Self-supervised Learning Method for Unstructured Road Detection using Fuzzy Support Vector Machines

Shengyan Zhou, Karl Iagnemma

***Abstract*— Road detection is a crucial problem in the application of autonomous vehicle and on-road mobile robot. Most of the recent methods only achieve reliable results in some particular well-arranged environments. In this paper, we describe a road detection algorithm for front-view monocular camera using road probabilistic distribution model (RPDM) and online learning method. The primary contribution of this paper is that the combination of dynamical RPDM and Fuzzy Support Vector Machines (FSVMs) makes the algorithm being capable of self-supervised learning and optimized learning from the inheritance of previous result. The secondary contribution of this paper is that the proposed algorithm uses road geometrical assumption to extract assumption based misclassified points and retrains itself online which makes it easier to find potential misclassified points. Those points take an important role in online retraining the classifier which makes the algorithm adaptive to environment changing.**

I. INTRODUCTION

Road detection is a crucial problem in the application of autonomous vehicle and on-road mobile robot. Many researchers have been studying road detection for several decades, there has been a dramatic development in this field [1]-[6] Since most of the researchers were focusing on road detection with lane marking, promising results were obtained especially in highways lane marking detection. But those methods are failed to apply to most of unstructured roads with inhomogeneous surfaces or without lane marks, such as rural roads and campus roads etc. Recently, there are several researchers has been working on unstructured-road detection. Most of them used machine learning method to solve this problem [7]-[9]. The author in [7] demonstrated a well-performed self-supervised road detection algorithm in rural road environment. It used one-dimensional template matching and the sum of squared differences (SSD) combined with optical flow to determine the most similar regions in front of vehicle. That method can just be applied to open road sections because an unexpected moving obstacle in the front view of ego vehicle would probably affect the template quality and matching result. The author in [9] used dynamical sampling windows to select training set and train a neural network classifier to detect road. But the window-based learning algorithm has a drawback that the training set derived from sampling windows can not well represents the real road/non-road classes feature space in the whole image. The classifier trained by that training set can not classify the real data accurately (See Fig.1). The pixels in sky area are misclassified as road class because the features in sky haven't been learnt in the sampling windows.

(a) (b)

Fig.1.Result of Window-based Learning Algorithm. (a).Sampling windows image. Red window contains positive training set while blue window contains negative training set. (b). Result of Classification. Red pixels are positive class while yellow pixels are negative class.

In this paper, we will introduce a novel machine learning based road detection algorithm in order to solve the above problems. What's more, The proposed algorithm is capable of not only online evaluating the quality of previous classification result, but also self-supervised online learning by automatically detecting the new training set which has more contribution in determining the hyperplane which makes the proposed algorithm adaptive to environment changing.

The primary contribution of this paper is, in our algorithm, instead of using sampling windows to select training set, we build the dynamical RPDM based on previous detection result which is used for weighting the training points to train a FSVMs classifier. The combination of RPDM and FSVMs solves the problem that learning from an inaccurate training set to get a relative accurate classifier. The secondary contribution is the proposed algorithm uses geometrical assumption which makes it easier to find possible misclassified points. This innovation answers the question that how we can find the possible misclassified points online without ground truth. The algorithm proposed in this paper can also be applied as a novel framework for self-supervised online learning method in the application of vision-based classification in the robotic field.

The rest of paper is organized as follows: In Section II, we provide brief overview of FSVMs which we use in our algorithm and demonstrate some comparison results between SVMs and FSVMs to demonstrate the advantage of FSVMs in some particular situation. The overview of proposed algorithm is presented in Section III. A detailed introduction

Manuscript received March.10, 2010. This work was supported by the National Natural Science Foundation of China under grant No. 90920304 and the Robotic Mobility Group in Massachusetts Institute of Technology, at Boston, USA.

Shengyan Zhou is now with the Robotic Mobility Group, Massachusetts Institute of Technology, on leave from the Intelligent Vehicle Research Center, Beijing Institute of Technology. (e-mail: zhousy@mit.edu)

Karl Iagnemma is with the Robotic Mobility Group, Massachusetts Institute of Technology (e-mail: kdi@mit.edu).

of individual part of proposed algorithm is given in Section IV. Then, the experimental results and conclusion are individually presented in Section V and Section VI.

