

Rough Lane Marking Locating Based on Adaboost

Wuwen Jin^(✉) and Mingwu Ren

Nanjing University of Science and Technology, Nanjing, China
ww_jin@yeah.net, renmingwu@njjust.edu.cn

Abstract. Lane marking detection is a basic task of Driver Assistance Systems (DAS) and Autonomous Land Vehicle (ALV). In order to improve the accuracy of lane marking detection, we design a rough lane marking locating method based on predecessors' work. Considering the characteristic of lane markings, we extract Haar-like features of lane marking regions and train a strong cascade classifier by Adaboost Algorithm. The classifier is simple in principle and can fast locate the possible areas of lane markings accurately. Experimental results show that our method performs well.

Keywords: Lane marking locating · Adaboost algorithm · Haar-like features

1 Introduction

With the rapid development of science and technology, driving assistance systems (DAS) gradually enter people's daily life. It can not only raise an alarm when the vehicle is at risk, but also take driving permission to control the vehicle to void dangers, to prevent accidents. At the same time, as a typical representative of wheeled robot, autonomous land vehicle (ALV) senses environment information, and control the vehicle's steering and speed, relying on on-board camera or other sensors. It must ensure the vehicle run safety and reliably on the road without manual intervention.

Such intelligent transportation systems are developing towards to sense more complex environment and challenge more complicated tasks. The bottleneck of the development lies in the accurate perception of the environment [1]. This refers to two principal problems. One is road and lane marking detection, the other is obstacle detection. The lane marking is one of the key technologies for visual navigation. In this paper, we will focus on this problem.

Like human's driving, semi-autonomous or autonomous vehicles running on the road also rely on main perceptual cues such as road color and texture, road boundaries, lane markings and so on. Lane markings are identifications of road information and it can make vehicles driver orderly on the road. However, due to the complex and changeable of the traffic environment, and the high robust and real-time demands, fast and accurately detection of the lane markings is a difficulty and emphasis in the study of intelligent transportation [2].

The structure of this paper is as follows: in the next section we will introduce the current research situation on lane marking detection and put forward our method. In Sects. 3, 4, 5 we will introduce the work of our method particularly. Section 6 presents

the experiments on lane marking locating. Section 7 concludes the paper with a summary and directions for future work.

2 Related Work

The research on lane marking detection has been going on for decades. According to different detection methods, we can simply divide all these methods into three flavors, respectively based on models, based on regions and based on features [3]. Methods based on models firstly establish the models of roads. They obtain the lane markings after calculating the parameters related to these models. Common road models include line models and curve models, etc. These methods can satisfy all sorts of changeable environment of highways and the real time demand of the task. They can detect the lane markings fast and simply, but they are sensitive to noise [4]. Methods based on regions are only suitable for structured road. When the environment becomes complex, or road marking is blurred, the detection accuracy fell sharply [5]. Methods based on features using the color or texture features of the roads. They are simple but easy influenced by shadows, climate, illumination and so on. The detection accuracy is low [6]. So some scholars put forward the combination of other features such as geometric or shape features [7–9]. The fusion of such features indeed increases the accuracy, but increase the time cost as well. Considering Hough transform is not sensitive to the noise, many researchers apply it into the lane marking detection [10, 11]. But the computational complexity is high, so the applied range is limited [5].

In recent years, some scholars put forward the lane marking detection algorithms based on classifiers. They use neural network, Bayesian classification algorithm, k-Nearest Neighbor to establish classifiers for lane markings and obtain ideal results [12–14].

Based on predecessors' work, we proposed a new method to get the possible areas of lane marking by training a lane marking classifier. The process is presented in Fig. 1.

