2014 IEEE 17th International Conference on
Intelligent Transportation Systems (ITSC)
October 8-11, 2014. Qingdao, China

A Robust Road Segmentation Method Based on Graph Cut with Learnable Neighboring Link Weights

Jun Yuan, Shuming Tang, Fei Wang, Hong Zhang

Abstract— **Road region detection is a crucial functionality for road following in advanced driver assistance systems (ADAS). To address the problem of environment interference in road segmentation through a monocular vision approach, a novel graph-cut based method is proposed in this paper. The novelty of this proposal is that weights of neighboring links (n-links) in a s-t graph are estimated by Multilayer Perceptrons (MLPs) rather than calculating by the neighboring contrast simply in previous graph-cut based methods. Estimating n-link weights by MLPs reinforces the ability of graph-cut based road segmentation algorithms to tolerate the complex and changeable appearance of road surfaces. Additionally, the Gentle AdaBoost algorithm is integrated into the graph-cut framework to estimate the terminal link (t-link) weights in the s-t graph. Experiments are conducted to show the robustness and efficiency of the proposed method.**

I. Introduction

Road detection system is a pivotal component of environmental perception systems of ADAS. The relative position between road boundaries and the vehicle itself can be acquired through road detection for vehicle navigation, lateral control, lane departure warning, etc.

Most of road detection methods are based on computer vision approaches so far, owing to the richness of features, low cost and high resolution of camera data. In computer vision based road detection methods, the process of separating road region pixels from others in each road image is referred to road segmentation, which is in focus in this paper.

Road segmentation is essentially a challenging problem. On the one hand, road surfaces show changeable appearance, owing to different surface materials, weather conditions, illumination, artificial markings and so on. On the other hand, traffic scenes are complex owing to continuously changing surroundings and the presence of various vehicles or pedestrians. A large amount of research has been carried out in the last two decades for more accurate, more robust and faster road segmentation. The general idea of road segmentation is recognizing road pixels and background pixels by some distinguishing features, most often colors and texture cues. To mitigate impacts of some environmental factors, transformation of features [1] can be performed. In principle, the existing approaches to road segmentation can be categorized to two classes: unsupervised and supervised. Unsupervised methods collect pixels or regions with similarity into groups. Crisman *et al.* [2] adopted ISODATA (Iterative Self-Organizing Data Analysis Technique) clustering in their road following system. Cheng *et al.* [3] applied mean-shift algorithm for unstructured road segmentation. Malik *et al.* [4] used a self-organizing map to detect drivable road surfaces. Supervised methods regard road segmentation as a two-category classification problem and train classifier with labeled data to distinguish road regions. Zhou *et al.* [5] employed a Fuzzy SVM classifier for road segmentation and an online-learning scheme was combined. For optimal feature set is hard to be determined manually by prior-knowledge, Sha *et al.* [6] adopted boosting algorithm for its ability of feature selection. Kuhnl *et al.* [7] also adopted a boosting algorithm, i.e. GentleBoost, to extract road areas with a certain confidence. Basically, the methods mentioned above utilize only the local information, resulting in that the segmentation is sensitive to local interference, such as shadows and pavement cracks. Whereas graph-cut based methods [8, 9] integrate local and global information tightly, showing better robustness to local factors to a certain extent. In addition, applying graph cut to road segmentation is intuitive, for graph cut divides a road image into a road part (foreground) and a non-road part (background). However, previous graph-cut methods are prone to regard margins with high contrast caused by lane markings or strong shadows as road boundaries. It is attributed to that weights of neighboring links (n-links) in a s-t graph are calculated simply by neighboring contrast in previous graph-cut based methods.

In this paper, the graph-cut framework is adopted to extract road areas. It's different from previous methods that we use Multilayer Perceptrons (MLPs) to learn mapping relations from features of neighboring pixels to n-link weights and then estimate n-link weights by MLPs, which is the meaning of "learnable" in the title. This approach reinforces the ability of graph-cut based algorithms to tolerate complex and changeable appearance of road surfaces. Additionally, the Gentle AdaBoost algorithm is integrated into the graph-cut framework to estimate the terminal link (t-link) weights in a s-t graph. Because boosting methods have been proved to be very successful in feature selection [6] and the Gentle AdaBoost is often good at regression. Contrast experiments are carried out to show the performance of our method.