II. FSVMs REVIEW

A Support Vector Machine (SVM) learns decision surface from the training points of two distinct classes. In many applications, the training points are labeled by human or some supervisor with high confidence. In another word, the training sets are all correct. In that case, we can use this training set to train a SVM classifier to get a well decision surface. However, there are still some applications which the training points are not fully corrected. But we have some confidences (weights) on those points. Even the training points are not one hundred percent correct. We still want to take those training points into decision surface determination because discarding those training points can be seen as information loss. What's more, we hope that the point with low weight has less contribution in determining the decision surface, vice versa. Fortunately, we found a novel SVMs algorithm called Fuzzy Support Vector Machines (FSVMs) can solve this problem. The FSVMs algorithm was first introduced by Chun-Fu Lin [10]. We will give a brief review about FSVMs and provide the comparison results between SVMs and FSVMs to demonstrate the differences of those two methods. The readers can refer to [10] for the detail about FSVMs.

A. FSVMs

Suppose we are given a set S of labeled training points with weights

$$(y_1, x_1, s_1), \ldots, (y_i, x_i, s_i)\,. \tag{1}$$

Each training point $x_i \in \Re^N$ belongs to either of two classes and is given a label $y_i \in \{-1,1\}$ and s_i with $i = 1,\ldots,l$, $\sigma \le s_i \le 1$ and sufficient small σ . Let $z = \varphi(x)$ denote the kernel which maps the x from original feature space $\Re^N$ to a feature space Z . The weight s_i is the confidence of the corresponding point x_i belonging to either of two classes. For example: $(y_i = 1, x_i, s_i = 0.80)$ means x_i 80% belongs to class one and 20% belongs to meaningless. With those training points, the optimal hyperplane problem is then regarded as the solution to

$$\text{minimize: } (w \cdot w)/2 + C\sum_{i=1}^{l} s_i \xi_i$$

$$\text{subject to } y_i(w \cdot z_i + b) \ge 1 - \xi_i, \quad i = 1,\ldots,l$$

$$\xi_i \ge 0, \quad i = 1,\ldots,l \tag{2}$$

where $w \in Z$ and $b \in \Re$, C is a constant. It is noted that smaller s_i reduces the effect of the parameter ξ_i in problem (2) such that the corresponding point x_i has less contribution in the minimization in (2).

Like SVMs, searching the optimal hyperplane in (2) is a QP problem, which can be solved by constructing a Lagrangian and transformed into the dual

$$\text{maximize } W(a) = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j K(x_i, x_j)$$

$$\text{subject to} \sum_{i=1}^{l} y_i a_i = 0 \quad 0 \le a_i \le s_i C\,,\ i = 1,\ldots,l \tag{3}$$

and the Kuhn-Tucker conditions are defined as

$$\overline{a}_i(y_i(\overline{w} \cdot z_i + \overline{b}) - 1 + \overline{\xi}_i) = 0\,, i = 1,\ldots,l \tag{4}$$

$$(s_i C - \overline{a}_i)\overline{\xi}_i = 0\,, i = 1,\ldots,l \tag{5}$$

The point x_i with the corresponding $\overline{a}_i > 0$ is called a support vector. There are two types of support vectors. The one with corresponding $0 < a_i < s_i C$ lies on the margin of the hyperplane. The one with corresponding $a_i = s_i C$ is misclassified. An important difference between SVM and FSVM is that each point has individual constraint in FSVM in (3). The only free parameter C in SVM controls the tradeoff between the maximization of margin and the amount of misclassifications. A larger C makes the training less misclassifications and narrower margin. The decrease of C makes SVM ignore more training points and get wider margin. In FSVM, if we fix the value of C. With different value of s_i , we can control the tradeoff of the respective training point x_i . A smaller value of s_i makes the corresponding point x_i less important in the training.

