.-Comput. Stud.*, vol. 51, no. 2, pp. 135–147, Aug. 1999.

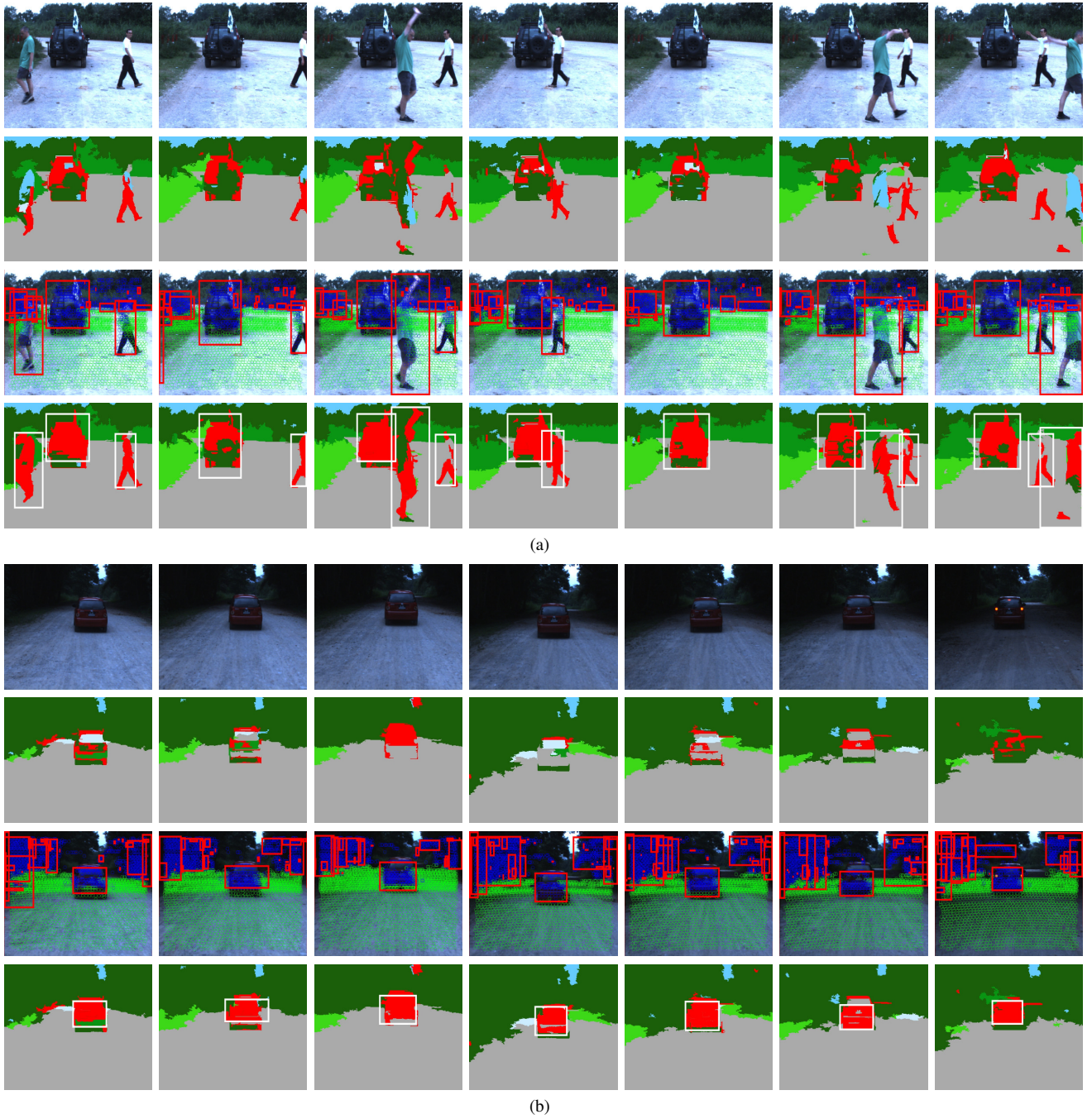


Fig. 7. Sample results of scene classification and obstacle detection using two datasets in rural area. The panel (a) presents the result of dataset 1 while the panel (b) presents the result of dataset 2. In each panel, the first row shows the original camera image and the second row shows the image classification result. The red color illustrates the region of detected obstacles and other colors have similar meaning as in Fig. 3(d). The third row shows the detected result using scanner pointcloud and the results are projected to the camera image. Each bounding box localizes one candidate obstacle. The green circles represent the projection of ground points and the blue circles represent the projection of above-ground points. There are ground points inside several bounding boxes due to the 3D-2D projection. The fourth row shows the result of our fusion method. Each white bounding box localizes one detected obstacle.