Fusion of 3D LIDAR and Camera Data for Scene Parsing

Gangqiang Zhao\textsuperscript{1}, Xuhong Xiao\textsuperscript{2}, Junsong Yuan\textsuperscript{1}, Ng Gee Wah\textsuperscript{2}

\textsuperscript{1}School of EEE, Nanyang Technological University, Singapore
\textsuperscript{2}DSO National Laboratories, Singapore

Abstract

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 parsing and data fusion for a 3D-LIDAR scanner (Velodyne HDL-64E) and a video camera is described. First of all, a geometry segmentation algorithm is proposed for detection of obstacles and ground areas from data collected by the Velodyne scanner. Then, the corresponding image collected by the video camera is classified patch by patch into more detailed categories. After that, parsing result of each frame is obtained by fusing the result of Velodyne data and that of image using the fuzzy logic inference framework. Finally, parsing results of consecutive frames are smoothed by the Markov random field based temporal fusion method. The proposed approach has been evaluated with datasets collected by our autonomous ground vehicle testbed in both rural and urban areas. The fused results are more reliable than that acquired via analysis of only images or Velodyne data.

Keywords:
scene parsing, Velodyne scanner, camera, fuzzy logic, temporal fusion, MRF
1. 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. Studies of the human visual system show us that scene perception is a highly complex process of information fusion which involves not just the human eyes, but also other human senses including hearing, tasting, etc. Even within a human vision system, there is clearly fusion of information from color, motion, depth and a whole variety of ways to infer shape, movement and physical characteristics of the things within the view [1]. In other words, efficient perceptual performance often requires integration of multiple sources of information, both within the senses and between them. As a matter of fact, other sensors like infrared laser projector in Kinect [2] and LIDAR scanners [3] have been applied to complement video cameras in recent years.

In this work, in order to help unmanned vehicles to understand their environment, two sensors are used: Velodyne HDL-64E 3D-LIDAR scanner [3] and monocular video camera. A Velodyne scanner provides 3-dimensional but sparse pointcloud of the surrounding environment. The pointcloud is trustworthy for obstacle detection but lacks color and texture information, which is valuable for more detailed categorization of objects. Besides, although Velodyne HDL-64E is a powerful LIDAR scanner in the market, its effective coverage limits within 70 meters from the center of the sensor. Con-
sidering some time will be taken for information processing and task scheduling, the 70-meter distance may not be sufficient for an unmanned moving vehicle to respond. Furthermore, for some tasks, we hope the vehicle can “see” as far as 200 meters for advanced planning. On the contrary, images captured by video cameras can easily cover a much broader and further 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 above-mentioned complementary features between cameras and LIDAR sensors, it is possible to acquire more reliable scene parsing by fusing the information derived from these two sensors.

In addition, the sequential scene parsing also requires fusing the results of consecutive frames. In fact, even after fusing the results of two sensors, the parsing results of consecutive frames may have abrupt changes due to stochastic errors. These abrupt changes of parsing results may confuse the vehicle navigation system. Intuitively, it is possible to obtain more cohesive sequential parsing results by including the temporal fusion.

In this research, we first propose a new way to fuse the results of two sensors by employing fuzzy logic inference [4]. Then we propose a Markov random field based approach to fuse the results of consecutive frames. Fig. 1 illustrates the fusion process. Fuzzy logic is preferable for our application due to its advantages. First, 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 a priori knowledge. Second, fuzzy logic can model nonlinear functions of arbitrary complexity. This is important
Figure 1: Illustration of the proposed sensor fusion approach. The data of Velodyne scanner and camera are first parsed simultaneously. Then the results of two sensors are fused by the proposed fuzzy logic based method. After that, the parsing results of consecutive frames are smoothed by the proposed Markov random field based temporal fusion method. By fusing results of two sensors, it localizes the obstacles correctly and therefore improves the scene parsing result.

As scene parsing is not a trivial problem. Third, fuzzy logic can tolerate imprecise results of two sensors. Moreover, fuzzy logic is a flexible fusion framework so that results of more sensors can be easily integrated to the system in future.

To fuse the results of consecutive frames, we propose a Markov random field (MRF) based temporal fusion method [5][6]. Correspondences between consecutive frames are first estimated by the dense optical flow method [7]. Then, a MRF model is built to integrate the results of multiple consecutive frames. The result of each frame is refined by the Belief Propagation (BP) algorithm [8]. The following contributions have been made in this paper:
1. 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.

2. The MRF based temporal fusion method is introduced to obtain the cohesive video parsing results. It can smooth whole frame simultaneously by integrating the results of multiple consecutive frames.

3. We test the proposed approach on datasets collected by our autonomous ground vehicle testbed. The datasets are captured from urban and rural areas either in day or night time. The results validate the robustness and effectiveness of our fusion approach for scene parsing.

A preliminary version of this paper was described in [9]. The current version described here differs from the former in several ways, including: the introduction of MRF based temporal fusion method; more comprehensive evaluation on the method with three more databases; further analysis and discussion of the whole approach, as well as the introduction of more related works about sensor fusion and scene parsing. While the preliminary version in [9] focuses on fuzzy logic based fusion strategies, the current version will provides more details on image parsing techniques, too.

This paper is organized as follows: In Section 2, we briefly survey the sensor fusion and scene parsing literature. After giving the parsing methods for individual sensors, we describe the fuzzy logic based method to fuse the results of two sensors in Section 4. The MRF based temporal fusion is presented in Section 5. Thorough experiments are conducted in Section 6 for evaluation, and in-depth discussion is provided in Section 7. We conclude
our paper in Section 8.