B. Comparison Results of SVMs and FSVMs

Fig.2 shows an extremely bad training set. We hope that the decision hyperplane is x=y. So we assume that all the

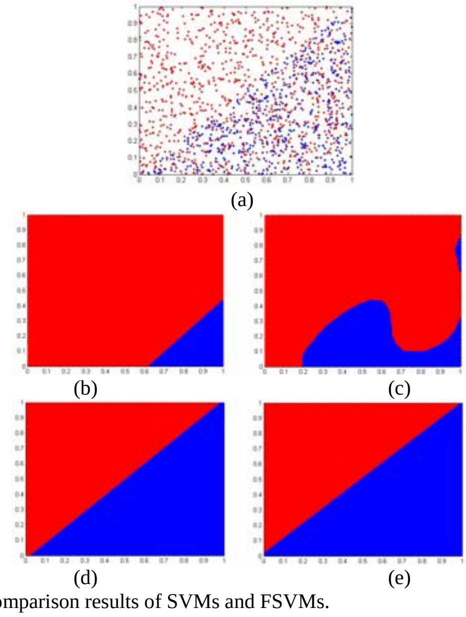


Fig.2. Comparison results of SVMs and FSVMs.
(a). Poor Training Set. Red points are positive training data. Blue points are negative training data. (b) and (c) are decision hyperplanes derived from SVMs with linear kernel and RBF kernel. (d) and (e) are decision hyperplanes derived from FSVMs with linear kernel and RBF kernel.

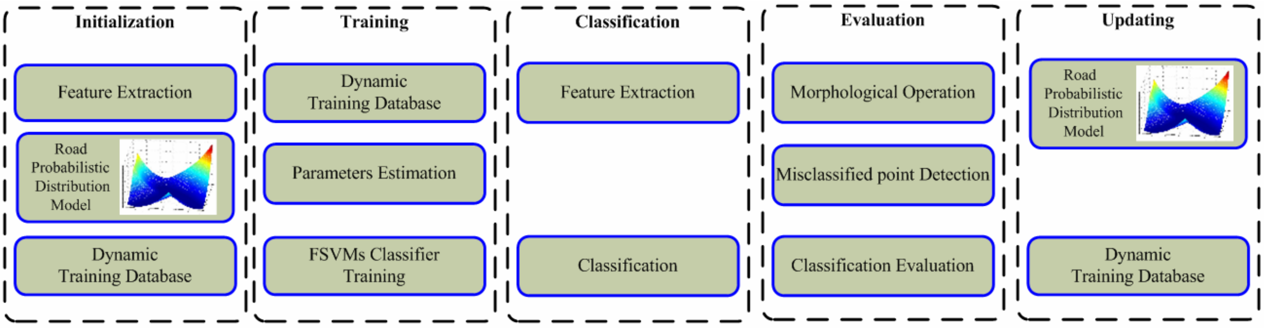


Fig.3 Algorithm Components

positive training points lay in x<y are mislabeled. The only free parameter C in SVMs can't control the contributions of each point. From the results in Fig.2 (b) and (c), we could see that both decision hyperplanes are affected by the mislabeled positive training points. Instead of SVMs, we weight each mislabeled positive training point with smaller value (0.01 in this example). Those points become less important in determining the hyperplane. The reasonable results is shown in Fig.2 (d) and (e).

In proposed algorithm, the training set can be automatically found out from online. However, it is probably that some training points are mislabeled. In traditional SVM, those misclassified would disturb decision hyperplane in training SVMs classifier. Given the RPDM built in our algorithm, the points with high probability mislabeled will be weighted with low weights. In that case, the relative accurate classifier can still be trained using FSVMs training method. In following sections, we provide a detailed description of this innovative algorithm.

III. OVERVIEW OF PROPOSED ALGORITHM

The proposed algorithm contains five components which are initialization, training, classification, evaluation and updating (as show in Fig.3). In initialization, the module of feature extraction extracts pixel-based visual color and texture features from the input image. Meantime, the module of RPDM is initialized according to the camera parameters by calibration [11]. Then, based on the initial RPDM, the dynamic training database (DTD) is built by randomly choosing certain number of positive training points and negative training points in the image which include visual features and probabilistic weights at the position of those points. Then in training process, the algorithm estimates the kernel parameters and trains the FSVMs classifier using the training data in DTD. In the component of classification, the features of all the pixels are extracted and classified. Given the classification result, morphological operation is implemented to reduce the noise in the classification result. By comparison of morphological result and classification result, the quality of current classifier is evaluated and the potential misclassified points are detected. Then, the DTD and RPDM are updated based on the result of evaluation component.