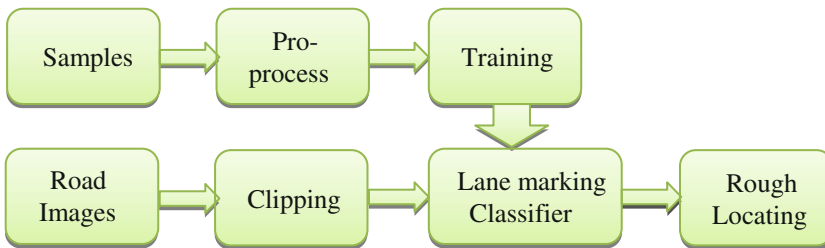


Fig. 1. Our method's processes

3 Image Pretreatment

Before training a classifier, we design the image pretreatment process to enhance the features of lane marking regions to make training and detecting stages faster and more accurate, including ROI clipping, image grayed and image enhancement.

3.1 ROI Clipping

Images captured by an on-board camera have broad vision, high resolution in complicated environment. As far as lane marking detection concerned, we do not have too much time and cost to focus on surroundings outside the road. Hence, before detection we need to clip the image to obtain the region of interest (ROI). After considering the structure characteristics of lane markings as well as speed and effect demands of our task, we clip the original image to get 4/5 part of regions near the bottom under the lane markings vanish point as our ROI. After that, we obtain a new road image in which the lane markings are approximate to straight lines and relatively clear shown as Fig. 2.

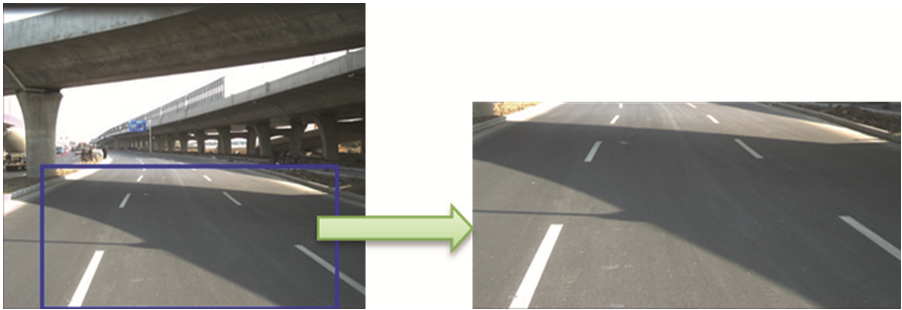


Fig. 2. ROI clipping

3.2 Graying

As we consider the structure features but not color features while detecting the lane markings, we convert the color images to grayscale images while considering the performance of the classifier and detection speed. Common methods to gray an image include component method, maximum method, average method and weighted average method. As we human eyes are highest sensitive to green and lowest to blue, we calculate the sum of three components by different weights. And then we get the average value. The proportions of R, G, B are 0.30, 0.59, 0.11. For an RGB image, the corresponding grayed image F can be expressed as:

$$F_{ij} = 0.30 R_{ij} + 0.59 G_{ij} + 0.11 B_{ij} \quad (1)$$

3.3 Image Enhancement

Considering the nature of the positive sample images, that is, there are only two types of objects we concerned, we process these samples after a gray-scale transformation. In order to enhance the lane markings regions, we need to strengthen the gray level which is greater than a certain value and suppress it when it is lower than the value. Therefore, it is a difficult problem to choose a suitable threshold. Obviously, it is not a good way to choose a fixed threshold for all samples.

As we know, lane markings in these positive samples are easy to distinguish from the road regions. OTUS may take this problem easily.

OTSU was proposed for image binarization by a Japan scholar named Otsu in 1979. Through the way of traversing all gray level, it selects a gray value which makes the variance between different classes greatest as the threshold of the whole image. For simple reasons, we choose this method to obtain an optimal threshold T for each positive sample image F_{ij} . We use T to stretch the positive sample by piecewise linear method and obtain a new grayed image G_{ij} . Figure 3 shows the graying and enhancement steps.

$$G_{ij} = \begin{cases} \min(1.2 * F_{ij}, 255), & \text{if } G_{ij} > T \\ 0.7 * F_{ij}, & \text{if } G_{ij} \leq T \end{cases} \quad (2)$$



Fig. 3. Graying and stretching process

4 Feature Extraction

We firstly need to choose some simple features for training if we want to obtain a lane marking classifier. These features should have to easily distinguish lane markings and non-lane markings. We use extended Haar features proposed by Rainer and Lienhart in [15]. They added several features with 45° rotating Angle based on original Haar features [16]. The extended features include four types: edge, linear, central, diagonal shown by Fig. 4.