2. Related work

By combining data from multiple sensors, we can achieve improved accuracies and more specific inferences than that achieved by the use of a single sensor alone [10]. The existing methods for fusing LIDAR data and camera images can be grouped into three categories: centralized approaches, decentralized approaches and hybrid approaches. In centralized approaches, the fusion process occurs at the pixel-level or feature level, i.e., features from both LIDAR and video camera are combined in a single vector for posterior classification. Douillard et al. present a rule based 3D classifier by combining Velodyne data and monocular camera data [11]. A set of twenty one binary features are defined based on the 3D pointclouds and the camera image. The logical rules are learned from training data. Häselich et al. present a novel approach for online terrain classification from fused camera and laser range data [12]. Laible et al. propose to handle the terrain classification at different lighting conditions by fusing the camera and LIDAR data[13]. Kaempchen et al. perform centralized free-form object tracking using laser scanner and video [14]. Schneider et al. address the problems of synchronization, correction and occlusion reasoning for the fusion of Vision and LIDAR [15].

Centralized methods can simplify the posterior classification 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 one sensor.
Decentralized approaches separately classify the data for individual sensor first, the classification results are then combined by a fusion method. Kidono et al. propose a fusion system for reliable pedestrian recognition using Velodyne and a vision sensor to achieve high performance under various conditions [16]. Himmelsbach et al. propose to evaluate the tentacle by fusing LIDAR and camera for autonomous navigation [17]. Labayrade et al. propose a fusion strategy by matching the set of obstacles from laser scanner with the set of obstacles coming from stereo vision based on the belief theory [18]. Premebida et al. also obtain a better performance than the single classifiers by using the decentralized scheme [19]. 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 [20] [21] [22] [23]. Garcia and Olmeda propose a hybrid fusion strategy by fusing the low and high level information simultaneously [20]. Tang et al. propose to learn the contextual information from input data and then combined with given expert knowledge in classification [21]. Habtemariam et al. propose a multiple detection probabilistic data association (MD-PDA) filter for tracking a target when more than one target originated measurement may exist within the validation gate [22]. Martin explores another type of fusion by updating the classifications of multiple objects simultaneously when given a measurement on only one of the objects [23]. Matthaei and Dyckmanns use laser and radar to classify motion for cross traffic in urban environments [24].

Now we take an overview of the related scene parsing works. Image scene
parsing aims to assign a category label to each pixel of a given image. Over the last several years, many methods have been proposed for this problem. They can be broadly categorised on the basis of their basic process units. Several methods are using the pixels as basic units [6], others using segments [25][26][27], group of segments [28], or intersections of multiple segmentations [29], while the whole image is considered in the extreme case [30]. Several methods are using multiple types of information to improve the parsing results. Tu et al. propose to combine segmentation, detection, and recognition for the scene parsing [31]. Ladicky et al. [32] propose an image segmentation and parsing method by combining object recognition, detection and segmentation with a conditional random field defined on pixels, segments and objects. Felzenszwalb and Veksler propose a tiered scene labeling method by using the dynamic programming approach [33].

Other scene parsing methods employ the nonparametric classification method [5][34] or deep feature learning [35]. Liu et al. [5] propose a nonparametric scene parsing method via label transfer algorithm. Tighe and Lazebnik [34] pre-process the video using a spatiotemporal segmentation method that gives 3D regions that are spatially coherent within each frame as well as temporally coherent between frames. Then each 3D region is classified. Farabet et al. propose a scene parsing method by leveraging the deep learning method [35].

Besides the image parsing work, several approaches have tried many strategies to employ the cues contained in video data. Brostwo et al. [36], Strurgess et al. [37] and Zhang et al. [38] recover the 3D structure information (e.g., dense depth maps or sparse point clouds) from the video sequences
and then combine the 3D information and image information to parse individual frames. Xiao and Quan [39] propose a region-based parsing system on each frame and enforce temporal coherence between regions in adjacent frames by temporal fusion in a batch model.

With the development of range sensors, several recent works obtain the scene semantic labels with the 3D LIDAR data only. Spinello et al. track the people in 3D pointcloud data using a bottom-up top-down pedestrian detector [40]. Bradley et al. employ the 3D pointcloud to detect vegetation for driving in complex environments [41]. Teichman and Thrun propose a semi-supervised approach to the problem of track classification in dense 3D range data [42]. Behley et al. evaluate several local features for the classification of 3D laser range data in urban environments [43].

3. Parsing modules for individual sensors

As a decentralized fusion method, a geometry segmentation algorithm is proposed to detect obstacles 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 advantages and disadvantages.

