2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Traversable Region Detection with a Learning Framework*

Qingquan Zhang[1], Yong Liu[2], Yiyi Liao[1], Yue Wang[1]

Abstract— **In this paper, we present a novel learning framework for traversable region detection. Firstly, we construct features from the super-pixel level which can reduce the computational cost compared to pixel level. Multi-scale super-pixels are extracted to give consideration to both outline and detail information. Then we classify the multiple-scale super-pixels and merge the labels in pixel level. Meanwhile, we use weighted ELM as our classifier which can deal with the imbalanced class distribution since we only assume that a small region in front of robot is traversable at the beginning of learning. Finally, we employ the online learning process so that our framework can be adaptive to varied scenes. Experimental results on three different style of image sequences, i.e. shadow road, rain sequence and variational sequence, demonstrate the adaptability, stability and parameter insensitivity of our method to the varied scenes and complex illumination.**

I. Introduction

Vision-based traversable region detection is the key technique in driver assistance systems and autonomous navigation systems. It uses images from camera mounted on robots to extract traversable regions from current viewfield. Although many successful traversable region detection methods [1], [2], [3], [4], [5], [6] have been proposed, vision based traversable region detection is still a challenge due to the diversity of traversable regions, limited geometrical constraints[1], variation of surfaces and complex illumination.

Generally speaking, the traversable region detection methods can be divided into three categories. The first category [1], [7] tries to solve this problem by detecting vanishing point and then estimating the edges of the road. Further improved methods [5], [6], [8] extend the idea of vanishing point detection by combining other cues such as color, texture and detailed road shape models etc. to make the result more stable and adaptive. The drawback of these methods is obvious that they highly rely on the geometric constraints of the road, such as the vanishing points, parallel edges, boundaries of the roads etc. The methods [2], [3], [4] in the second category try to employ probabilistic models to represent the traversable possibility of the region. For example, Dahlkamp H.et al. [3] detect the nearby traversable region with lasers and represent the traversable region with a mixture of Gaussians (MOG)-model in RGB channels which is used to predicte the far region. Similarly, the RPDM (road probability density map) in pixel space [4] and Gaussian model in super-pixel space [2] are also used to represent the traversable regions. Although the above methods can use an optimal model, controlled by a set of parameters, to represent current traversable region, those parameters in the models are quite sensitive to the scenes. The methods [9], [10] in the third category try to transform the original image features to an illumination-invariant image feature space and then detect the traversable region in the new feature space. Although these methods can be quite sensitive if a single RGB channel is over-exposed or under-exposed [10]. Furthermore, the illumination-invariant mapping will change the image features from RGB channel to intensity channel which will also lose some discriminated information.

In this paper, we address on problems in the above methods and present a novel learning framework that can detect the traversable region with less constraint, and is robust against varied scenes and complex illumination conditions. In our framework, we only need a weak assumption that a small region in front of the robot is traversable, which can be satisfied in most of situations [2]. We then generate feature vectors from the images which are segmented into super-pixels with similar size, each feature vector is corresponding to a super-pixel. Then the traversable region detection can be regarded as a classification problem, namely labeling the super-pixels based on corresponding feature vectors. We segment the same image with multiple-scale super-pixels, the super-pixels in each scale are labeled independently, and the final traversable region is combined by voting of the multiple-scale labeled results. To provide our framework the capability of adapting varied scenes, we also introduce the online training process. Instead of obtaining the whole traversable region from the weak assumption by Growcut algorithm [2], our approach employs a weighed ELM [11] based learning method to expand the traversable region from the original assumption. By the above learning framework, our method can be stable to adaptive the scene and illumination changes, as it uses the prior traversable region knowledge to train the classifier without reset the control parameter manually.

The following sections are organized as follows. We show our basic idea in Section II and detailed learning framework in Section III. Experiments validate the contributions of our learning framework in Section IV and then we present conclusion in Section V.

