Real-time Lane Marker Detection Using Template Matching with RGB-D Camera

Cong Hoang Quach, Van Lien Tran, Duy Hung Nguyen, Viet Thang Nguyen,
Minh Trien Pham and Manh Duong Phung
VNU University of Engineering and Technology
Hanoi, Vietnam
Email: hoangqc@vnu.edu.vn

***Abstract*—This paper addresses the problem of lane detection which is fundamental for self-driving vehicles. Our approach exploits both colour and depth information recorded by a single RGB-D camera to better deal with negative factors such as lighting conditions and lane-like objects. In the approach, colour and depth images are first converted to a half-binary format and a 2D matrix of 3D points. They are then used as the inputs of template matching and geometric feature extraction processes to form a response map so that its values represent the probability of pixels being lane markers. To further improve the results, the template and lane surfaces are finally refined by principal component analysis and lane model fitting techniques. A number of experiments have been conducted on both synthetic and real datasets. The result shows that the proposed approach can effectively eliminate unwanted noise to accurately detect lane markers in various scenarios. Moreover, the processing speed of 20 frames per second under hardware configuration of a popular laptop computer allows the proposed algorithm to be implemented for real-time autonomous driving applications.**

I. Introduction

Studies on automated driving vehicles have received much research attention recently due to rapid advancements in sensing and processing technologies. The key for successful development of those systems is their perception capability, which basically includes two elements: road and lane perception and obstacle detection. It is certainly that road boundaries and lane markers are designed to be highly distinguishable. Those features however are deteriorated over time under influences of human activities and weather conditions. They together with the occurrence of various unpredictable objects on roads cause the lane detection a challenging problem. Studies in the literature deal with this problem by using either machine learning techniques or bottom-up features extraction.

In the first approach, data of lanes and roads is gathered by driving with additional sensors such as camera, lidar, GPS and inertial measurement unit (IMU) [2]. Depending on the technique used, the data can be directly fed to an unsupervised learning process or preprocessed to find the ground truth information before being used as inputs of a supervised learning process. In both cases, advantages of scene knowledge significantly improve the performance of lane and road detection. This approach however faces two main drawbacks. First, it requires large datasets of annotated training examples which are hard and costly to build. Second, it lacks efficient structures to represent the collected 3D data for training and online computation. As those data are usually gathered under large-scale scenes and from multiple cameras, current 3D data structures such as TSDF volumes [6], 3D point clouds [7], or OctNets [8] are highly memory-consuming for real-time processing.

In the bottom-up feature extraction approach, low-level features based on specific shapes and colours are employed to detect lane markers [1]. In [11], [12], gradients and histograms are used to extract edge and peak features. In [13], steerable filters are introduced to measure directional responses of images using convolution with three kernels. The template matching based on a birds-eye view transformed image are proposed in [14] to improve the robustness of lane detection. Compared with machine learning, the feature-based approach requires less computation and smaller datasets. The detection results however are greatly influenced by lighting conditions, intensity spikes and occluded objects [11], [12], [14].

On another note, both aforementioned approaches mainly rely on colour (RGB) images. The depth (D) information however has not been exploited. In lane marker detection, using both depth and colour information can dramatically increase the accuracy and robustness of the estimation tasks, e.g, obstacles like cars and pedestrians can be quickly detected and rejected by using depth information with a known ground model. The problem here is the misalignment between the colour and depth pixels which are recorded by different sensors such as RGB camera and lidar with heterogeneous resolutions and ranges. With recent advance in sensory technology, this problem can be handled by using RGB-D sensors such as Microsoft Kinect and Intel Realsense SR300 [3]–[5]. Those sensors can provide synchronised RGB-D streams at a high frame rate in both indoor and outdoor environments by using structured-light and time-of-flight technologies.