3.1. 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 [44]. 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 inliers points should be near the hypothesis plane (i.e., the distance to the plane is less than 0.3 meter). The RANSAC algorithm terminates after testing 100 hypothesis planes. 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 non-empty above-ground voxels according to 3D adjacency [45]. Each obstacle is represented by a voxel cluster. Denote all the voxel clusters as $\mathcal{C}$ and the voxels
Figure 2: Illustration of obstacle and ground classification using Velodyne scanner. (a) is the 3D pointcloud of Velodyne scanner; (b) is the ground points; (c) is the above-ground points; (d) shows the detected bounding boxes of the candidate obstacles; (e) shows the pointcloud and the detected results; (f) shows the detected results which are projected to the camera image. Each bounding box represents one candidate obstacle in (d), (e) and (f).
Table 1: The category definition

<table>
<thead>
<tr>
<th>Category</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground</td>
<td>traversable ground plane</td>
</tr>
<tr>
<td>Obstacle</td>
<td>pedestrian, vehicle and other objects above ground</td>
</tr>
<tr>
<td>Building</td>
<td>human-made structure</td>
</tr>
<tr>
<td>Grass</td>
<td>vegetation with height less than 0.3 meter</td>
</tr>
<tr>
<td>Bush</td>
<td>vegetation with height between 0.3 meter and 2.0 meter</td>
</tr>
<tr>
<td>Tree</td>
<td>vegetation with height more than 2.0 meter</td>
</tr>
<tr>
<td>Pavement</td>
<td>/</td>
</tr>
<tr>
<td>Sky</td>
<td>/</td>
</tr>
<tr>
<td>Water</td>
<td>/</td>
</tr>
</tbody>
</table>

in one voxel cluster $O \in \mathcal{C}$ should meet the following 3D adjacency criterion:

$$\forall V_{i,j,k} \in O \ ( \exists V'_{i',j',k'} \in O) \land (|i - i'| < d \lor |j - j'| < d \lor |k - k'| < d),$$

(1)

We set $d = 2$ in current implementation. The detected results are projected to the image as shown in Fig.2(f). 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.

3.2. Image parsing module

Contrary to Velodyne information processing which concerns whether the terrain is traversable [46], we intend to identify more specific categories of objects from the images. From the cameras images that we have collected, we have identified nine possible categories, which include ground (road), build-
ing, water, tree, grass and obstacles, etc. As a matter of fact, obstacles can be further divided into human, car, 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 obstacles. For a particular task, specific models can be trained for individual interesting classes too. Table 1 summarizes all defined categories. 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 [47]. 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 [48]. An MLP (multilayer perceptron) classifier is trained to classify the local image patches into object categories [49]. 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 “tree” (area B) is classified into “water”, and a part of “grass” (area C) is classified as “tree”. 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
Figure 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. The classification errors in area A, B and C in (b) are corrected after top-down contextual analysis.

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 “tree” in the image if it is taken from a moving vehicle. This property has been well recognized and
employed in several computer vision systems [50]. 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, where 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). For each direction, we check the existence of each category, as well as whether the component is adjacent to the boundary of the image. 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 model was learned using the TAN (tree augmented naive Bayes) training algorithm [51]. Therefore, for each node, except the root node, there will be at most two parents. Fig. 4 illustrates the corresponding Bayesian network with three 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 up-
Figure 4: Illustration of the Bayesian networks for contextual analysis. Here we show four categories “Tree”, “Road”, “Obstacle” and “Image” (i.e., pseudo category for image boundary) for drawing convenience. The root node “class” corresponds to the category of the component under consideration.

dated 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.

3.3. 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. 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 water,
Figure 5: Illustration of the range covered by Velodyne. (a) shows the camera image. (b) shows the image with the projected Velodyne points. It can be seen that only nearby areas have the Velodyne points.

sky and the areas out of the sensor’s range, as shown in Fig.5. 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 multiple categories. 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 parsing and obstacle detection by combining them.

4. Fuzzy logic based sensor fusion

Both the results of laser scanner and the results of camera image have their own advantages and disadvantages. To parse the scene correctly, the primary work of fusion is to categorize the detected candidate obstacles by Velodyne scanner. The scene parsing results are then improved based on the categorization. As a good way to utilize the a priori knowledge and experience of human experts [4], we propose to use the fuzzy inference method to fuse the results of two sensors.

4.1. 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 (rc) is the detection result for the candidate obstacles. Each input and output 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 four linguistic labels TIN (tiny), SMA (small), MID (middle)
and LAR (large) within the universe of discourse (0, 100) percents. The candidate obstacle size is defined as the percents of all image pixels which are inside the candidate obstacle bounding box. The variable class is expressed using three linguistic labels NOBS (non-obstacle), MID (middle) and OBS (obstacle) within the universe of discourse (0, 100) percents. The classification is measured by the percent of non-obstacle pixels among all the pixels inside the candidate obstacle bounding box. All the detected grass, bush, tree and building pixels by image classification method are considered as non-obstacle pixels. When most of inside pixels belong to the non-obstacle category, the candidate obstacle is probably not the pedestrian and vehicle, and vice versa. We count the number of non-obstacle pixels to describe the bounding boxes as the tree, bush or building areas are possible to be detected as candidate obstacles while they might be far from the traversable ground area. Therefore, by removing the tree, bush or building from the candidate obstacles, the autonomous vehicle will focus on the obstacles which are above on the traversable ground.

The spatial context s-context is expressed using two linguistic labels NOBS (non-obstacle) 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. It is important to note that the flat-world assumption is used here to make the absolute height work.

The output result score \((r_c)\) is simply expressed using three linguistic labels NOBS (non-obstacle), 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. 6.