The remained structure of this paper is organized as follows: Section II analyzes the problem of previous graph-cut

* This work was supported in part by National Natural Science Foundation of China under Grants 91120008 and 61220002. The correspondence authors are Shuming Tang and Fei Wang.

Jun Yuan is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China (e-mail: yuanjun2012@ia.ac.cn).

Shuming Tang is with the High-Tech Innovation Engineering Center, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China (phone: 86-10-82544602, e-mail: shuming.tang@ia.ac.cn).

Fei Wang is with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University, Xi'an, 710049, China (e-mail: wfx@mail.xjtu.edu.cn).

Hong Zhang is with the School of Information Engineering, Minzu University of China, Beijing, 100081, China (e-mail: zhanghong751103@muc.edu.cn).

based methods firstly, and elaborates the proposed n-link and t-link weights estimate method next. In Section III, experiments are conducted and their results are described. Finally, the work is concluded in Section IV.

II. A Proposed Road Segmentation Method

A. Problem Formulation

Essentially, road segmentation can be considered as a pixel labeling problem. Each pixel p in an image is represented by a feature vector v_p. An image P can be represented by a set of feature vector denoted by $V = \{v_1, v_2, \cdots, v_N\}$, where N denotes the pixel number in an image. The target of road segmentation on a single road image is to find a corresponding label set $L = \{l_i \mid l_i \in \{0,1\}, i = 1,2,\cdots N\}$, i.e. assign a label to each pixel in the image. The labels of road region pixels are denoted by 1 while those of background pixels are set to 0.

In graph-cut based methods, a road image is modeling by a *s-t* graph [10, 11]. A s-t graph is composed of two different types of nodes and edges. Such two types of nodes refer to neighborhood nodes and terminal nodes, respectively. The former correspond to pixels in a road image, while the latter include a source node s and a sink node t. Edges those connect neighborhood nodes are *n-links*, and edges those link terminal nodes with neighborhood nodes are *t-links* [11]. To indicate link strength, a non-negative weight is assigned to each edge. A cut is a subset of edges which are to be cut off, and its cost is the sum of weights of edges in the cut. With respect to image segmentation, a cut divides nodes of a s-t graph into two disjoint subsets, which represent foreground and background. A min-cut is the one with the minimum cost, corresponding to an optimal segmentation of an image. Finding a min-cut can be formalized as a process of minimizing an energy function below [10, 11]:

$$E(L) = R(L) + \gamma B(L), \tag{1}$$

where $R(L)$ is the regional term which is with respect to the property of each individual node in a s-t graph, $B(L)$ is the smoothness term which is with regard to the relation between neighboring nodes, γ is a constant that indicates the relative importance of the two terms.

The regional term in (1) is defined as follows [10, 11]

$$R(L) = \sum_{p \in P} R_p(l_p), \tag{2}$$

where $R_p(l_p)$ is a t-link weight which can be regarded as a penalty for assigning label l_p to a neighborhood node p. $R_p(0)$ denotes the weight of the link to the source node while $R_p(1)$ corresponds to the weight of the link to the sink node. The regional term is supposed to be minimum only if all the pixels are labeled correctly. Generally, $R(l)$ can be calculated according to a probability distribution of data. For instance, Rother *et al.* [12] used Gaussian Mixture Models (GMMs) to model both foreground and background and defined $R(l)$ as a negative logarithm of the likelihood function of GMMs. But GMMs assume that data obey the Gaussian distribution, which is not realistic for changeable and complex traffic scenes. Thus, we apply the Gentle AdaBoost algorithm to estimate $R(l)$, i.e. the t-link weights, for its feature selection and regression ability.