In this paper, we present a novel method for lane boundaries tracking using a single RGB-D camera. Low-level features of land markers are first extracted by using template matching with enhancements from geometric features. Dynamic thresholds are then applied to obtains the lane boundaries. Here, our contributions are threefold: (i) the formulation of a respond map for lane marker by using both colour and depth information; (ii) the proposal of a processing pipeline and refining feedback for RGB-D template matching; and (iii) the creation of 3D lane model estimation method by using high

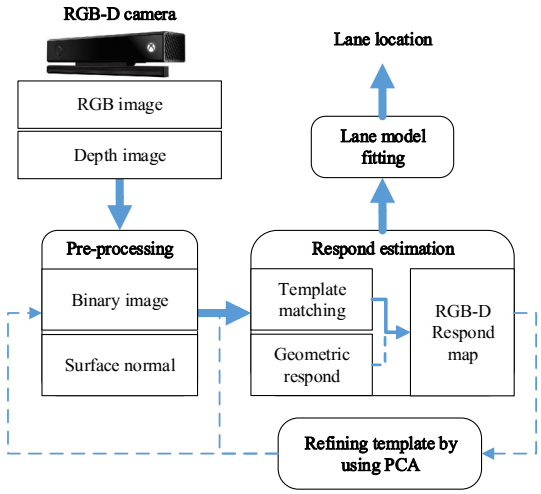


Fig. 1: Lane detection pipeline.

reliable lane marker points to deal with overwhelmed outlying data in scenes.

The remaining parts of the paper are structured as follows. Section II describes the methodology. Section III presents the experimental setup and results. The paper ends with conclusions and discussions presented in section IV.

II. Methodology

An overview of the proposed lane detector is shown in Fig.1. A single RGB-D camera attached to the vehicle is used to collect data of the environment. Recorded RGB images are then converted to binary images whereas depth ones are registered and transformed into 3D point clouds. Respond maps of lane markers are then built based on the combination of template matching and geometric feature outputs. The principal component analysis (PCA) technique is then used to refine the templates used. Finally, lane locations are obtained based on its model with a set of detected feature points. Details of each stage are described as follows.

A. Image pre-processing

In this stage, data from RGB channels are combined and converted to a eight-bit, grayscale image. This image is then converted to a half-binary format by using a threshold τ_c so that the intensity of a pixel is set to zero if its value is smaller than τ_c. At the same time, data from depth channel is transformed to a 2D matrix of 3D points in which the coordinate of each point, $p = (x, y, z)^T$, is determined by:

$$\begin{cases} x = \frac{i - c_x}{f_x} D(i,j) \\ y = \frac{j - c_y}{f_y} D(i,j) \\ z = D(i,j) \end{cases} \quad (1)$$

where $D(i,j)$ is the depth value at location (i,j) of the 2D matrix and (c_x, c_y) and (f_x, f_y) are the center and focal length of the camera, respectively. The Fast Approximate Least Squares

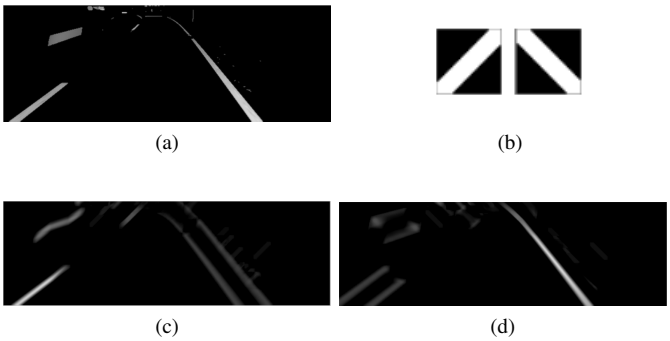


Fig. 2: Template matching process: (a) Haft-binary image; (b) Left and right templates; (c) Matching result with left template; (d) Matching result with right template.