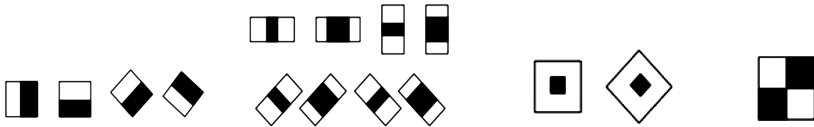


Fig. 4. Haar-like features

Haar eigenvalue is defined as the integral of the pixel gray value in the rectangular region. Black region has a negative weight and white region has a positive one. Actually, the eigenvalue reflects the information of gray scale changes. Some characteristics of lane marking regions can be described with Haar-like features. For example, color of regions on both sides of lane markings are deeper than lane marking itself.

5 Adaboost Algorithm

Adaboost is an adaptive boosting algorithm [17]. It combines multiple weak learning algorithms into a strong learning algorithm. This algorithm was proposed by Rreund and Robert in 1995 [18]. Its basic idea is using a large number of weak classifiers to establish a strong classifier with well classification capacity. Theoretically as long as classification capacity of each weak classifier is better than random guesses, the stronger one will obtain very low error rate close to zero.

At the beginning of training the first weak classifier H_0 , each sample in the training set S_0 is evenly distributed. We reduce its distribution probability if it is classified exactly and we increase the probability when it is classified falsely. Then we obtain a new training set S_1 which is mainly for the samples difficult to classify. After T iterations, we obtain T weak classifiers. If the accuracy of a classifier is high, its weight is high followed.

Adaboost classifier is based on cascade classification model. This model can be presented as Fig. 5.

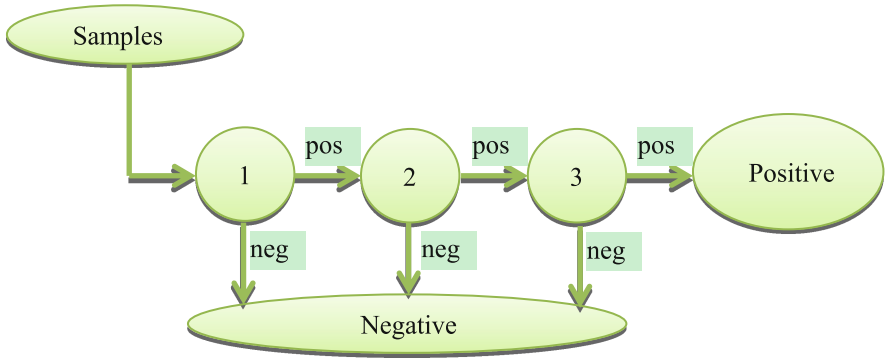


Fig. 5. Cascade classifier model

The procedure is presented as follows.

Step 1: For a given training set $S \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, while $y_i = 0$ means negative and $y_i = 1$ means positive, n is the total number of training samples.

Step 2: Initialize the weights:

$$w_{1,i} = \begin{cases} \frac{1}{2^m} \\ \frac{1}{2^l} \end{cases} \tag{3}$$

where m, n are numbers of negative samples and positive samples respectively.

Step 3: While $t = 1, 2, \dots, T$, t is times of iterations:

First, normalize the weights:

$$q_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,i}} \tag{4}$$

Then we train a weak classifier $h(x, f, p, \theta)$ for each feature. The weak classifier for j_{th} feature is presented as:

$$h_j(x) = \begin{cases} 1, & p_j f_j(x) < p_j \theta_j \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The weak classifier is decided by a threshold θ_j and an offset p_j . p_j decides the direction of the inequality and $p_j = \pm 1$.