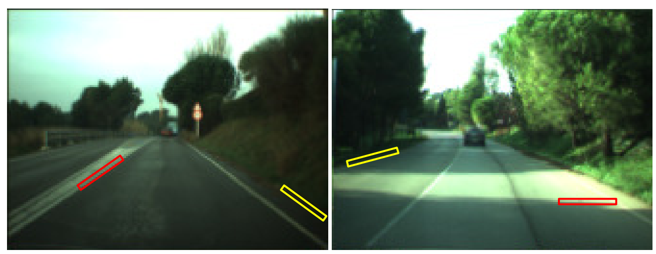
The boundary term in (1) is defined as follows [10, 11]:

$$B(L) = \sum_{(p,q)\in\mathcal{N}} \delta(l_p, l_q) \cdot B_{(p,q)}, \tag{3}$$

where $\mathcal{N}$ is a set composed of neighboring pixel pairs, $\delta(l_p, l_q)$ is a function taking value 0 and 1 for $l_p = l_q$ and $l_p \neq l_q$ respectively. $\delta(l_p, l_q)$ indicates that penalties only exists at the boundary between foreground and background. $B_{(p,q)}$ denotes a n-link weight, which represents the similarity between two neighboring pixels p and q. In previous graph-cut based segmentation methods, $B_{(p,q)}$ is usually defined in a format as follows [9, 11, 12]:

$$B_{(p,q)} \propto \exp[-\beta(v_p - v_q)^2], \tag{4}$$

where β is a constant which can be determined by the average contrast over an image [12]. Obviously, a corresponding penalty is high if different labels are assigned to two neighboring pixels with high similarity, whereas labeling two pixels with high contrast differently leads to a low penalty.



(a) (b)

Fig. 1 Changeable road surfaces. (a) is a traffic scene after raining, and (b) shows a road surface with serve shadows and strong illumination. Red rectangles mark sub-regions with strong contrast while yellow ones mark road boundaries with low contrast.

However, a road surface often shows considerable intra-regional diversity. Appearance of road surfaces is changeable and complex for the sake of changing illumination, shadows, weather conditions, different road surface materials, lane markings and so on. As shown in Fig. 1, pixels on both sides of lane marking edges or shadow margins generate strong contrast, while contrast on road boundaries is not as strong as hoped. Indeed, contrast at road boundaries is often much weaker than that of some sub-regions of road areas. As a result, the min-cut may deviate from the real road boundaries, when n-link weights are defined by (4) as in previous graph-cut methods.

In perspective of energy function designing for road segmentation, the energy function should be minimized only when the min-cut occurs at the road boundary. Thus, it's necessary to seek other approaches to define n-link weights instead of calculating by neighboring contrast simply as in (4). The key lies in how to find the mapping relations from features of neighboring pixels to n-link weights. An intuitive way is to learn such mapping relations automatically from data, i.e. labeled road images. In this paper, we adopt MLPs to model such mapping relations for their innate ability on data fitting.

B. Estimating n-link weights by MLPs

In our method, neighborhood nodes of a s-t graph are linked in a 8-connected way. Specifically, MLPs are adopted to estimate n-link weights of neighboring pixels which are adjacent either horizontally/vertically or diagonally. Actually, a graph composed of neighborhood nodes connected in the 8-connected way can be divided into numbers of basic units, each of which is with a single node and four edges, as shown in Fig. 2. Thus, we can obtain all n-link weights in a s-t graph by estimating weights of n-links in four direction as Fig. 2 (b) shown. Certainly, nodes on borders actually don't have such four edges, but we can regard absent edges as ones with zero weights. Four independent MLPs are constructed to model the mapping relations of n-links in such four directions respectively.

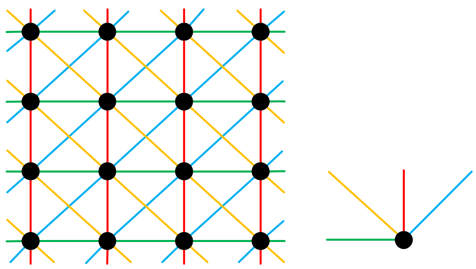


Fig. 2 (a) is a part of a graph whose nodes are linked in a 8-connected way, and (b) is a basic unit of (a).