(FALS) method is then employed to obtain 3D surface normals [15]. It includes three steps: identifying neighbours, estimating normal vectors based on those neighbours, and refining the direction of the obtained normal vectors. Specifically, a small rectangular window of size $k_N = w \times h$ around the point to be estimated is first determined. Least squares are then formulated to find the plane parameters that optimally fit the surface of that window. The optimisation process uses a loss function defined based on spherical coordinates to find the plane parameters in the local area as:

$$\hat{e} = \sum_{i=1}^{k} (v_i^T \hat{n} - r_i^{-1})^2, \quad (2)$$

where v_i is the unit vector, r_i is the range of point p_i in the window k_N, and $\hat{n}$ is the normal vector. $\hat{n}$ is computed by:

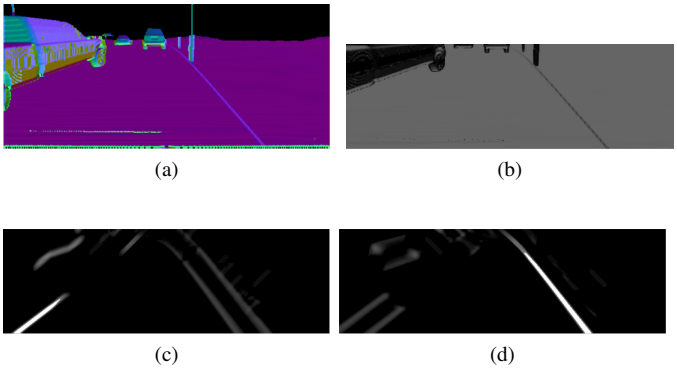
$$\hat{n} = \hat{M}^{-1} \hat{b}, \quad (3)$$

where $\hat{M} = \sum_{i=1}^{k} v_i v_i^T$ and $\hat{b} = \sum_{i=1}^{k} \frac{v_i}{r_i}$. As matrix $\hat{M}^{-1}$ only depends on parameters of the camera, it can be pre-computed to reduce the number of multiplications and additions required for computing surface normals.

B. Respond map computation

Given the standardised colour and depth images, our next step is to compute for each pixel a probability that it belongs to the lane marker. A combination of all probabilities forms a map called *respond map*. For this task, the evaluation is first carried out separately for the colour and depth images. A rule is then defined to combine them into a single map.

For colour images, we define two templates having shapes similar to the size and direction of lane markers in existing roads as shown in Fig.2b. Those templates are then used to extract features of lane markers from half-binary images by using normalized cross correlation (NCC). The matching result, M, as shown in Fig.2c and 2d is normalized to the range from 0 to 1 in which the higher value implies a higher probability of being lane markers.



(a) (b) (c) (d)

Fig. 3: Computation of respond maps: (a) 3D normal image; (b) G map; (c) respond map of left marker; (d) respond map of right marker.

On the other hand, the geometric feature map, G, is created from the depth image based on a predefined threshold, T_D, as:

$$G(i,j) = \begin{cases} \alpha(\vec{n}.\vec{O_y}) + \beta\frac{D(i,j)}{T_D} & \text{if } D(i,j) \leq T_D \\ \alpha(\vec{n}.\vec{O_y}) + \beta\frac{j}{imgHeight} & \text{otherwise} \end{cases} \tag{4}$$

Eq.4 can be illustrated as follows:

- If the depth value of a pixel is smaller than T_D, the corresponding value in G is the dot product between pixel normal $\vec{n}$ and the unit vector $\vec{O_y}$ of the camera view, which has a similar direction as the road's plane normal.
- If the depth value is greater than T_D or unknown due to noise, the corresponding value in G is set to a value between 0 and 1 depending on its horizontal location j in the 2D image.

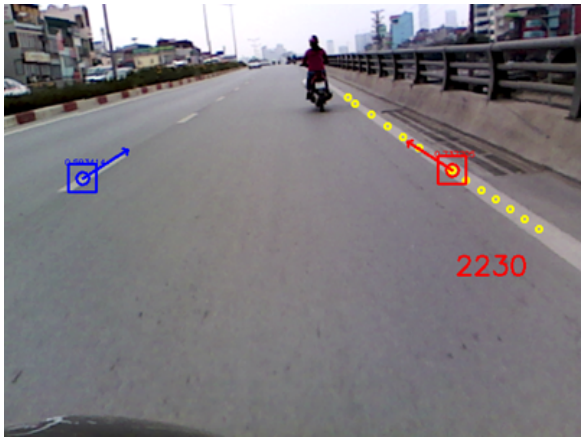
Based on the matching result M and the geometric feature map G, the respond map is established by the following equation:

$$R(i,j) = \begin{cases} M(i,j) & \text{if } M(i,j) < \tau_G \\ M(i,j) + G(i,j) & \text{otherwise} \end{cases} \tag{5}$$