4.2. 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 obstacle is also important knowledge. When the candidate obstacle is surrounded by ground region, it is probably an obstacle. However, when the candidate obstacle is on the edge of ground region, its categorization highly depends on image classification result and other information like height of the obstacle. By analyzing the application scenario of our auto-driving vehicle, the following rules are selected.
Figure 6: 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 $r_c$. 
The group of rules when the size of object box is large:

\begin{align*}
R_1 & : \text{if size is LAR and class is OBS then } r_c \text{ is OBS}; \\
R_2 & : \text{if size is LAR and class is MID then } r_c \text{ is MID}; \\
R_3 & : \text{if size is LAR and class is NOBS then } r_c \text{ is NOBS}; \\
R_4 & : \text{if size is LAR and class is NOBS and s-context is NOBS then } r_c \text{ is NOBS}; \\
R_5 & : \text{if size is LAR and class is NOBS and t-context is NOBS then } r_c \text{ is NOBS}; \\
R_6 & : \text{if size is LAR and class is NOBS and s-context is OBS then } r_c \text{ is OBS}; \\
\end{align*}

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:

\begin{align*}
R_7 & : \text{if size is MID and class is OBS then } r_c \text{ is OBS}; \\
R_8 & : \text{if size is MID and class is MID then } r_c \text{ is MID}; \\
R_9 & : \text{if size is MID and class is NOBS then } r_c \text{ is NOBS}; \\
R_{10} & : \text{if size is MID and class is NOBS and s-context is NOBS then } r_c \text{ is NOBS}; \\
R_{11} & : \text{if size is MID and class is NOBS and t-context is NOBS then } r_c \text{ is NOBS}; \\
R_{12} & : \text{if size is MID and class is NOBS and s-context is NOBS then } r_c \text{ is NOBS}; \\
R_{13} & : \text{if size is MID and class is NOBS and s-context is NOBS then } r_c \text{ is NOBS}; \\
\end{align*}

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:

\begin{align*}
R_{14} & : \text{if class is NOBS and height is MID then } r_c \text{ is NOBS}; \\
R_{15} & : \text{if class is MID and height is MID then } r_c \text{ is NOBS}; \\
R_{16} & : \text{if height is HIG then } r_c \text{ is NOBS}; \\
\end{align*}

Although 20 rules do not cover the complete relationships of different attributes, these rules help to integrate the results of two sensors and the
4.3. Fuzzy reasoning

After synthesizing these 20 rules for fusion, their roles are further coordinated through Mamdani’s fuzzy reasoning method in this section [52]. The process of Mamdani fuzzy inference involves steps fuzzification, inference, aggregation and defuzzification. The information flow of the fuzzy reasoning is shown in Fig. 7.

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. 6. 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 $size^*$ and class*, the output grade $r_s^*$ of the label OBS due to this rule can be inferred as:

$$U_{OBS}^7(r_s^*) = \min(U_{MID}(size^*), U_{OBS}(class^*))$$  \hspace{1cm} (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:

$$U_{OBS}(r_s^*) = \max( U_{OBS}^1(r_s^*), U_{OBS}^2(r_s^*), \cdots, U_{OBS}^{20}(r_s^*) )$$  \hspace{1cm} (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 $U_{NOBS}(r_s)$ respectively at the grades $U_{OBS}(r_s)^*$, $U_{MID}(r_s)^*$ and $U_{NOBS}(r_s)^*$, and combing them point by point:

$$U_O(r_s) = \max( \min(U_{OBS}(r_s^*), U_{OBS}(r_s)), \min(U_{MID}(r_s^*), U_{MID}(r_s)), \min(U_{NOBS}(r_s^*), U_{NOBS}(r_s)) )$$  \hspace{1cm} (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:

$$r_s^* = \frac{\int U_O(r_s) r_s \, dr_s}{\int U_O(r_s) \, dr_s}$$  \hspace{1cm} (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.

4.4. Automatic setting the fuzzy logic inference

To fuse the results of two sensors, the fuzzy logic is employed by defining the fuzzy variables and fuzzy rules. Although the fuzzy rules and fuzzy variables are decided manually by analyzing the application scenarios in our implementation, the neuro-fuzzy approach can select the rules and tune the parameters automatically [53]. Neuro-fuzzy approach combines the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Through the neuro-fuzzy approach, the proposed fuzzy logic based fusion method is easy to be applied in many different applications.

5. Temporal fusion of consecutive frames

By fusing the results of camera and scanner, we can have a better parsing result of each frame. The image parsing result is helpful to understand the environment for the ground vehicle. However, the results of consecutive frames may have abrupt changes due to the car moving, partial occlusion, etc. Fig.8 shows this phenomenon and several incohesive regions are marked in one frame by the white circles. These abrupt changes of parsing results will mislead the vehicle navigation system. One major reason of cohesiveness
problem is that the temporal information is not included in the scene parsing process.

There are several challenges to do the temporal fusion for video scene parsing. First, the whole frame should be considered simultaneously to obtain the spatial coherence for all pixels. Second, to do the temporal fusion, the pixels should be matched densely between consecutive frames. Therefore, we cannot use the sparse feature matching algorithm as hundreds of visual features are not enough to cover the whole frame. Moreover, one single previous frame may not have the enough information for the temporal fusion and multiple previous frames are required.

To address the above problems for the temporal fusion, we model each frame as a Markov random field (MRF) [5][6]. The correspondences between two consecutive frames are first estimated by the dense optical flow method [7]. Then each frame is represented by a MRF model to integrate the results of multiple previous frames. After that the result of each frame is refined by solving the MRF model using the Belief Propagation (BP) algorithm [8]. Fig.9 describes the MRF model.