*Rearsch supported by National Natural Science Foundation of China(61173123). Zhejiang Provincial Natural Science Foundation of China(LR13F030003).

[1]Qingquan Zhang, Yiyi Liao and Yue Wang are with Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027, China

[2]Yong Liu is with the State Key Laboratory of Industrial Control Technology and Institute of Cyber-Systems and Control (He is the corresponding author of this paper, e-mail: yongliu@iipc.zju.edu.cn)

[1]In our article, traversable region, e.g. the grassland in mountains, is not equal to the road, which means the traversable region may not contain the parallel edges in road and geometric characteristics in road detection.

978-1-4799-6923-4/15/$31.00 ©2015 IEEE

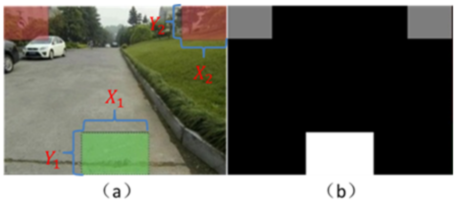


Fig. 1. Weak assumption in our approach. (a) The basic traversable and impassive regions assumption in our framework.where the bottom middle region (green rectangle with width X_1 and height Y_1) is always traversable, while the up left and right regions, the red rectangle with width X_2 and height Y_2, are impassable. (b) The mask where white means traversable, gray and black means impassable and unknown respectively.

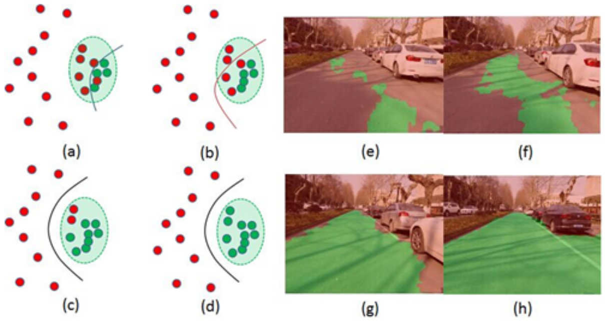


Fig. 2. The online training processing of the sequenced images. (e)-(h) are four sequenced images obtained by the robot. (a) - (d) show the labeling conditions and classification hyper-planes of the corresponding image sequence (e) - (h). As the misclassification cost functions for two categories are varied, the classification hyper-plane will move toward the lower risk area, that is the area of negative sub-regions,during the iterative training.

II. Basic Idea of Our Learning Framework

To loosen the constraints used in our method, we only assume a small region in front of the robot is traversable and two small regions on the top of image are impassable, shown in figure 1(a). And we can get a Fundamental Mask (*FM*), shown in figure 1(b). We also assume that there is a relatively clear boundary between the traversable region and the impassable region.

The traversable region detection can be regarded as a binary classification problem. We define the problem formally as follows:

Assume the image is segmented into n sub-regions (namely super-pixels in our approach), which can be denoted as an instance set F_n with labels:

$$(f_1, L_1), (f_2, L_2), \cdots, (f_i, L_i), \cdots, (f_n, L_n)$$

where $1 \leq i \leq n$, $f_i = [f_{i1}, f_{i2}, \cdots, f_{iD}]$ is the feature vector extracted from the ith image sub-region with length D, and $L_i \in \{-1, +1\}$ is the label corresponding to f_i as an indication of whether f_i belongs to traversable region (+1) or impassable region (-1). In the initialization, all the unknown sub-regions are set as impassable for safety. So the core task of our learning based framework is trying to get the separation boundary shown in figure 2(d) with those labeled sub-regions obtained from the assumption.