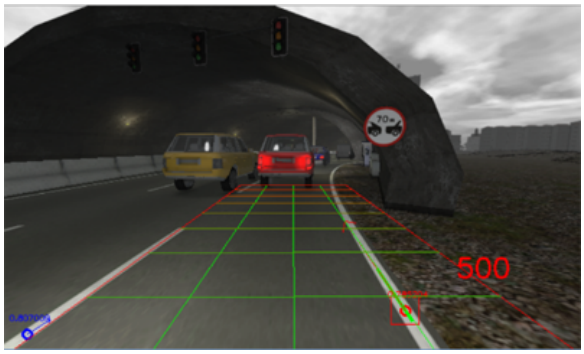
where τ_G is the threshold determined so that G only supports high-reliable colour features in the matching result M. Through this response map, both colour and depth information are exploited to evaluate the probability of a pixel belonging to lane markers.

C. Template enhancement

In lane marker detection, one important issue that need be tackled is the variance of markers with the scale and rotation of the camera. We deal with this problem by applying the principal component analysis (PCA) on the region of R corresponding to the highest probability of being lane markers. This region is selected by first choosing the pixel with the highest value (probability) and then expanding to its surrounding based on threshold P_{PCA}. The result is a set of points $P_r \subset R$ used as inputs for the PCA. As a result of PCA, the primary eigenvector output forms a new template



(a)



(b)

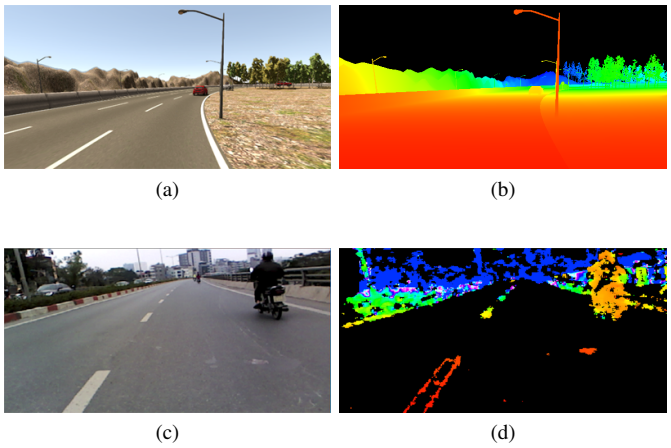
Fig. 4: Template enhancement: (a) Detection result with real datasets in which red and blue rectangles indicate two PCA-based analysis regions, arrows indicate lane direction, and yellow circles represent sliding box centers; (b) Detection result with synthetic datasets in which the grid represents the 3D plane model estimated by our algorithm.

angle θ that can be used to adjust the deflecting angle of the template in next frames for better detection.

On the other hand, the connected components of lane markers are selected by using a sliding box in the respond map M. The box has the same size as the matching templates and the centroid to be the highest positive point. To slide the box, its new origin ${O_b}^i$ is continuously updated from the centroid of the previous subset points P_r as:

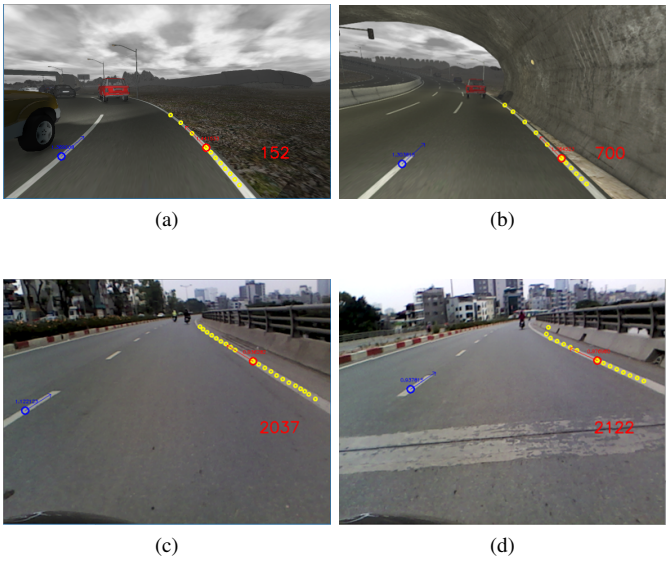
$${O_b}^{i+1} = \begin{cases} {O_b}^i + r(\cos\theta, \sin\theta) & \text{if } ({O_b}^i - {O_b}^{i+1})^2 \leq r^2 \\ \text{centroid of } P_{PCA} & \text{otherwise} \end{cases} \tag{6}$$