For a given frame $I_t$, we consider its $k$ previous frames for temporal fusion. Each previous frame $I_i$ is described by its initial scene parsing result $c_i$ and the optical flow field $v_i$. All the $k$ previous frames are denoted by the set \( \{c_i, v_i\}_{t-k \leq i \leq t-1} \). For previous frame $I_i$, $c_i(p) \in [0, 1]^L$ and $|c_i(p)| = 1$ is the category probability vector for pixel $p$ obtained by the image parsing algorithm. Here we assume there are a total of $L$ categories as defined in Table 1. $c_i(p)^l$ represents the probability of pixel $p$ is classified to category $l$. $v_i$ is the optical flow field (from $I_t$ to $I_i$). We want to obtain the smoothed
Figure 8: The cohesive problem between consecutive frames. (a) and (b) are two consecutive frames while (c) and (d) are the corresponding image parsing results. The white circles in (d) illustrate several places which do not have the cohesive parsing results between two frames.

Parsing result $c_t^s$ for the given frame $I_t$ by fusing the result of frame $I_t$ and that of $k$ previous frames. Therefore we build a probabilistic Markov random field model to integrate the results of multiple frames and the spatial smoothness
constraint. Inspired by [5] and [6], the posterior probability is defined as:

\[-\log P(c^*_t|c_t, \{c_i, v_i\}) = \sum_p \phi_1(c^*_t(p); c_t) + \sum_p \phi_2(c^*_t(p); c_{t-1}, v_{t-1}) + \sum_p \phi_3(c^*_t(p); \{c_i, v_i\}) + \sum_{\{p,q\} \in \epsilon} \phi_4(c^*_t(p), c^*_t(q); c_t) + \log Z\]

where $Z$ is the normalization constant of the probability and $\epsilon$ is the set which represents the neighborhood relation of all pixels in frame $I_t$. $p$ and $q$ are pixels in frame $I_t$. Among the four components of this posterior, $\phi_1$ ensures the smoothed result similar to the parsing result of current frame $c_t$ while
\( \phi_2 \) forces the smoothed result close to the parsing result of previous frame \( c_{t-1} \). \( \phi_3 \) depends on the parsing results of corresponding pixels in multiple previous frames \( \{c_i, v_i\}_{t-k \leq i \leq t-1} \) and \( \phi_4 \) incorporates the spatial smoothness constraints which depend on the smooth parsing result of current frame \( c_t^s \).

The optical flow field \( \{v_i\}_{t-k \leq i \leq t-1} \) is used to find the corresponding points between current frame and the previous frames.

The first term \( \phi_1 \) is defined as:

\[
\phi_1(c_t^s(p) = l) = (1 - c_t(p))^l \quad (7)
\]

where \( c_t(p)^l \) represents the probability of the pixel \( p \) is labeled as category \( l \) in image parsing result. The higher the probability \( c_t(p)^l \), the more chance the smoothed parsing result of pixel \( p \) is set to be \( l \). The second term \( \phi_2 \) is defined as:

\[
\phi_2(c_t^s(p) = l) = \begin{cases} 
\frac{\|I_t(p) - I_{t-1}(p_{t-1})\|}{\tau} & \text{if } \exists p_{t-1} \\
(1 - c_{t-1}(p))^l & \text{else} 
\end{cases} \quad (8)
\]

where \( p_{t-1} = p + v_{t-1}(p) \) is \( p \)'s corresponding pixel in previous frame \( I_{t-1} \). \( \tau \) is set to be the maximum intensity difference value between corresponding pixels of two frames \( \tau = \max_{p, p_{t-1}} \|I_t(p) - I_{t-1}(p_{t-1})\| \).

The term \( \phi_3 \) incorporates the probability that category \( l \) appears at pixel \( p \)'s corresponding pixels in several previous frames. \( \phi_3 \) is considered as the historical prior for the category \( l \) and its value is obtained from counting the occurrence of category \( l \) at pixel \( p \)'s corresponding pixels in the \( k \) previous frames:

\[
\phi_3(c_t^s(p) = l) = -\log(N_l + 1) \quad (9)
\]
where \( N_l \) is the occurrence number of category \( l \) in \( p \)'s corresponding pixels in the \( k \) previous frames. The smoothness term \( \phi_4 \) compels the neighboring pixels to have the same label in the event that no other information is available and its value depends on the parsing result of current frame \( c_t \):

\[
\phi_4(c_t^*(p), c_t^*(q)) = \delta[c_t^*(p) \neq c_t^*(q)] \|c_t(p)^{t*} - c_t(q)^{t*}\|
\] (10)

where \( c_t(p)^{t*} \) represents the maximum probability value in the category probability vector of pixel \( p \). To compel the neighboring pixels have the same label, \( \delta[c_t^*(p) \neq c_t^*(q)] \) is set to be 1 when \( c_t^*(p) \neq c_t^*(q) \) and it is set to be 0 when \( c_t^*(p) = c_t^*(q) \). \( \phi_4 \) can add a penalty if two neighboring pixels have different smoothed labels. Once the energy functions are calculated for frame \( I_t \), we use the BP algorithm to minimize the energy [8] and the parsing result is smoothed consequently.

6. Performance evaluation

To evaluate our fusion approach, we test it on four datasets collected by our autonomous ground vehicle testbed when driving in rural and urban areas and one public pedestrian dataset [54]. In the experiment, we compare the fusion result with that of using video camera only. In addition, the MRF based temporal fusion method is further evaluated.