We calculate weighted error rate of all weak classifiers respectively and take the lowest one. Then the corresponding classifier is the best one:

$$\epsilon_t = \min_{f,p,\theta} \sum_i q_i \cdot |h(x_i, f, p, \theta) - y_i| \tag{6}$$

Adjust the weights:

$$q_{t+1,i} = q_{t,i} \beta_t^{1-e_i} \tag{7}$$

where $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$, $e_i = 0$ if sample x_i is classified exactly, $e_i = 1$ if not.

Step 4: Combine T classifiers into a strong one:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}, \quad \alpha_t = \log \frac{1}{\beta_t} \tag{8}$$

6 Experiment

We collect large numbers of images by an on-board camera and establish a small lane marking database including common kinds of lane markings. Parts of the database are shown as follow (Fig. 6).

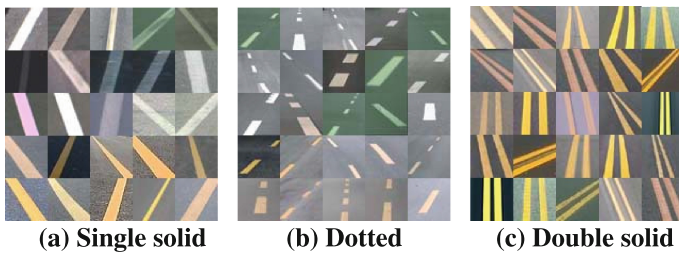


Fig. 6. Three different samples sets

We pre-process positive samples before training and normalize them to fixed size and then train a lane marking classifier. We use the classifier to detect possible regions of lane markings in some real images' ROIs. Based on enough experiments, we compare and analyze the results.

6.1 Training Time

Training process will last a long time, and the specific time is related to several factors.

Number of Samples. Time cost will increase as the number of training samples increases. We find that when the number of samples is between 100 and 200, the training process will finish in 1 h. But if the number adds up to 900–1000, the process will last as long as 6 h.

Layers of Classifier. It will cost longer time when the layers of cascade classifier become higher. Before training a classifier, we can set this training parameter manually. The experimental results show that when we set a small one, the capacity of the classifier is weak. But a larger one will lead to larger time cost. That is an annoying problem. In experiments we let the classifier to decide the layers itself. When the training process achieves a stable state or reaches the maximum false-alarm rate (FA) we set before training, it will end the training process.

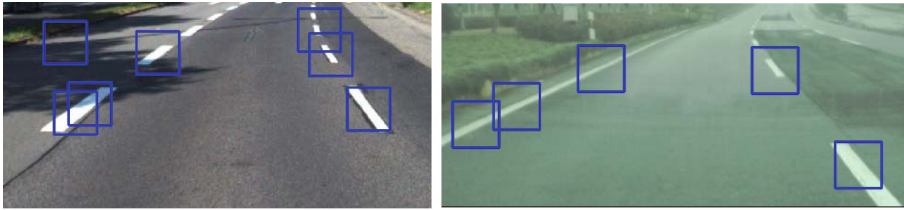
Experimental Platform. Training platform also determines the time consumption of training process. Our experiment platform is a laptop with Windows 7(64 bits), carrying Intel i5 CPU and 4 GB memory.

6.2 Detection Effect

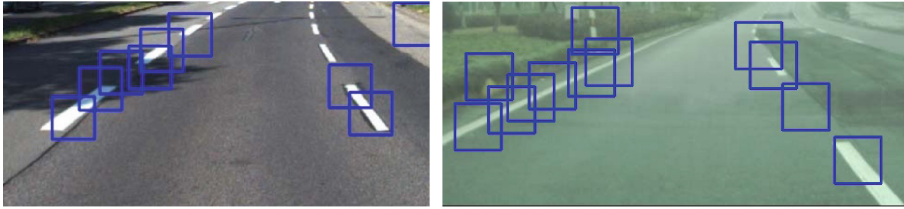
The stand or fall of the lane marking locating effect is directly influenced by the trained classifier's capacity. We design some contrast experiments among different combinations of sample sets, different pretreatment methods and different size normalization scales.