The stopping criteria include two cases: (i) the origin is out of the image area; or (ii) the set P_{PCA} is null. The main advantage of this method is that it does not require any change in viewpoint procedures as in [14] so that it is less sensitive to noise.



(a) (b) (c) (d)

Fig. 5: Differences between images captured (a)-(b) from synthetic data and (c)-(d) by a real RGB-D camera Intel RealSense R200.



(a) (b) (c) (d)

Fig. 6: Detection results with (a)-(b) Synthetic data and (c)-(d) Real data.

D. Lane model fitting

The plane model of lanes is defined by three points: two highest-value points in the respond map, v_{a1} and v_b , and the furthest centroid point of the right lane marker, v_{a2} . The plane normal of a lane is defined by the following equation:

$$\hat{n} = \overrightarrow{(v_b - v_{a1})} \times \overrightarrow{(v_{a2} - v_{a1})}$$

It is worth noting that the least square methods like RANSAC are not necessary to use here as most outliners have been removed by our RGB-D matching in previous steps as shown in Fig.3. As a result, our method has low computation cost and can overcome the problem relating to quantization errors of the depth map as described in [3].

III. Experiment

Experiments have been conducted with both synthetic and real datasets to evaluate the validity our method under different scenarios and weather conditions. The synthetic datasets are RGB-D images of highway scenario provided by [16]. The real datasets were recorded by two RGB-D cameras, Microsoft Kinect V2 and Intel Realsense R200. Figure 5 show the differences between images generated by synthetic data and a real RGB-D camera. As shown in Fig.6, the data was chosen so that it reflected different road conditions including:

- Summer daylight, cloudy and foggy weather.
- Lighting changes from overpasses.
- Solid-line lane markers.
- Segmented-line lane markers.
- Shadows from vehicles.

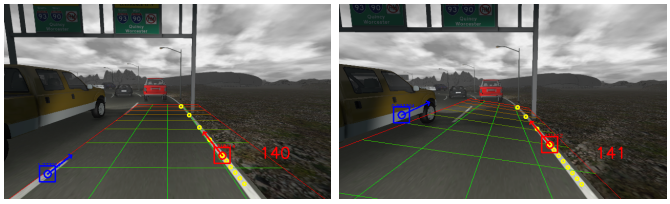
In all given conditions, we used templates of 32×32 pixels for the colour matching and 5×5 window size for normal estimation in FALS. For respond map computation, the depth threshold T_D was set to 20 m based on the range of sensory devices. We chose to use $\alpha = 0.4$, $\beta = 0.1$, and $\tau_G = 0.5$ to improve the RGB-D respond map. These parameters reflect the contribution of geometric information to the respond map. They are essential to remove the obstacles that cannot be handled by colour template matching. The size of the convolution kernel was 32×32 pixels and the minimum jump step r was 5. The condition to activate the template enhancement process is $P_{PCA} > 0.75$. It allows our system to work under moving viewpoint conditions. The camera parameters may affect template's shapes. However, our template size is small to show effects of view perspective.