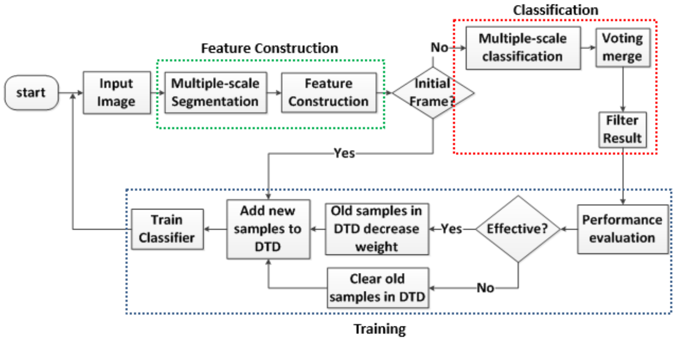


Fig. 3. Diagram of our learning framework for traversable region detection.

As the unknown sub-regions are all assigned with negative labels, there are much more negative instances than the positive instances at the beginning of learning. To solve this problem, we introduce the imbalanced learning approaches [11], [12], [13] into our learning framework, and set different misclassification cost for the two categories. Intuitively, the misclassification cost of the positive instances are set to be larger than that of the negative instances with respect to the quantity of these two categories.

The imbalanced learning approach is applied to our online training model for adaptive learning. We evaluate the classification results of current image and update the training dataset per frame. Then the classifier is retrained to predict the next frame. With the online training model, we can achieve an dynamically balanced classifier. If the number of negative instances is much larger than that of the positive instances, the misclassification cost of positive instances is set to be larger, which means the risk of misclassifying the negative instances is much less than that of the positive instances. So the separation boundary will move toward the negative instances during the online training process with sequenced images as shown in figure 2, and an increasing number of unknown sub-regions will be labeled as positive. On the other hand, if the class distribution of our training dataset is updated to be balanced, the separation boundary won't be influenced significantly by the quantity of different categories. In our learning framework, we use an empirical weighted scheme for each category to converge close to the ground-true separation boundary and establish the boundary in a dynamic balance during the online training.

III. Learning Framework of Detecting Traversable Region

Our learning framework for traversable region detection, shown in figure 3, can be general divided into three modules, i.e. feature construction, classification and training.

A. Feature Construction at Super-pixel Level

As mentioned in previous, our learning framework is established on the feature space of the image sub-regions, i.e. super-pixels. We segment each image with SLIC [14] due to its real-time performance. The feature vector that corresponds to each super-pixel is then constructed based

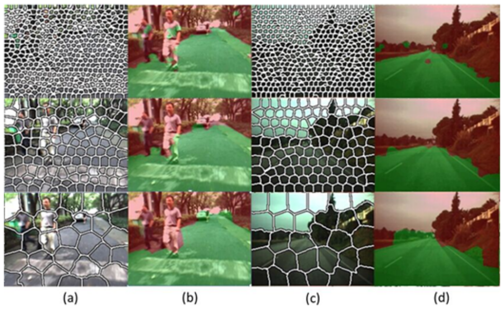


Fig. 4. Two samples of the three scales of segmentation. Column (a) and (c) are are the distributions of the super-pixels in varied scales, column (b) and (d) are the are the corresponding classification results, the green color area represents the traversable region.

on the color and texture information of the sub-region of the image in statistics. We use HSV histogram to construct the feature vector of color and the uniform LBP [15] descriptor to construct the feature vector of texture. To reduce the dimension of the feature vector, we divide the value of each channel in HSV color space into several regions and calculate the pixel-wise histogram upon those regions. Then the sub-vectors from three color channels and the sub-vector from the LBP descriptor are jointed and normalized to construct a uniformed feature vector for each super-pixel.

B. Classification in multiple-scale

In the learning framework, the labeled results of the super-pixels are highly depended on the size of the super-pixel. Small size of super-pixel may contain more detailed information of the scenes, but it also gives the risk of containing more noise. While large size of super-pixel is more stable and robust to noise, but lacks of details. So we segment the same image into multiple-scale super-pixels to combine both advantages of large size and small size of the super-pixels, shown in figure 4.