Table I. Definition of input features of MLPs

Feature	Definition
Normalized color	$r = R/(R+G+B)$ $g = G/(R+G+B)$ $b = B/(R+G+B)$ $h = H\big/\sqrt{H^2+S^2}$ $s = S\big/\sqrt{H^2+S^2}$
Neighboring color difference	$cd = \sqrt{(C_c - C_n)^2}$, where $C = [r, g, b, h, s]$, indexes c and n refer to a current node and its neighboring node respectively.
Gradient	$Gr' = Gr / Grm$, $\theta' = \theta/(2\pi)$, where Gr and θ are gradient strength and orientation at a current node, Grm is the maximum gradient strength in an image.
Retinal position	$x' = x / Iw$, $y' = y / Ih$, where (x, y) is the coordinate of a current node, Iw and Ih are the width and height of an image respectively.

Each MLP is a four-layer feedforward network, composed of one input layer, two hidden layer and one output layer. Definitely, such four layers contain 10, 20, 20, 1 neuron(s) respectively. The sigmoid function is adopted as the activation function of each neuron. We take 10×1 dimensional feature vectors as input. As listed in Table I, the input features include color features, neighboring color difference, gradient features and retinal position. The output layer of each MLP is of an output scale between 0 and 1. The output value is regarded as a corresponding n-link weight. Apparently, the closer the output value is to 1, the greater the n-link weight is.

Road images are labeled in pixels to train the MLPs. But it's not necessary to take all pixels in an image to train. On the one hand, it will make the quantity of training data be huge and thereby lead to a large amount of computation. On the other hand, it's difficult to make the MLPs focus on the data which we are really interest in, such as shadows, light spots, lane markings, etc. Therefore, we need to apply a certain sampling method to pick pixels from training images. For each training image, positive samples are extracted from only road areas, and negative samples are pixels just on the road boundaries. Positive samples are not picked from non-road areas for we don't care n-link weights of nodes in non-road areas. Additionally, reducing the diversity of training data contributes to convergence of MLPs. We follow some probabilities to choose positive samples randomly from road areas with high contrast, strong grayscale intensity and other normal areas. All pixels just on road boundaries are picked as negative samples for such pixels are far less than pixels in road areas. The corresponding output values for such positive and negative samples are 1 and 0 respectively. Once training samples are ready, we adopted the RPROP (Resilient Backpropagation) algorithm [13] to train the MLPs. Owing to the discrepancy on quantities of positive and negative samples, additional weights should be put on negative samples to balance between hit-rate and false-alarm rate. Each weight coefficient is set as follows:

$$w(y) = \begin{cases} 1 \;, & if \;\; y = 1 \\ N_p / N_n \;, & if \;\; y = 0 \end{cases}, \tag{5}$$

where N_p and N_n denote numbers of positive and negative samples respectively, y is the label of a sample, i.e. $y = 1$ for positive samples and $y = 0$ for negative ones.

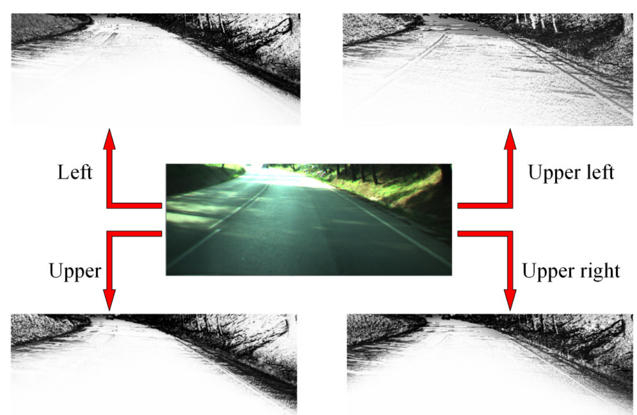


Fig. 3 Results of estimating n-link weights in the 4 directions by MLPs

After MLPs are trained, they are integrated into the graph-cut framework to estimate n-link weights:

$$B_{(p,q)} = MLP_d(v_{p,q}), \;\; d = dir(p,q)\,, \tag{6}$$

where $dir(p,q)$ denotes the direction of the edge linking pixel p and q , $v_{p,q}$ is the input feature vector. Fig. 3 shows results of n-link weights estimation of a road image with strong illumination and shadows. In Fig. 3, n-link weights in the 4 directions are converted to a range of 0 to 255, shown as grayscale images. It's shown that a majority of pixels in road areas are of high n-link weights with their respective neighbors while pixels near the road boundaries are of low n-link weights. Estimating n-link weights by MLPs shows resistance to illumination and shadows, which cause strong contrast in road areas and disturb previous methods estimating by contrast. Admittedly, a few pixels don't obtain consistent n-link weights as hoped. But it doesn't matter, because global energy minimization of graph cut can relieve impacts of some local interference to some extent.