TABLE I: PERFORMANCE OF 3D LANE MODEL FITTING IN SEVERAL SYNTHETIC AND REALISTIC DATASETS

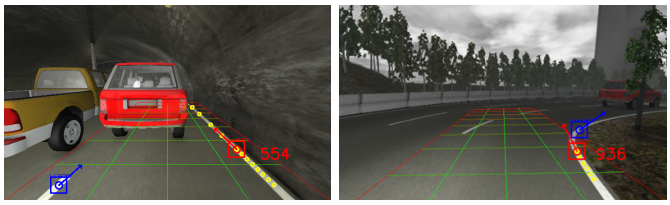
	Lane detection result		
Dataset	Frames	True positive	False positive
01-SUMMER	942	87%	0.6%
01-FOG	1098	84%	2%
06-FOG	857	80%	4%
Our Real Data	620	53%	7%

Figures 6 - 8 show the detection results. It can be seen that our method works well for both synthetic datasets and real data captured by the Kinect RGB-D camera. Table 1 shows the performance of our method on synthetic and realistic datasets. Changes in lighting conditions have a little effect on the results. However, wrong detections are sometimes happened, as illustrated in Fig.6d, when objects have similar shapes as lane markers. This problem can be improved by using negative filters.

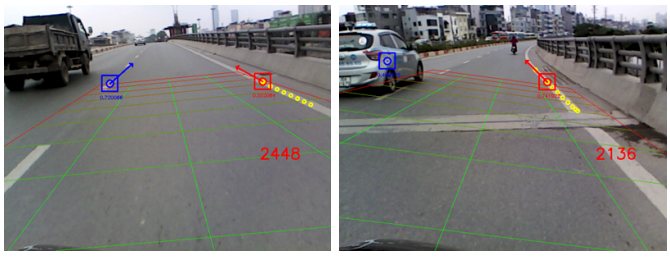
In experiments with real data recorded by the Realsense R200 RGB-D camera, as shown in Fig.5b and Fig.5d, low quality and sparse depth data reduce the quality of respond



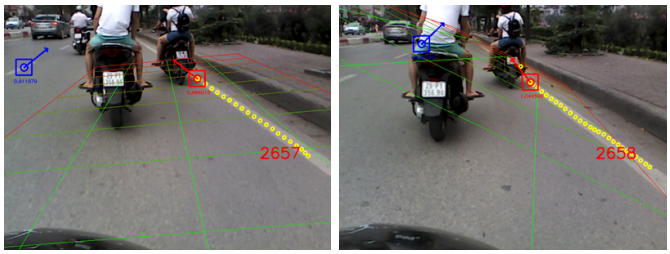
(a) (b)



(c) (d)

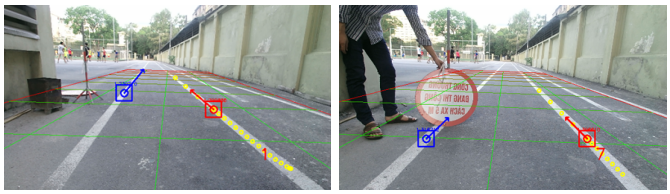


(e) (f)



(g) (h)

Fig. 7: Detection results with synthetic and real data: (a)-(g) True positive detection; and (b)-(h) False positive detection.



(a) (b)

Fig. 8: Detection results with real data recorded by Microsoft Kinect V2 RGB-D camera.

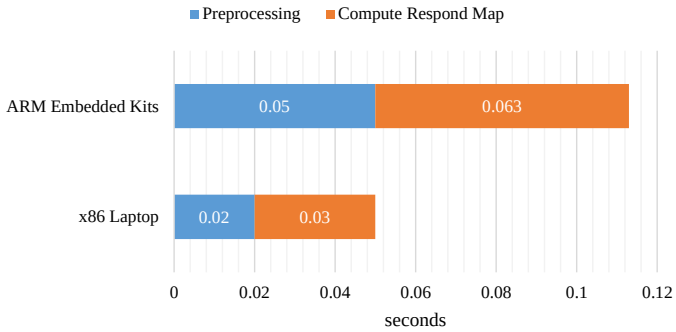


Fig. 9: Processing timelines of our algorithm running on laptop and embedded computers.

maps causing a number of frames to be skipped. We tried to tune parameters such as P_{PCA} and τ_G to improve the results, but the false positive rate was also increased. It can be concluded that depth information plays an important role in our detection algorithm. If high-quality devices like Kinect V2 is not available, approaches to interpolate sparse depth images should be considered.