Assuming that the number of the super-pixels in the ith scale is denoted as M_i and each image is segmented into θ scales, where θ is an odd number. The number of super-pixels decreases when moving to higher layers as:

$$M_1 > M_2 > \cdots > M_\theta$$

Further define the amount ratio between two consecutive scales as a constant number K:

$$K = \frac{M_i}{M_{i+1}}$$

By setting K as a constant, our learning framework can control the size of the super-pixels in each scale and thus the multiple-scale super-pixels can provide complete descriptions by setting proper size gap between two consecutive scales.

After we segment the same image into θ layers with different scales, we construct the feature vector for each super-pixel in its corresponding layer. Let's denote the feature vector of the mth super-pixel in ith layer as f_i^m and denote the corresponding predicted label as L_i^m. It is to be noted that all the pixels belonging to the mth super-pixel in ith level are also labelled as L_i^m. After labeling all the super-pixels in every layers, we get θ binary labeled maps of the image in pixel level.

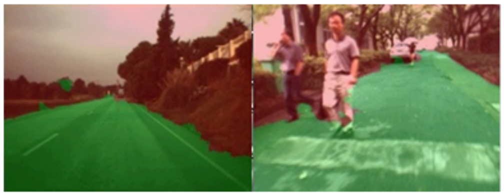


Fig. 5. The merged results for these two images shown in figure 4, which are much better than the results of single scale.

Then a voting strategy is applied to these θ binary labeled images to merge the results obtained by classifier of different scales. Let's denote the θ layer of labeled maps as $R = [r_1, r_2, r_3, \cdots, r_\theta]$, where $r_i(h \times w)$ is the ith labeled map of the image contains only +1 and -1. The voting strategy merges all the labels as:

$$R_{sum} = sign(\sum_{i=1}^{\theta} r_i)$$

where R_{sum} is the eventual results voting by all the levels' labels, shown in figure 5.

As the results voting by varied layers may still contain some noises or mislabeled areas, we introduce a post-filter processing by keeping the largest connected area and deleting isolated small areas.

C. Online Training with Dynamic Dataset

To address with the constantly changing scenes, we propose an online learning method which is based on the dynamic training dataset. In the context of online learning, we first classify the new coming image with current model, and then the training dataset is updated based on the results of classification, which is called dynamic training database (DTD) [4]. Finally we retrain our classifier with the updated training dataset.

Since the online learning updates the classifier once receiving a new image, we employ the weighted extreme learning machine (weighted ELM) [11] as our classification method for its high efficiency. Weighted ELM was proposed for dealing with imbalanced dataset which is simple and fast in implementation. It aims to minimize the training errors and maximize the marginal distance similar to SVM [16]. Given the feature vector f_i and label L_i, $i = 1, \cdots, N$, define an $N \times N$ diagonal weight matrix W with its elements w_{ii} corresponding to f_i. Then the cost function can be written as follow. Minimize:

$$L_{WELM} = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{N} w_{ii}\|\xi_i\|^2$$

Subject to:

$$h(f_i)\beta = L_i - \xi_i, i = 1, \cdots, N$$

where $h(f_i)$ is the randomly generalized hidden representation and ξ_i is the training error corresponding to f_i. β is the output weight which is calculated analytically using Moore-Penrose "generalized" inverse. C is a constant parameter which governs the ratio between the maximization of the marginal distance and the minimization of the training errors.

It can be seen that the imbalanced distribution can be well perceived if we set large weight w_{ii} to the feature vector f_i which belongs to the minority class. Denote n_{pos} and n_{neg} as the number of the samples belonging to traversable region and impassable region respectively. Thus the weights are given as follow:

$$\begin{array}{l} C_d = \frac{(n_{pos} - n_{neg})}{N} \in (-1, +1) \\ C_b = \lambda sign(C_d)|C_d|^{\sigma} \\ w_{ii} = w_0 - sign(L_i)(C_b - \phi) \end{array}$$

where C_d represents the divergence of quantity between the two categories, and C_b is the non-linear balance coefficient calculated based on C_d with constants λ and σ. We set $\sigma > 1$ so that w_{ii} varies gently when the class distribution is nearly balanced. ϕ is the threshold for prioritizing traversable region, which means we tend to label a feature as traversable when it lies near the separation boundary. w_0 is the constant value set artificially which represents the initial weight. Besides, the w_{ii} calculated as above is always non-negative with appropriate setting of λ and ϕ.