C. Estimating t-link weights by Gentle Adaboost

The Gentle AdaBoost (GAB) algorithm [14] is applied to estimate t-link weights of each pixel. As a variant of AdaBoost, GAB owns excellent ability of feature selection. Furthermore, GAB is more numerically stable and robust to noise, for it puts less emphasis on outliers. Additionally, GAB is known as one of the best out of box supervised regression algorithm [15].

We adopt a GAB classifier composed of 500 weak classifiers. The weak classifiers in GAB are set as decision tree stumps, i.e. decision trees with only a single split node per tree. Input of the GAB classifier is a 13×1 dimensional feature vector, including color features (RGB values, mean and variance of RGB values in a 9×9 window whose center is a current pixel), gradient information (gradient strength and orientation) and retinal position. The GAB classifier outputs a weighted sum of votes rather than a class label. And the GAB is supposed to output positive (negative) values for positive (negative) samples. Labeled road images are used to train the GAB. Positive and negative samples are picked randomly from road regions and non-road regions respectively.

After being trained, it's integrated into the graph-cut framework to estimate t-link weights. The t-links include links from pixels to the source node s and links from pixels to the sink node t. Weights on such two types of links of a pixel p are defined as follows:

$$R_p(x_p, l_p) = \begin{cases} \dfrac{1}{1+\exp(-w \cdot x_p)}, l_p = 0 \\ \dfrac{1}{1+\exp(w \cdot x_p)}, \ l_p = 1 \end{cases}, \qquad (7)$$

where w is a positive constant and is set as 0.5 in the experiments, x_p denotes output of GAB on the pixel p, l_p is a label, $l_p = 0$ for the link to s, $l_p = 1$ for the link to t. If x_p is positive, assigning label 0 to p will suffer larger penalty than labeling 1 to it.

Once n-link and t-link weights are estimated through above methods, we can get a label of each pixel by minimizing the energy defined in (1):

$$\hat{L} = \arg\min_L E(L) . \qquad (8)$$

This can be solved by the max-flow/min-cut algorithm [10]. Finally, we can get optimal segmentation of road images.

III. Experimental Results

Experiments are conducted on a labeled dataset [16] to evaluate our proposed road segmentation method. The dataset includes two road image sequences captured under different weather conditions. One sequence (RainySet) contains 251 images captured after raining. Thus, road surfaces in this sequence are wet. Another sequence (SunnySet) is composed of 332 labeled images, which were acquired during a sunny day. Shadows and lighting variations are widely distributed in this sequence. The dataset are shown to be a well-balanced validation set in terms of diversity of road scenes [16]. Each image in the dataset is with a resolution of 640×480. Specifically, the region of interest is set as the lower 60% part of each road image. It's based on the fact that road regions are usually in the lower part of road images. In real applications, vanishing lines can be estimated to determine where ground surfaces are roughly located.

The dataset are separated into two parts, i.e. a training set and a testing set. To build the training set, about 20% images are picked randomly from each sequence. The remained images compose the testing set. The MLPs and GAB are trained on the training set, which contains images from both sequences. Then, contrast experiments are carried out on the testing set. Our method is compared with graph-cut based methods in which n-link weights are calculated by neighboring contrast as in (4) and t-link weights are estimated by Gaussian mixture models (GMMs), respectively. Three pixelwise measures, i.e. average precision P, average recall R and average F-measure F, are adopted to perform quantitative evaluations of road segmentation results. Such three measures are defined as follows:

$$P = \frac{\sum_{i=1}^{N_t} Num(A_i \cap M_i)}{\sum_{i=1}^{N_t} Num(A_i)}, R = \frac{\sum_{i=1}^{N_t} Num(A_i \cap M_i)}{\sum_{i=1}^{N_t} Num(M_i)}, \quad F = \frac{2PR}{P+R} \qquad (9)$$

where A_i denotes the detected road area of a given image i, M_i is the corresponding ground-truth mask, $Num(\bullet)$ denotes the number of elements in a set, N_t is the number of testing images. These measures of testing results are listed in Table II and Table III, which record performance on the RainySet and SunnySet respectively. The first column of both tables indicates the methods integrated into the graph-cut framework to estimate t-link weights and n-link weights, with the format of "t-link weights estimation method + n-link weights estimation method". Specially, CON represents the method in which n-link weights are calculated by neighboring contrast as in (4). With respect to GMMs, models are necessary to update with changes of road scenes. The strategy that we updates the models by a segmentation result of a previous frame is indexed by ① in the tables. At the very first frame, the models in strategy ① is initialized by the ground-truth. To compare performance of MLPs and CON more specifically, another strategy that models in GMMs are updated by the ground-truth of a previous frame is also adopted, denoted by ② in the tables. Either foreground or background is modeled by 5 Gaussian models in GMMs. It's found that taking more or less components doesn't bring obvious changes in road segmentation results, after cases of 3 to 30 components are tested.

With respect to both testing sets, the original method, i.e. GMMs + CON, shows high average precision P. But R is relatively low for there exists high contrast margins on road surfaces, which result in incomplete road segmentation. After CON is replaced by MLPs, R increase obviously although P decrease slightly. After MLPs and GAB are introduced, both P and R are improved. It's shown that estimating n-link weights by MLPs improves the average F-measure on both sequences. We attribute such performance improvement to that MLPs learn the mapping relations from features of neighboring pixels to n-link weights. Therefore, MLPs make the estimation of n-link weights more robust to environment factors such as shadows, lighting variations, and lane markings, etc. In other words, estimating n-link weights by MLPs reinforces the ability of graph-cut based methods to tolerate the complex and changeable appearance of road surfaces. In terms of t-link weights estimation, GBA shows similar or better performance than GMMs. It's owing to that GMMs assume data obey the Gaussian distribution, which doesn't match with the complex traffic scenes to some extent. Additionally, GBA has an aptitude for feature selection and data regression. Different model updating strategies of GMMs lead to the performance discrepancy between methods indexed by ① and those indexed by ②. It's noticed in Table III that GMMs + CON ① achieves a low performance, for such a method fails to overcome significant environment interference and capture the road surfaces in a fragment of the SunnySet.

Table II. Performance on the RainySet

Methods	***P*** (%)	***R*** (%)	***F*** (%)
GMMs + CON ①	99.65	90.93	95.09
GMMs + MLPs ①	99.10	98.68	98.89
GMMs + CON ②	99.17	94.62	96.84
GMMs + MLPs ②	99.00	98.91	98.96
GBA + CON	97.02	97.26	97.14
GBA + MLPs	98.58	98.76	98.67

Table III. Performance on the SunnySet

Methods	***P*** (%)	***R*** (%)	***F*** (%)
GMMs + CON ①	98.03	68.65	80.75
GMMs + MLPs ①	97.41	98.00	97.70
GMMs + CON ②	98.05	96.67	97.35
GMMs + MLPs ②	97.56	98.58	98.07
GBA + CON	98.65	98.62	98.63
GBA + MLPs	98.82	98.99	98.91

To illustrate the effectiveness qualitatively, some results of road segmentation on both sequences are shown in Fig. 4. Some interference factors, such as serve shadows, vehicles on the road, lane markings and so on, are distributed in the shown examples. By observing col. 2 and col. 4 (col. for column), it can be found easily that methods using CON for n-link weights estimation are sensitive to high and low contrast, which is pointed out in Part A of Section II. Whereas estimating n-link weights by MLPs are shown to be able to relieve such impact. Comparing col. 2 and col. 4 or col.3 and col. 5, we see that GBA outperforms GMMs on t-link weights estimation. Results in col. 5 are very approximate to the ground-truth.