A detailed description of initialization, evaluation and updating is given in the next section. The components of training and classification are implemented according to the instruction of LIBSVM [12]. The readers can refer to [12] for the detail.

IV. DETAILED DESCRIPTION OF PROPOSED ALGORITHM

A. Initialization

In initialization, the module of feature extraction extracts pixel-based visual features such as color, texture. Meantime, the module of RPDM is initialized according to the camera parameters by calibration [11]. Then, the DTD is built by randomly choosing certain number of positive training points and negative training points in the image which include visual features and probabilistic weights.

1) Feature Extraction:

The visual features used in our algorithm are color features and texture features. For color features, hue, saturation, and value (HSV) representation of color is used. Texture is a measure of the local spatial variation in the intensity of an image. In this paper, the first five Haralick statistical features [13] are exploited as texture features. Those three color features and five texture features are combined to form an eight-element feature vector as following:

$$F_{i,j} = [\, f_{t_1(i,j)}, f_{t_2(i,j)}, f_{t_3(i,j)}, f_{t_4(i,j)}, f_{t_5(i,j)}, f_{c_1(i,j)}, f_{c_2(i,j)}, f_{c_3(i,j)} \,]$$
$$i = 1,\ldots,240 \quad j = 1,\ldots,320 \qquad (6)$$

where $f_{t_n(i,j)}$ is the n-th Haralick statistical texture feature at the point (i, j), $n = 1,\ldots,5$. $f_{c_m(i,j)}$ is the m-th color feature at the point (i, j) in HSV color space, $m = 1,2,3$.

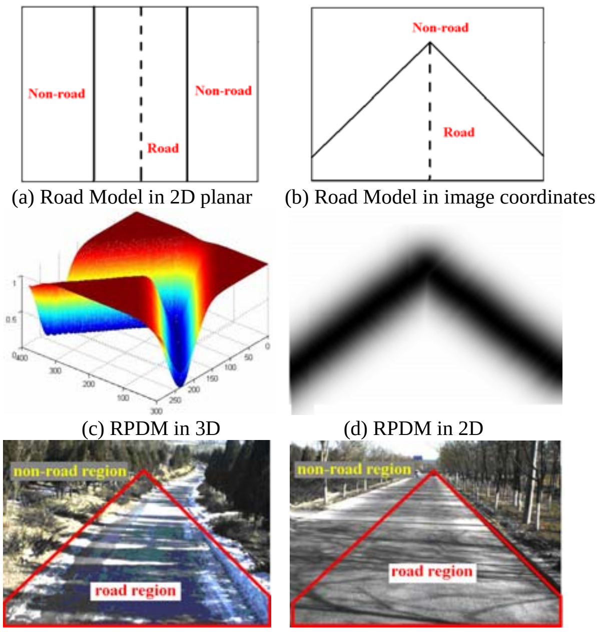
2) Initial RPDM:

RPDM means that the longer distance the pixel far from road edge, the higher weight the pixel has. In other words, we hope that the machine can learn more on those pixels which have high possibility in belonging to their classes. Unlike the other road models [1][14], our initial RPDM is not necessary to match the real road accurately. Without loss of generality, we build a flat and straight road model in initialization. Given the camera parameters and an initial road width by reasonable guessing, the road in 2D planar can be project into image coordinates as shown in Fig.4 (a-b). And the RPDM can be calculated as following:

$$s_{i,j} = 1 - e^{-f(i,j)}, i = 1,\ldots,240, \; j = 1,\ldots,360 \qquad (7)$$

$$\bar{s}_{i,j} = \begin{cases} s_{i,j} \Big/ \max\limits_{(i,j)\in road}(s_{i,j}) & if \quad (i,j) \in Road \\ s_{i,j} \Big/ \max\limits_{(i,j)\in nonroad}(s_{i,j}), & if \quad (i,j) \in Nonroad \end{cases} \qquad (8)$$

where $f(i, j)$ is the Euclidean distance between the pixel (i, j) to road edge. The RPDM of $\bar{s}_{i,j}$ is shown in Fig.4-c.