Once a new image is received, we employ the weighted ELM for classification. The key point of online learning is that soon after classifying the new image, we can get its true label. This is hardly realistic in the road detection problem. However, since we have the *FM*, we can still evaluate the previous classification results and update our DTD. Here the DTD is constructed similar to [4] that a training sample s_i is composed of three parts, i.e. feature vector f_i, label L_i and weight w_{ii}, which can be formulated as:

$$s_i = [f_i, L_i, w_{ii}]$$

With the *FM*, we assume the bottom middle region (white) is always traversable while the up left and right regions (gray) are always impassable. Then we can evaluate the accuracy according to the super-pixels which are contained in the *FM*. Denote the number of super-pixels of all layers contained in the *FM* as N_{FM}, while the number of true positive super-pixels contained in the white region is N_{WTP} and the number of true negative super-pixels contained in the gray regions is N_{GFN}. Then the accuracy is given as:

$$Acc = \frac{N_{WTP} + N_{GFN}}{N_{FM}}$$

If $Acc > 0.9$, we consider that existing DTD is suitable for current scene, the labeled super-pixels of current image are also added to the DTD. Otherwise, the existing DTD might be no more suitable for the changing scene, so we clear all of the samples in DTD and only use the results of current image to train a new classifier for next classification. Note that when adding new samples to DTD, super-pixels contained in the *FM* are always labeled consistent with the label of *FM* for training, while the rest super-pixels are labeled with the voted results.

For the new samples added to DTD, the weight W is also calculated corresponding to the ratio of samples belonging to the two classes. As for the pre-existing samples, we decrease their original weight when adding the new samples, which can be formulated as:

$$\hat{W} = W - \Delta W$$

where W denotes the original matrix and $\hat{W}$ denotes the new weight matrix, ΔW is the constant matrix for decreasing the weight. Therefore, the weights on training errors of pre-existing samples are lightened with the receiving of new frames. This is intuitive since the scene is constantly changing and the new samples are more effective to the future classification than the pre-existing samples. When updating lasts for several frames, the weights of earlier samples will be less than zero. Then these samples will be discarded, which means the number of the samples in DTD is limited to a constant. It combines the high efficiency and generalization ability at the same time.

IV. Experiments

In this section, we carry out experiments on three challenging datasets to evaluate the proposed method (Learning Framework for Traversable region Detection, abbreviated as *LFTD* in the following experiments), one is a subset of *rain sequence* dataset[2] with 425 frames, where 217 frames are labeled. The other two datasets, i.e. *shadow road* and *variational road* are captured by our autonomous robot system. Please visit our website for more details of our datasets[3]. Some samples of all datasets have been displayed in the first row of figure 7.

These datasets are typical scenes in daily life. The *rain sequence* dataset is a relatively simple scene with consistent appearances on roads, the vanishing point and relatively straight edges can always been detected in each frame. The *shadow road* dataset contains many shadows caused by the trees which will bring some troubles to the road detection, because the shadows may introduce confusions in the edge detection or model based road representation. *Variational* road is a challenging dataset containing image blur, barriers, varied illumination and significant texture-color changes on road surfaces. In our experiments, we choose these three datasets with varied styles to evaluate the robustness and adaption our method, as our method uses the same parameters on all those three varied datasets.

In our comparative experiments, we compare our *LFTD* with three state-of-art approaches. Growcut is a road detection algorithm proposed by Lu K et al [2]. VP is based on vanishing point and edges detection proposed by H. Kong et al [1]. Gaussian is a Gaussian mixture model(GMM) based road detection method proposed by H. Dahlkamp et al [3].