Experiments above have shown the robustness and efficiency of our proposed method.

IV. Conclusions

In this paper, we propose a novel graph-cut based method on road segmentation. Firstly, we point out that previous graph-cut based methods are prone to regard regions with high contrast as road boundaries. To overcome such an obstacle, weights of n-links in a s-t graph are estimated by MLPs rather than calculating by the neighboring contrast simply in previous methods. Additionally, the GAB is integrated in the graph-cut framework to estimate t-link weights. Contrast experiments have shown the performance of our proposed method.

The weakness of our proposed method is time-consuming. The average time cost by a single frame is 1.8 s on a PC with a CPU of Intel Core i5. In the future, we will accelerate the process through parallel computing by GPU, for MLP and GAB are suitable for parallel process. Additionally, more evaluation experiments will be carried out next.

References

[1] J. M. Alvarez, A. M. Lopez and et al., "Illuminant invariant model-based road segmentation", in *Proc. IEEE Intelligent Vehicles Symposium*, 2008, pp. 1175-1180.

[2] J. Crisman and C. Thorpe, "UNSCARF, a color vision system for the detection of unstructured roads", in *Proc. International Conference on Robotics and Automation*, 1991, pp. 2496-2501.

[3] H. Cheng, C. Yu, and et al., "Environment classification and hierarchical lane detection for structured and unstructured roads", *IET Computer Vision*, vol. 4, iss. 1, pp. 37-49, 2010.

[4] U. A. Malik, S. U. Ahmed and F. Kunwar, "A self-organizing neural scheme for road detection in varied environments", in *Proc. International Joint Conference on Neural Networks*, California, 2011, pp. 3049-3054.

[5] S. Zhou, K. Iagnemma, "Self-supervised learning method for unstructured road detection using Fuzzy Support Vector Machines", in *Proc. International Conference on Intelligent Robots and Systems*, 2010, pp. 1183-1189.

[6] Y. Sha, G. Zhang, Y. Yang, "A road detection algorithm by boosting using feature combination", in *Proc. IEEE Intelligent Vehicles Symposium*, Beijing, 2007, pp. 364-368.

[7] T. Kuhnl, F. Kummert and J. Fritsch, "Monocular road segmentation using slow feature analysis", in *Proc. IEEE Intelligent Vehicles Symposium*, Badenm, 2011, pp. 800-806.

[8] J. Tsai, S. Huang and et al., "Road detection and classification in urban environments using Conditional Random Field Models", in *Proc. IEEE Intelligent Transportation Systems Conference*, 2006, pp. 963-967.

[9] G. Zhang, N. Zheng, and et al., "An efficient road detection method in noisy urban environment", in *Proc. Intelligent Vehicles Symposium*, 2009, pp. 556-561.

[10] Y. Boykov and V. Kolmogorov, "An experimental comparison of Min-Cut/Max-Flow algorithms for energy minimization in vision", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124-1137, Sep. 2004.

[11] Faliu Yi and Inkyu Moon, "Image segmentation: a survey of graph-cut methods", in *Proc. International Conference on Systems and Informatics*, 2012, pp. 1936-1941.

[12] G. Rother, V. Kolmogorov and A. Blake, "GrabCut - interactive foreground extraction using iterated graph cuts", *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 309-314.
[13] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm", *in Proc. IEEE International Conference on Neural Networks*, 1993, pp. 586-591.
[14] J. Friedman, T. Hastie and R. Tibshirani, "Additive logistic regression: a statistic view of Boosting", *The Annals of Statistics*, vol. 28, no. 2, pp. 337-407, 2000.
[15] T. A. Pham, M. Delalandre and S. Barrat, "A contour-based method for logo detection", in *Proc. International Conference on Document Analysis and Recogniton*, 2011, pp. 718-722.
[16] J. M. Alvarez and A. M. Lopez, "Road detection based on illuminant invariance", *IEEE Trans. Intelligent Transportation Systems*, vol. 12, no. 1, 2011, pp. 184-193.