(a) Road Model in 2D planar (b) Road Model in image coordinates

(c) RPDM in 3D (d) RPDM in 2D

(e) RPDM on real road (f) RPDM on real road

Fig.8. Initial RPDM

Fig.4.Road Probabilistic Distribution Model Construction

3) Initial DTD

It has two stages to build the DTD: which are initial stage and updating stage. The latter will be discussed in the component of updating. In initial stage, the training points are randomly chosen from the road/non-road area in RPDM. To reduce the computation of training process, the size of DTD we used in our algorithm is limited to 1000 pairs of training points.

B. Evaluation

1) Morphological operation

Morphological operations [15] are commonly used to understand the structure of image. Morphological operations play a key role in applications such as machine vision and automatic object detection.

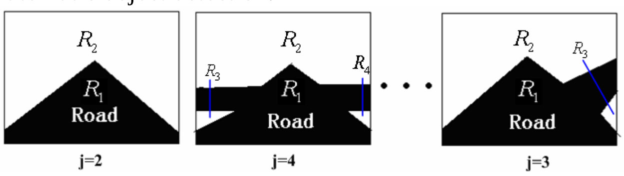


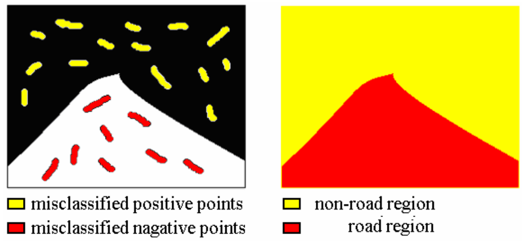
Fig.5.Simply connected road model. R_1 is simply connected road region. $R_j(j>1)$ is non-road region.

In this paper, the main morphological operation is flood-filling based on the assumption that road region is simply connected (as shown in Fig 5). This operation is implemented to determine the largest connected road region and erode all the holes (non-road pixels) in that connected road region. Then that largest connected road region is labeled as road region and all the other regions are non-road regions. The process of morphological operation is showed in Fig.6.

Fig.6.Morphological Operation (a): Classification result (white: represents road; black: represents non-road). (b): Largest connected road region (Red). (c): Erosion operation. (c): Morphological operation result (red is road region and yellow is non-road region)

2) Potential Misclassified Points Detection

Based on the comparison between classification result and morphological result (Shown in Fig.7) and the assumption of simply connected road, the points classified as road in classification result lying in the non-road regions of morphological result are assumed to be potential misclassified points, vice versa. The potential misclassified points labeled as new road samples or new non-road samples are considered to be candidate points prepared for updating DTD.



(a) Classification result (b) Morphological result

Fig.7.Comparison of Classification result and Morphological result

3) Classification Evaluation

The reason of online learning the road detection classifier is that the driving environment is continually changing in moving vehicle; the classifier performing high accurate result in last frame may not work well in next frame. In order to make the road detection method adaptive to environment changing, the proposed algorithm is designed to be capable of online learning according to the quality of last classification result derived from evaluation. The evaluation function is implemented to evaluate the performance of previous classification and determine if the online learning is necessary to be implemented for future road detection task. This evaluation function shown in the following formulas is also based on the assumption we mentioned in previous sections that the road region is simply connected.