6.1. Dataset and sensor calibration

The 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 calibration of extrinsic parameters is done in a coarse-to-fine manner. We first estimate the extrinsic parameters of the
camera using the Caltech calibration tool box [55]. Then we initially estimate
the tilt, roll and yaw of the camera with regard to the world coordinate system
based on the estimated vanishing line on selected images. In this step, we
assume the Velodyne coordinate system and the world coordinate system are
aligned. After obtaining the initial result, we fine-tune the parameters based
on the mapping result between Velodyne points and image pixels.

The first dataset corresponds to an open ground in rural area while the
second one corresponds to the road in rural area. As shown in Table 2, the
first dataset consists about 440 frames and the second one consists about 450
frames. The third and fourth datasets correspond to the road in urban area
while they both have about 1500 frames. The first three datasets are collected
in the day time while the fourth one is collected in the night time. Besides
these four self-collected datasets, we select the challenging pedestrian data
from the recent public dataset [54] as our fifth dataset. Table 2 summaries
all datasets. To quantify the performance of the proposed approach, we
manually labelled about 20% of all frames in the first and second datasets,
and 5% of all frames in the third and fourth datasets. There are total 9
labelled categorizes which include road, obstacle, building, tree, sky, water,
etc. For the fifth dataset, we use the groundtruth of pedestrians provided
by [54].

The proposed scene parsing approach aims to provide the environment
situation awareness ability for the autonomous ground vehicles. We evaluate
the fusion method on the collected datasets. The $k = 4$ previous frames are
considered in the temporal fusion step and $\epsilon$ contains eight neighbours of each
pixel in order to obtain the spatial smooth. Let $DR$ and $GT$ be the discovered
Table 2: The information of five datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sensor</th>
<th>Frame No.</th>
<th>Label level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset 1</td>
<td>Camera + Velodyne</td>
<td>440</td>
<td>Pixel</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>Camera + Velodyne</td>
<td>450</td>
<td>Pixel</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>Camera</td>
<td>1500</td>
<td>Pixel</td>
</tr>
<tr>
<td>Dataset 4</td>
<td>Camera</td>
<td>1500</td>
<td>Pixel</td>
</tr>
<tr>
<td>Dataset 5 [54]</td>
<td>Camera + Velodyne</td>
<td>150</td>
<td>Bounding Box</td>
</tr>
</tbody>
</table>

object regions and the bounding boxes of ground truth, respectively. For each category, the set of pixels which are classified to this category by our method is denoted as $DR$ (i.e., detect region). The set of pixels which are manually labelled to this category is denoted as $GT$ (i.e., ground truth). 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 $recall$. In each frame, these criteria values are first calculated for each category, respectively. Then the average value of all categories is used to evaluate one frame.
Figure 10: Sample results of scene parsing and obstacle detection using Dataset 1 and Dataset 2. 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 fourth row shows the result of our fusion method. Each white bounding box localizes one detected obstacle.
Figure 11: The performance evaluation of the proposed fusion based scene parsing method (Fusion) and the image parsing method (Camera Only) using Dataset 1 and Dataset 2. The results of all categorizes are averaged to obtain the $F$-measure, precision and recall values for each frame. In each figure, the frames are ordered according to the $F$-measure value of fusion 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.
Figure 12: The obstacle detection performance of the proposed fusion based scene parsing method (Fusion) and the image parsing method (Camera Only) using Dataset 1 and Dataset 2. In each figure, the frames are ordered according to the result of fusion 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.
6.2. Scene parsing by fusing results of two sensors

Fig. 10 shows the result of several frames. 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 frames contain only one obstacle and some frames do not contain any obstacles. These results show that the proposed approach performs well for scene parsing and obstacle detection from real-world driving environment.

To further evaluate the proposed method, we compare its result with that of using image only. As shown in Fig. 11, our proposed fusion approach improves the scene parsing result in terms of $F$-measure value. The position of obstacles is very important information in the scenario of autonomous ground vehicles. The obstacle parsing evaluation is shown in Fig. 12 and our proposed fusion approach improves the obstacle parsing 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 parsing 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.

6.3. Pedestrian detection using the fusion method

Recently, Geiger et al. published one dataset which have both the data of camera and Velodyne laser scanner [54]. As this dataset does not provide the pixel-level category label, we evaluate the fusion method for pedestrian detection only. We first detect the pedestrians from the camera images using the method provided by [56]. Each detected pedestrian is located by a bounding box and the corresponding image classification result class is set to be the pedestrian detection response [56]. Other fuzzy variables are set according to description of sec. 4.1. Then we apply the proposed sensor fusion method to remove the false pedestrian detection. Fig. 13 shows the sample results of pedestrian detection. It can be seen that by fusing data of camera and Velodyne, we can remove many false detections. The quantitative evaluation is shown in Fig. 14 and our proposed fusion approach improves the pedestrian detection significantly in terms of F-measure value. However, there is clearly room for improve the detection performance. For example, detect pedestrians who are in shadow area using the Velodyne pointcloud directly as [40].
Figure 13: Sample results of pedestrian detection using Dataset 5 [54]. The panel (a) presents the result of pedestrian detection using the image only; the panel (b) shows the image with the projected Velodyne points; the panel (c) shows the pedestrian detection result using the proposed sensor fusion method. Each bounding box represents one detected pedestrian. By fusing of camera and Velodyne, we can remove many false detections. Meanwhile there is clearly room for improvement. For example, detect pedestrians in shadow area. Better see in color.
Figure 14: The pedestrian detection performance of the proposed fusion based method (Fusion) and the image based method [56] (Camera Only) using Dataset 5. In each figure, the frames are ordered according to the results of fusion method. (a) shows the F-measure value of Dataset 5; (b) shows precision value of Dataset 5; (c) shows recall value of Dataset 5.