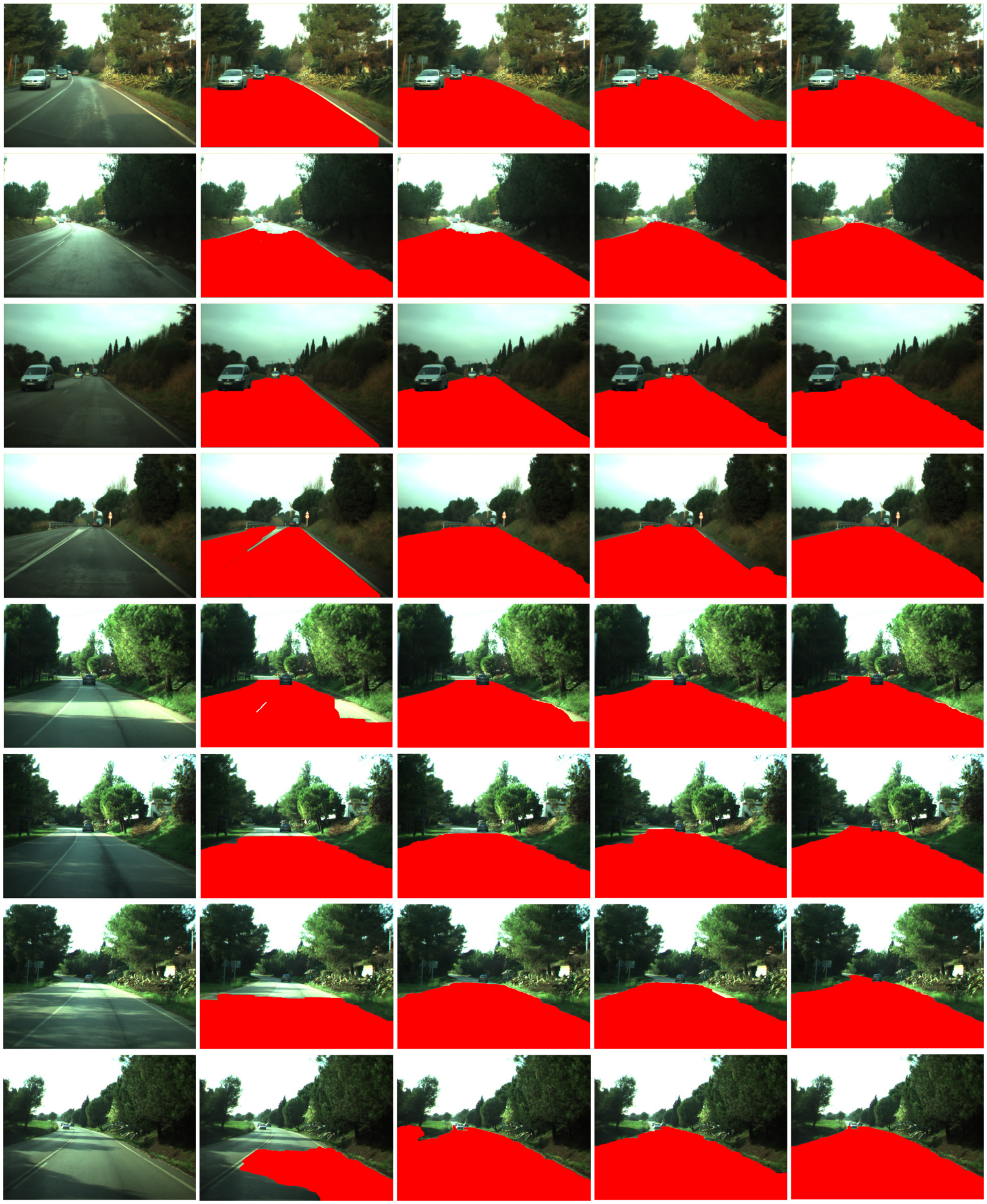


Fig. 4 Examples of road segmentation results. Detected road surfaces are marked by red regions. Row 1 to 4 display results on the RainySet, and Row 5-8 show those on the SunnySet. Column 1 list original images, and the remained columns are results of GMMs + CON ①, GMMs + MLPs ①, GBA + CON, GBA + MLPs, respectively.