We use three pixel-wise quantitative metrics [17], [18], i.e. *FPR*(false positive rate), *FNR*(false negative rate) and

[2] https://rsu.forge.nicta.com.au/people/jalvarez/research bbdd.php.

[3] The databases can be download in our website http://www.nlict.zju.edu.cn/yliu/Visual.html

TABLE I

QUANTITATIVE PERFORMANCE METRICS OF FOUR METHODS IN THREE VARIED DATASETS

	Metrics	*Growcut*	*VP*	*Gaussian*	*LFTD*
Shadow Road	ErrorRate	6.6	12.3	4.66	**3.93**
	FPR	4.09	2.48	4.61	**0.85**
	FNR	9.88	25.73	**4.7**	7.87
Rain Sequence	ErrorRate	7.2	**4.14**	12.36	5.46
	FPR	3.29	7.81	**0.99**	1.64
	FNR	9.61	**1.96**	19.21	7.77
Variational Road	ErrorRate	11.66	12.14	15.21	**7.8**
	FPR	13.62	9.64	22.51	**2.1**
	FNR	14.51	16.62	**13.08**	14.92
Overall performance	ErrorRate	8.19	8.55	10.91	**5.63**
	FPR	6.21	6.77	7.64	**1.53**
	FNR	10.97	12.55	13.47	**9.67**

ErrorRate, to evaluate the accuracy of the detection. The metrics are calculated as follows:

$$FPR = \frac{N_{FP}}{N_P} \times 100\% \quad FNR = \frac{N_{FN}}{N_N} \times 100\%$$
$$ErrorRate = \frac{N_{FP}+N_{FN}}{N_P+N_N} \times 100\%$$

where N_{FP} is the amount of pixels being wrongly classified as traversable region; N_{FN} is the amount of pixels being wrongly classified as impassable region; N_P and N_N are the amounts of traversable and impassable pixels in ground-truth respectively.

These three metrics are calculated for each single frame, and then we average the metrics of all the image sequences in the same dataset to obtain the average performance on each dataset. Furthermore, we calculate the overall performance of each method by weighted average on the metrics on all the three datasets, here the weight of each dataset is positive corresponding to its image frames used in the experiments.

The experimental results of these four methods in three varied datasets are given in Table I, which includes the metrics on each dataset and the overall performances on those three datasets. Table I suggests that our method has superior performances on all the three datasets with the same parameter setting comparing with other methods, which validates the adaptability of our method. Although the *Growcut* also gives good results on these datasets, it needs to search optimal parameters for each specific dataset since it is sensitive to the variation.