Different Combinations of Sample Sets. Choice of sample elements, different combinations of sample sets lead to different capacity of classifiers. We train different classifiers using different types of lane marking sets and their hybrid. The results are shown as Fig. 7.

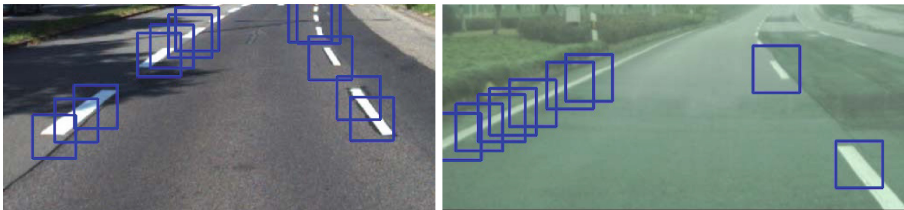
In Figs. 7(a), (b) and (c) are detected by classifiers trained by single solid lane marking set, mixture of single solid lane marking set and dotted lane marking set, mixture of single, double solid lane marking set and dotted lane marking set. We train other classifiers trained by dotted lane set and double lane set alone or mixed as well. It is proved that classifier trained by mixed sample set performs well than single sample set. From all results of all kinds of combinations, we find the classifier trained by three kinds of lane markings performs best. Hence, we chose this kind of mixture samples as our training set at last.



(a) Solid lane marking set



(b) Mixture of single solid lane marking set and dotted lane marking set



(c) Mixture of single, double solid lane and dotted lane marking set

Fig. 7. Locating results of classifiers trained by different combinations of samples sets

Different Pretreatment Methods. Different pretreatment methods for samples before training will lead to different capacity of classifiers. We process the samples through three different ways before training. The results are show as Fig. 8.

In Fig. 8(a), (b) and (c) are detected by classifiers trained by samples which are pro-processed by different ways including non-processing, binarization and linear grayscale stretching. Threshold in latter two methods is obtained by OTSU. From the results it is easy to find that if we do nothing for samples before training, although most lane marking regions are located, other regions without any lane markings are located as well, because these regions have similar features as lane marking regions due to road texture, shadows or illumination. Classifier trained by samples after binarization does better than the former but worse than the latter in which classifier is trained by stretched samples as the latter method reserves more information of the road. We now maintain the training set and pretreatment method to continue our experiment and study how the different size normalization scales affect our classifiers.