6.4. Evaluate the temporal fusion method

The temporal fusion method is proposed to smooth the results of consecutive frames as the abrupt changes of parsing results will mislead the vehicle navigation. Fig.15 shows the sample scene parsing results of consecutive frames using two datasets in urban area. In each panel, the first row shows the original camera image. The second row shows the image parsing result before the temporal fusion and it can be seen that each frame has several places which are not cohesive with its consecutive frames. After the temporal fusion, the results are more cohesive between consecutive frames, as shown in the third row. To measure the cohesive performance of the proposed approach, we define a Jump factor criterion for each frame:

\[
\text{Jump factor} = \frac{\text{No. of pixels changing label}}{\text{No. of all pixels}}
\]
The *Jump factor* represents the ratio of pixels which have different labels with their corresponding pixels in the previous frame and the *Jump factor* is used to qualitative evaluation of the proposed temporal fusion method. The pixel to pixel correspondence between two consecutive frames are obtained by the dense optical flow method [7]. The larger the *Jump factor*, the more pixels in one frame have changed their parsing labels comparing with the previous frame. Fig.16 shows the comparison of the *Jump factor* values with or without using the temporal fusion in four datasets. It can be seen that the *Jump factor* values are significantly reduced by the proposed temporal fusion method.

It is important to note that temporal fusion will introduce a latency in scene parsing result. This can be seen in Fig.15(a). Although pixels behind the vehicle are classified to be road by the image parser, the fusion method adopts these new measurements after several frames.

To further evaluate the proposed Markov random field based temporal fusion method, we compare its result with that of multiple frames voting (MFV) method. The multiple frames voting method decides the label of pixel $p$ in current frame by the voting of pixel $p$ and its corresponding pixels in the $k = 4$ previous frames. The label with maximum votes is assigned to pixel $p$. Table 3, Table 4, Table 5 and Table 6 show the comparison of F-measure values for four datasets. As Dataset 5 only has the bounding box label, we do not evaluate the temporal fusion performance on it. It can be seen that the proposed MRF based temporal fusion method can obtain a better performance with more than 3% improvement in terms of the average F-measure values for these datasets. The MRF based temporal fusion method
has a better performance in Dataset 1 and Dataset 2 as the obstacles are moving fast in these two datasets. These comparisons clearly demonstrate the advantages of the proposed MRF based temporal fusion method.

Table 3: The $F$-measure of two temporal smoothing methods in Dataset 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Obstacle</th>
<th>Road</th>
<th>Bush</th>
<th>Tree</th>
<th>Sky</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>0.76</td>
<td>0.97</td>
<td>0.70</td>
<td>0.84</td>
<td>0.81</td>
<td>0.59</td>
</tr>
<tr>
<td>MFV</td>
<td>0.48</td>
<td>0.96</td>
<td>0.68</td>
<td>0.77</td>
<td>0.77</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Table 4: The $F$-measure of two temporal smoothing methods in Dataset 2.

<table>
<thead>
<tr>
<th>Method</th>
<th>Obstacle</th>
<th>Road</th>
<th>Bush</th>
<th>Tree</th>
<th>Sky</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>0.79</td>
<td>0.94</td>
<td>0.21</td>
<td>0.91</td>
<td>0.68</td>
<td>0.44</td>
</tr>
<tr>
<td>MFV</td>
<td>0.51</td>
<td>0.93</td>
<td>0.24</td>
<td>0.90</td>
<td>0.67</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Table 5: The $F$-measure of two temporal smoothing methods in Dataset 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>Obstacle</th>
<th>Road</th>
<th>Building</th>
<th>Tree</th>
<th>Sky</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>0.35</td>
<td>0.82</td>
<td>0.67</td>
<td>0.47</td>
<td>0.57</td>
<td>0.30</td>
</tr>
<tr>
<td>MFV</td>
<td>0.31</td>
<td>0.81</td>
<td>0.66</td>
<td>0.44</td>
<td>0.53</td>
<td>0.29</td>
</tr>
</tbody>
</table>

Table 6: The $F$-measure of two temporal smoothing methods in Dataset 4

<table>
<thead>
<tr>
<th>Method</th>
<th>Obstacle</th>
<th>Road</th>
<th>Building</th>
<th>Tree</th>
<th>Sky</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRF</td>
<td>0.35</td>
<td>0.87</td>
<td>0.24</td>
<td>0.65</td>
<td>0.61</td>
<td>0.25</td>
</tr>
<tr>
<td>MFV</td>
<td>0.37</td>
<td>0.85</td>
<td>0.18</td>
<td>0.65</td>
<td>0.61</td>
<td>0.24</td>
</tr>
</tbody>
</table>
Figure 15: Sample results of scene passing of consecutive frames using Dataset 3 and Dataset 4. The panel (a) presents the result of day-time Dataset 3 while the panel (b) presents the result of night-time Dataset 4. In each panel, the first row shows the original camera image and the second row shows the image parsing result. The third row shows the result smoothed by the temporal fusion method. The red color illustrates the region of detected obstacles and other colors have similar meaning as in Fig. 3(d). After the temporal fusion, the results between consecutive frames are more cohesive.
Figure 16: The comparison of the Jump factor with or without using the temporal fusion. (a), (b), (c) and (d) show the results of Dataset 1 to Dataset 4, respectively. The Jump factor values are significantly reduced by the proposed MRF based temporal fusion method. The Jump factor is only used to qualitative evaluation of the proposed temporal fusion method.