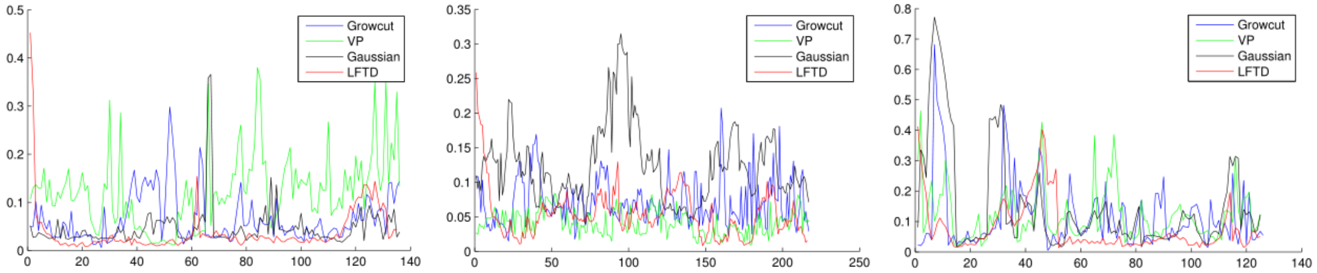
A further analysis on the experimental results in Table I, the results in *shadow road* dataset show our method outperforms the other three state-of art methods on most of the quantitative metrics, which indicates our method may be better adapted to the complicated lighting and shadow conditions. The *rain sequence* dataset has relatively stable geometric structure in the images so the *VP* performs best. However, the adaptivity of *VP* is limited by its relying on the strictly constrains, that's why the overall performance of *VP* is not so good. In addition, the *FPR* of our method is slightly larger than that of the *Gaussian* method on the *rain sequence*, the reason may be that the *Gaussian* method has over-optimized the metric of *FPR*, thus its metric on *FNR* is the worst in all the four methods and leads to the worst *ErrorRate* in *rain sequence* dataset. As for the *variational road* dataset, *LFTD* gives a relative large *FNR* compared to the *Gaussian* method which gives the best result. The reason is that the scenes of *variational road* dataset change frequently, once the change occurs, *LFTD* needs to clear the DTD and start training with the *FM* of current image. Thus many traversable regions will be labeled as impassable in the beginning of the training and the *FNR*s of those re-training frames will increase the average *FNR* of *LFTD*. This condition can also be reflected by figure 6(c), which contains several peaks in the curve of *LFTD*. On the other hand, the results also show that our method is relatively conservative. This may be good for autonomous navigation since it won't lead a robot to impassable regions when entering new environments.

In figure 6, we present the *ErrorRate* curves of consecutive frames in all datasets. All these figures show that *LFTD* outputs high error rates in the initial frames when our online training is mainly based on the *FM*. With the frames increasing, the error rates of *LFTD* will converge fast and become lowest among all the methods. In addition, figure 6(c) also shows that *LFTD* will output high error rate when the scene changes significantly. Similarly, the error rate converges fast and remains stable within the minimal range, which means our *LFTD* can provide an adaptive solution to the significantly changed scenes for autonomous navigation.

We also illustrate the detection results of all these four methods, shown in figure 7. For each dataset, we sample the frames at a regular interval. These frames of each dataset are listed from left to right in the order of timing. In figure 7(a), *LFTD* cannot detect all the road due to the shadow region at the first frame. With the online training framework, *LFTD* will learn to label the shadow regions correctly and achieve best performances in the rest frames. Besides, it is obvious to find that *VP* tends to regard the straight shadow, which is actually projected by the tree trunk, as the edges of road. Thus *VP* can't find the correct vanishing point and its performance is severely affected. It is also worth to mention that there is a dynamical pedestrian in the sixth frame of figure 7(c) and only our *LFTD* can detect the right traversable region in this frame. In this frame, the *Growcut* labels most part of the pedestrian as traversable region, which indicates that *Growcut* may perform bad on the variational scenes, especially when there are dynamical pedestrians. As for the other two methods, *VP* and *Gaussian*, there are also some false negative results labeled to the pedestrian. It further shows our *LFTD* is able to handle the situation with dynamic objects as it learns from recent frames. In conclusion, our *LFTD* can perform well with only one set of parameters. It will converge fast and remain stable when facing new environments, thus it is adaptive to the lighting, shadows and other variational road conditions.