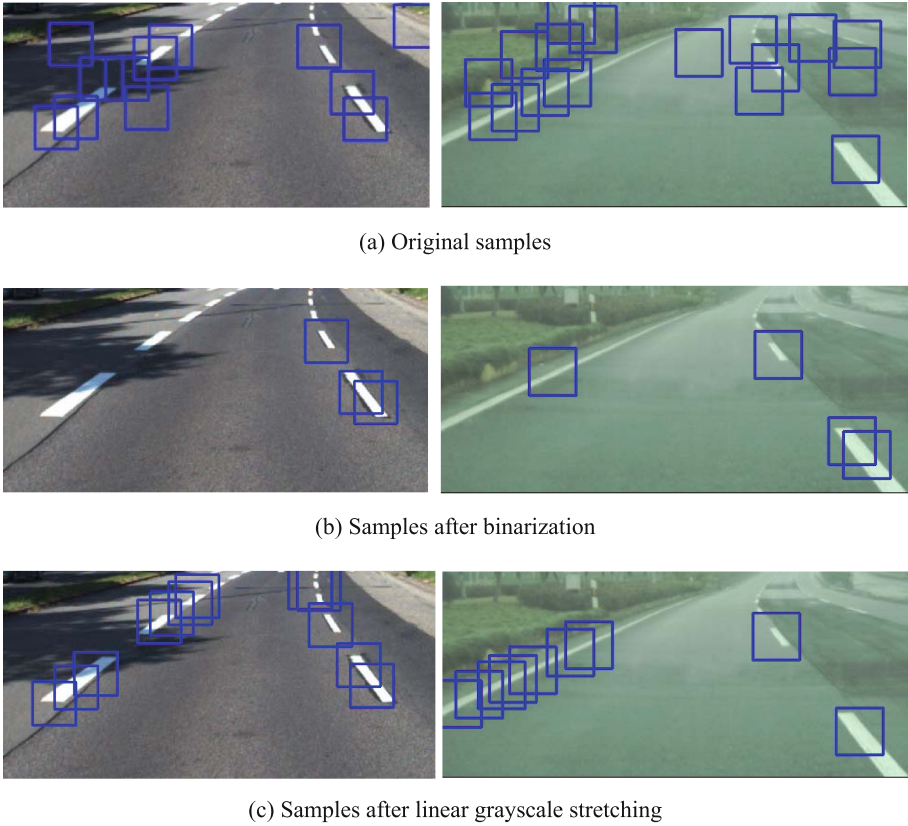


Fig. 8. Locating results of classifiers trained by different processed samples

Different Size Normalization Scales. Different size normalization scales lead to different results of lane marking detection. We normalize all samples to different sizes and then train the classifiers. The results are shown as Fig. 9.

In Fig. 9(a), (b) and (c) are detected by classifiers trained by samples with different sizes of 20×20 , 24×20 , 24×24 pixels respectively. We find the effects of different lane marking locating are similar because these three size scales of samples are all close to real lane marking regions' size.

We mix three types of samples together, totally 996 positive samples. After grayscale stretching, we normalize all these samples into 24×24 pixels. A 13 layers classifier is achieved at last. We select several images from on-board camera and opened database and then detect the possible regions contain lane markings through this classifier. Figure 10 shows the results.

As Fig. 10 shows, Fig. 10(a) is detection on normal road regions. Figure 10(b) shows lane markings interfered by shadows and vehicles. In Fig. 10(c), there are some instructional markings on roads. Figure 10(d) shows the results in bad weather.

We then test this strong classifier on several road data sets who are collected from daily life and network. The result is shown as Table 1.

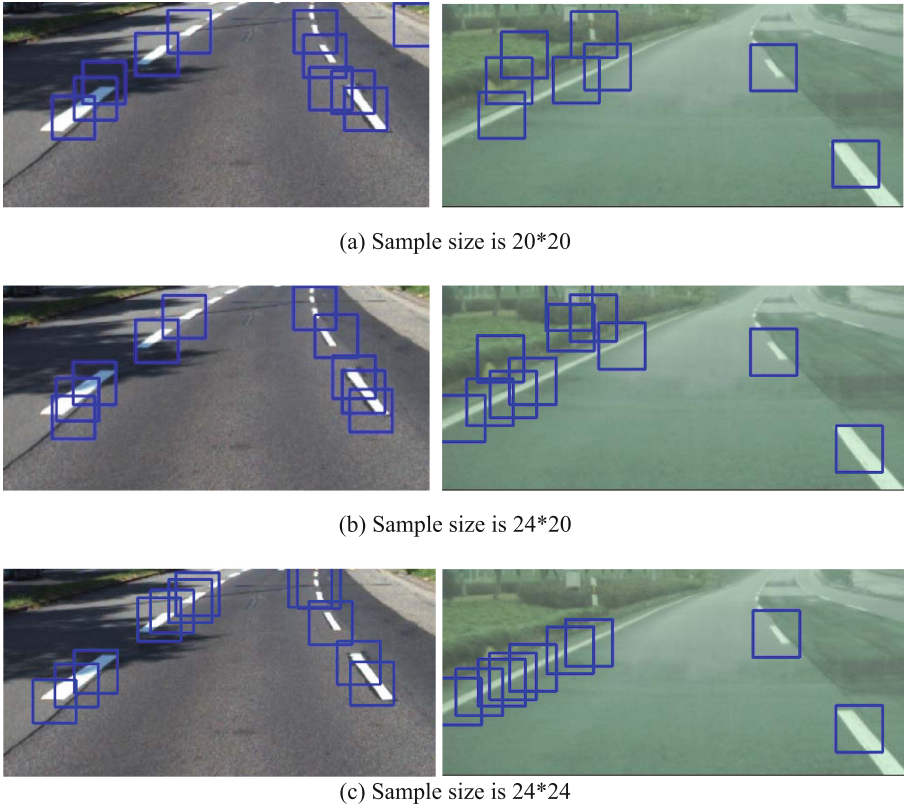
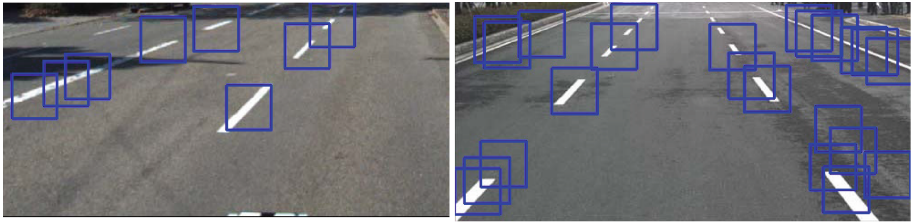