7. Discussions

7.1. Selection of image classifier

An MLP (multilayer perceptron) classifier has been finally chosen to parse the image superpixels due to its lower computational cost than other classifiers like the kernel support vector machine (SVM) or the structured learning
Figure 17: Illustration of occlusion problem. (a) shows one camera image. (b) shows the image with the projected Velodyne points. The white bounding box localizes one cyclist. Due to the occlusion problem, wrong color information will be assigned to some 3D points inside the bounding box.

approaches like the conditional random field (CRF) [32]. According to our experiments, the linear SVM did not work in our case. The non-linear SVM
with RBF kernel could achieve comparable F-measure with MLP. However, the non-linear SVM runs much slower than the MLP as the number of learned support vectors amounts to 8000, while the MLP contains only 131 input nodes, 66 hidden nodes and 13 output nodes.

Our image parser only requires about 0.5 second to process one frame (400 × 300 pixels) in the common PC computer. The whole system can have a real time performance after the appropriate optimization. We suggest to speed up the feature extraction using GPU parallel computing technique as R. Benenson et al. [57]. Besides, both the optical flow and MRF can obtain the real time implementation by using the GPU technique [58][59]. Moreover, both the proposed fuzzy logic based sensor fusion method and the MRF based temporal fusion method are able to integrate the results of any classifiers.

7.2. Fusing two sensors at the feature level

In this paper, we have demonstrated promising results of the fuzzy logic fusion method by showing how it outperforms the results of individual sensors. Due to the sparseness of the pointcloud data of Velodyne scanner, we propose the geometry segmentation method to detect the obstacles and ground area from the scanner data. However, we do not think that our system alone is the ultimate answer to fuse Velodyne scanner and camera data. It is possible to extract discriminative features from the pointcloud sequence [43] and train a scene classifier by using both the image features and pointcloud features. Therefore, a natural future step is to combine the centralized and decentralized fusion methods for scene parsing.
7.3. Occlusion reasoning for the fusion of Vision and LIDAR

The fusion of data is correct if both sensors capture data from the same viewpoint. However, due to different viewpoints of both sensors, the occlusion occurs sometimes in the process of sensor fusion [15]. One example is shown in Fig. 17. This lets LIDAR obtain the 3D points of objects which are occluded in the camera view. The occlusion problem results in the wrong fused categorization of 3D points that are not visible to the camera. This problem can be solved by ordering the occluded 3D points or by using the pointcloud segmentation algorithm. Further details can be found in [15].

7.4. Integration of dynamic object tracking results

In the temporal fusion process, we represent each frame by a MRF model and integrate the results of multiple previous frames. Although this can smooth the scene parsing results, the object motion information is not incorporated. By considering the dynamic objects, we can leverage object detection techniques [60] and object tracking techniques [61] to obtain the category of corresponding pixels directly. Furthermore, the object track information is also helpful for occlusion reasoning and collision warning.

8. Conclusions

In this paper, we present a sensor fusion method for scene parsing using laser scanner and video camera. By employing fuzzy logic inference, our method can incorporate not only results of two sensors, but also the human experience and knowledge. To smooth parsing results of consecutive frames, we further propose a Markov random field based temporal fusion method.
The proposed approach has been evaluated with five datasets. Four of them are collected by our autonomous ground vehicle testbed in the rural and urban areas while the other one is a new public vision and laser dataset [54].

Our experiments underline the observation that the fused results are more reliable than those provided by individual sensors. For future work it would be interesting to explore the fusion with complementary sensors such as RADAR or stereo camera, which should allow for further improvements. The feature level fusion of laser scanner and video camera also deserved to be explored. Moreover, occlusion handling and dynamic object tracking are also important for robust environment perception.

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 proofread the paper. We greatly appreciate the insightful comments of three reviewers.

References


[20] F. Garcia, D. Olmeda, Hybrid fusion scheme for pedestrian detection
based on laser scanner and far infrared camera, in: Proceedings of the

[21] W. Tang, K. Z. Mao, L. O. Mak, G. W. Ng, Z. Sun, J. H. Ang, G. Lim,
Target classification using knowledge-based probabilistic model, in: Pro-
ceedings of the 14th International Conference on fusion (FUSION), 2011.

[22] B. K. Habtemariam, R. Thamarasa, T. Kirubarajan, D. Grimmett,
C. Wakayama, Multiple detection probabilistic data association filter
for multistatic target tracking, in: Proceedings of the 14th International
Conference on fusion (FUSION), 2011.

[23] S. Martin, Sequential bayesian inference models for multiple object clas-
sification, in: Proceedings of the 14th International Conference on fusion
(FUSION), 2011.

[24] R. Matthaei, H. Dyckmanns, Motion classification for cross traffic in
urban environments using laser and radar, in: Proceedings of the 14th
International Conference on fusion (FUSION), 2011.


[26] C. Galleguillos, A. Rabinovich, S. Belongie, Object categorization using
co-occurrence, location and appearance., in: CVPR, IEEE Computer
Society, 2008.


52


53


[58] G. Pascal, Z. Henning, G. Sven, B. Andres, W. Joachim, A highly efficient gpu implementation for variational optic flow based on the euler-
lagrange framework, in: ECCV Workshop on Computer Vision with GPUs (CVGPU), 2010.