In our implementation, the detection algorithm was written in C++ with OpenCV library and tested in two different hardware platforms: a laptop running Core i7 2.6 GHz CPU and an embedded computer named Jetson TX2 running Quad ARM A57/2 MB L2. Without using any computation optimisation, the program took around 0.05 seconds on the laptop and 0.113 seconds on the embedded computer to process a single frame. The algorithm is thus feasible for real-time detection. In a further evaluation, the computation time includes nearly 40% for preprocessing steps and 60% for computing the respond map (Fig.9). Other processing steps require so low computation cost that they do not influence the real-time performance. The cause is numeric operations on large-size matrices. This suggests future works to focus on matrix operation as well as taking advantage of parallel computing techniques such as CUDA with graphical processing units (GPU) for better processing performance.

IV. Conclusion

In this work, we have proposed a new approach to detect lane markers by using a single RGB-D camera. We have also shown that by utilising both colour and depth information in a single processing pipeline, the detection result can be greatly improved with the robustness against illumination changes and obstacle occurrence. In addition, the approach can achieve the real-time performance within a low computational hardware platform with low-cost cameras. It is thus suitable for implementing in various types of vehicle from cars to motorcycles. Our future work will focus on finding the rationale in false-positive scenarios to further improve the detection performance.

Acknowledgment

This work is supported by the grant QG.16.29 of Vietnam National University, Hanoi.

REFERENCES

[1] A. B. Hillel, R. Lerner, D. Levi, G. Raz, "Recent Progress in Road and Lane Detection: A Survey," Machine Vision and Applications, vol.25, no.3, pp. 727–745, 2014.

[2] J. Janai, F. Gney, A. Behl and A. Geiger, "Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art," in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[3] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," IEEE Transactions on Cybernetics, vol.43, no.5, pp. 1318–1334, 2013.

[4] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in Proceedings of the 2015 International Conference on Advanced Robotics (ICAR), 2015.

[5] T. H. Dinh, M. T. Pham, M. D. Phung, D. M. Nguyen, V. M. Hoang, Q. V. Tran, "Image Segmentation Based on Histogram of Depth and an Application in Driver Distraction Detection," In Proceedings of the 13th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV), 2014.

[6] B. Curless, M. Levoy. "A volumetric method for building complex models from range images," in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH), 1996

[7] M. D. Phung, C. H. Quach, D. T. Chu, N. Q. Nguyen, T. H. Dinh, Q. P. Ha, "Automatic Interpretation of Unordered Point Cloud Data for UAV Navigation in Construction," In Proceedings of the 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2016.

[8] G. Riegler, A. O. Ulusoy, A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[9] S. Hinterstoisser, C. Cagniart, S. Holzer, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal Templates for Real-Time Detection of Texture-Less Objects in Heavily Cluttered Scenes," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011.

[10] X. Ren, L. Bo, D. Fox, "RGB-(D) scene labeling: Features and algorithms," In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2759-2766, 2012.

[11] F. Samadzadegan, A. Sarafraz, M. Tabibi, "Automatic lane detection in image sequences for vision based navigation purposes," in Proceedings of the ISPRS Image Engineering and Vision Metrology, 2006.

[12] M. Nieto, L. Salgado, F. Jaureguizar, J. Arrospide, "Robust multiple lane road modeling based on perspective analysis," in Proceedings of the International Conference on Image Processing, pp. 2396–2399, 2008.

[13] J. McCall, M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," IEEE Transactions on Intelligent Transportation Systems, Vol. 7, no. 1, pp. 20–37, 2006.

[14] A. Borkar, M. Hayes, M. T. Smith, "A Template Matching and Ellipse Modeling Approach to Detecting Lane Markers," in Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS), 2010.

[15] H. Badino, D. Huber, Y. Park and T. Kanade, "Fast and Accurate Computation of Surface Normals from Range Images," in Proceedings of the 2011 International Conference on Advanced Robotics (ICAR), 2011.

[16] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.