Fig. 9. Locating results of classifiers trained by samples with different size normalization scales

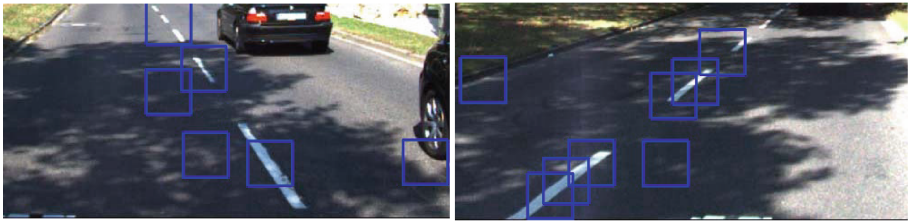
Set 1 and Set 2 are collected from the internet and road in images are in usual environment in the daytime. Set 3 is an opened road database from KITTI and it is a challenging task to detect the road markings because of the complicated road environment. We collect a large number of road data from on-board cameras. Then we select road images in bad weather and ban illumination condition and construct Set 4 and Set 5.

Table 1. The result on different road data sets

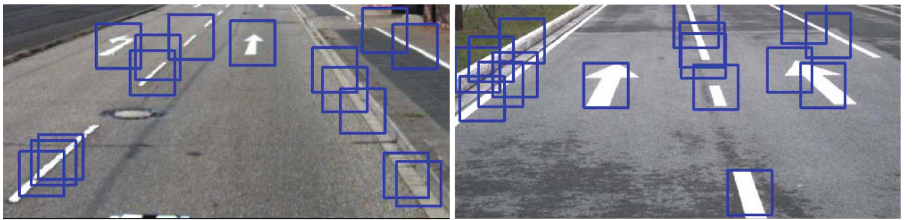
Set name	Image size	Number of images	Accuracy	Average time
Set 1	320*105	324	98.72 %	152.16 ms
Set 2	320*105	334	88.98 %	46.92 ms
Set 3	1392*262	908	71.04 %	1024.12 ms
Set 4	736 *200	1543	92.56 %	592.40 ms
Set 5	736 *200	1070	93.22 %	461.35 ms



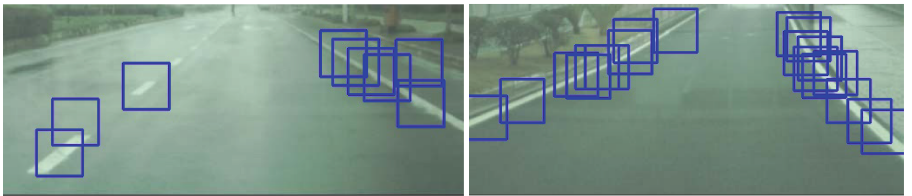
(a) Normal road regions



(b) Lane markings interfered by shadows and vehicles



(c) Instructional markings on roads



(d) Detections in bad weather

Fig. 10. Locating results in different environments

All the results show that our method can locate most possible lane marking regions in complicated surroundings although there are some noise regions and take a not ideal time.