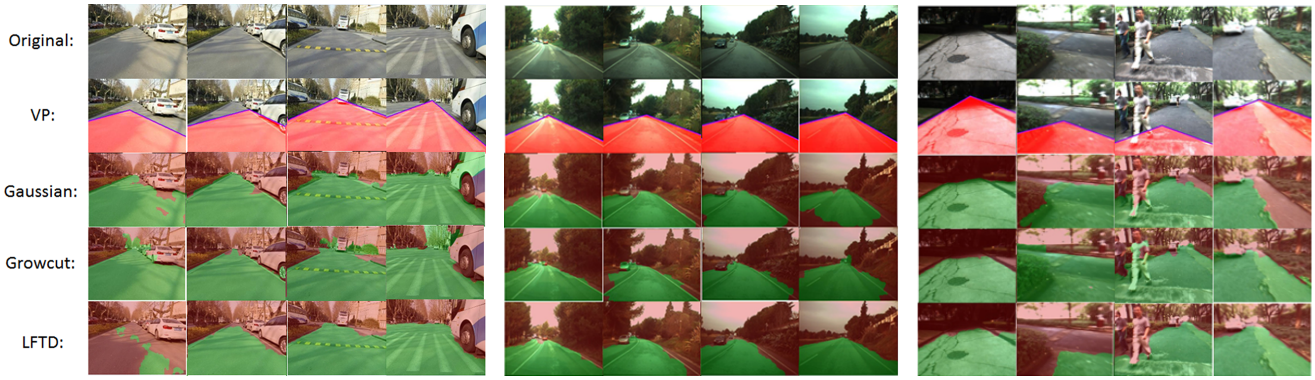
V. CONCLUSION

In this paper, we present a novel learning framework to detect travelable region, which is robust to varied challenged environments, such as varied illuminations, color and texture



(a) *Shadow road* (b) *Rain sequence* (c) *Variational road*

Fig. 6. The *ErrorRate* curves on all datasets. (a) is the *ErrorRate* curve on *shadow road*. (b) is the curve on *rain sequence* and (c) is the curve on *variational road*. The horizontal ordinate is the frame number and the ordinate is the average *ErrorRate*.



(a) *Shadow road* (b) *Rain sequence* (c) *Variational road*

Fig. 7. Consecutive frames sampled from all methods' results in all datasets. We select four frames in one dataset sequentially at regular intervals. In second row, traversable region is represented with red color setting by Kongs code, traversable regions in other rows are represented with green color.

changing in road surface, shadows, image blur and dynamic objects. The experimental results comparing with state-of-art methods show that our approach can provide adaptive and stable traversable region detecting performances.

References

[1] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *Image Processing, IEEE Transactions on*, vol. 19, no. 8, pp. 2211–2220, 2010.

[2] X. A. Keyu Lu, Jian Li and H. He, "A hierarchical approach for road detection," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 517–522.

[3] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain." in *Robotics: science and systems*. Philadelphia, 2006.

[4] S. Zhou and K. Iagnemma, "Self-supervised learning method for unstructured road detection using fuzzy support vector machines," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1183–1189.

[5] J. M. Alvarez, T. Gevers, and A. M. Lopez, "3d scene priors for road detection," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 57–64.

[6] W.-h. Zuo and T.-z. Yao, "Road model prediction based unstructured road detection," *Journal of Zhejiang University SCIENCE C*, vol. 14, no. 11, pp. 822–834, 2013.

[7] C. Siagian, C.-K. Chang, and L. Itti, "Mobile robot navigation system in outdoor pedestrian environment using vision-based road recognition," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 564–571.

[8] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot monocular vision navigation based on road region and boundary estimation," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1043–1050.

[9] J. M. Alvarez, A. López, and R. Baldrich, "Illuminant-invariant model-based road segmentation," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 1175–1180.

[10] B. Upcroft, C. McManus, W. Churchill, W. Maddern, and P. Newman, "Lighting invariant urban street classification," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014.

[11] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.

[12] H. He and E. A. Garcia, "Learning from imbalanced data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 9, pp. 1263–1284, 2009.

[13] R. Batuwita and V. Palade, "Fsvm-cil: fuzzy support vector machines for class imbalance learning," *Fuzzy Systems, IEEE Transactions on*, vol. 18, no. 3, pp. 558–571, 2010.

[14] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, 2012.

[15] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[16] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[17] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *International Conference on Intelligent Transportation Systems (ITSC)*, vol. 28, 2013, pp. 38–61.

[18] C. Guo, S. Mita, and D. McAllester, "Mrf-based road detection with unsupervised learning for autonomous driving in changing environments," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010, pp. 361–368.