7 Conclusion and Future Work

This paper designs a rough lane marking locating method based on Adaboost algorithm. Through extracting Haar-like features of lane marking regions, this method trains a

cascade classifier and locates the possible regions which contain lane markings. The experimental results show that this method is robust and accurate.

Our work will continue. First we will enrich the lane marking database and process the results which this paper has gained to get the precise lane markings. Second we will adjust several key details to short the detection time. At the same time, we will focus on lane marking tracking algorithm among frames. Our final aim is to establish a system which from image acquisition to exactly detecting and tracking the lane markings on the road.

References

1. Thorpe, C., Hebert, M., Kanade, T., Shafer, S.: Toward autonomous driving: the CMU Navlab. Part I: perception. *IEEE Expert* **6**, 31–42 (1991)
2. Li, S., Shen, H.: Lane marking detection in structured road based on monocular vision method (结构化道路中车道线的弹幕视觉检测方法). *Chin. J. Sci. Instrum.* **31**(2), 397–403 (2010)
3. Jung C.R., C., R’Kellber, C.R.: Lane following and lane departure using a linear–parabolic model. *Image Vis. Comput.* **23**, 1192–1202 (2005)
4. Wang, X., Wang, Y.: The lane marking detection algorithm based on linear hyperbolic model (基于线性双曲线模型的车道线检测算法). *J. Hangzhou Dianzi Univ.* **30**(6), 64–67 (2010)
5. Ju, Q., Ying, R.: A fast lane marking recognition based on machine vision (基于机器视觉的快速车道线识别). *Appl. Res. Comput.* **30**(5), 1544–1546 (2013)
6. Hu, X., Li, S., Wu, J., et al.: The lane marking detection algorithm based on feature colors (基于特征颜色的车道线检测算法). *Comput. Simul.* **28**(10), 344–348 (2011)
7. Shen, Y., Luo, W.: A new and fast lane marking recognition algorithm (一种新的车道线快速识别算法). *Appl. Res. Comput.* **28**(4), 1544–1546, 1550 (2011)
8. Wang, Y., Teoh, E.K., Shen, D.: Lane detection and tracking using B-Snake. *Image Vis. Comput.* **22**(4), 269–280 (2004)
9. Fan, C., Di, S., Hou, L., et al.: A research on lane marking recognition algorithm based on linear model (一种基于直线模型的车道线识别算法研究). *Appl. Res. Comput.* **29**(1), 326–328 (2012)
10. Zhao, Y., Wang, S., Chen, B.: Fast detection of lines on highway based on improved Hough transformation method (基于改进 Hough 变换的公路车道线快速检测算法). *J. China Agric. Univ.* **11**(3), 104–108 (2006)
11. Cai, A., Ren, M.: Robust method for vehicle lane marking extraction (鲁棒的车辆行道线提取方法). *Comput. Eng. Des.* **32**(12), 4164–4168 (2011)
12. Zhang, H., Lai, H., Tan, X.: Lane marking detection method based on wavelet analysis and minimum variance in the class (基于小波分析与类内最小方差的车道线检测). *Laser J.* **35**(3), 31–32 (2014)
13. Kim, Z.: Robust lane detection and tracking in challenging scenarios. *IEEE Trans. Intell. Transp. Syst.* **9**(1), 16–26 (2008)
14. Huang, C.L., Wang, C.J.: A GA-based feature selection and parameters optimization for support vector machines. *Expert Syst. Appl.* **31**(2), 231–240 (2011)
15. Lienhart, R., Maydt, J.: An Extended set of haar-like features for rapid object detection. In: *IEEE ICIP 2002*, vol. 1, pp. 900–903 (2002)
16. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vis.* **57**, 137–154 (2004)
17. Valicant, L.G.: A theory of the learnable. *Commun. ACM* **27**(11), 1134–1142 (1984)
18. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**, 119–139 (1997)