$$E_{AFP} = \sum_{r=1}^{240}\sum_{c=1}^{360} V_1(r,c) \Big/ \sum_{r=1}^{240}\sum_{c=1}^{360} R_1^M(r,c) \qquad (9)$$

$$E_{AFN} = \sum_{j=2}^{N}\sum_{r=1}^{240}\sum_{c=1}^{360} V_j(r,c) \Big/ \sum_{j=2}^{N}\sum_{r=1}^{240}\sum_{c=1}^{360} R_j^M(r,c) \qquad (10)$$

$$E_{AF} = \sum_{j=1}^{N}\sum_{r=1}^{240}\sum_{c=1}^{360} V_j(r,c) \Big/ \sum_{j=1}^{N}\sum_{r=1}^{240}\sum_{c=1}^{360} R_j^M(r,c) \qquad (11)$$

$$V_j(r,c) = \begin{cases} 1, if\,(R_j^C(r,c) \neq R_j^M(r,c)) \\ 0, if\,(R_j^C(r,c) = R_j^M(r,c)) \end{cases}$$

$$r = 1,\dots,240; c = 1,\dots,360; j = 1,\dots,N \qquad (12)$$

where AFP refers to Assumption-based False Positive, AFN refers to Assumption-based False Negative and AF refers to Assumption-based classification False, the r and c denote the row and column, the $R_j^C(r,c)$ is the value of classification result at (r,c), the $R_j^M(r,c)$ is the morphological operation result at (r,c), the j is the number of region. Apparently, $V_j(r,c)$ indicates whether $R_j^C(r,c)$ and $R_j^M(r,c)$ are belonged to the same class. So the performance of the road detection classifier is assessed by computing the value of E_{AFP}, E_{AFN} and E_{AF} derived from the formulas (9)-(12).

Given the values of E_{AFP}, E_{AFN} and E_{AF}, we have three thresholds of $T_{E_{AFP}}$, $T_{E_{AFN}}$ and $T_{E_{AF}}$ to tune. If the value of evaluation is larger than its threshold, the retraining process is implemented.

C. *Updating*

The component of updating is the crucial part in proposed algorithm. It is this component that makes our road detection algorithm finding the optimized way in the next frame and adaptive to environment changing. This component includes two parts: RPDM updating and DTD updating. The updating process in proposed algorithm is trying to answer those questions: what we can inherit most from the previous result and how we can inherit them without reducing the performance of the classifier.

1) *RPDM Updating*

The RPDM is initialized in the component of initialization. We have to admit that the initialized RPDM is not very accuracy (See Fig.4). However, because the rough model ensure that most of correct training points have higher weights than the mislabeled training points, in other words, the decision hyperplane of classifier is affected more by the correct training points than mislabeled ones, the classification result in first frame is still acceptable. It is the combination of RPDM with FSVMs that solves the contradiction of inaccurate model and relative accurate classification result (Results are shown in Section V). After initialization, the RPDM is recalculated using equation (7) according to the current morphological result because it is more accurate and closer to the real road than the initial RPDM.

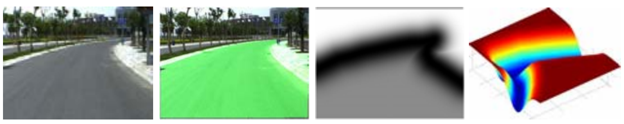
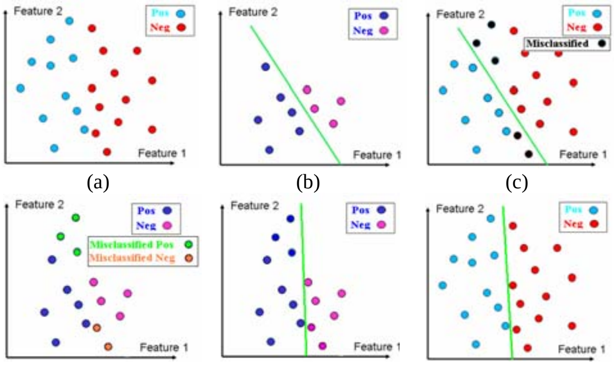

(a) (b) (c) (d)

Fig.9.RPDM updating (a).Road Image (b).Morphological result (c). Updated RPDM in 2D (d).Updated RPDM in 3D

2) *DTD Updating*

As we mentioned above, each training point in DTD is associated with weight, labeled class and visual features which are used for training FSVMs classifier. In initialization, the training points are selected from initial RPDM. Because of environment changing, the training points need to be updated online in order to well represent the changing environment. The DTD should be able to be updated according to environment. In proposed algorithm, the potential misclassified points are considered to be the candidate updating points. We provide an example in Fig. 10 to explain the contribution of misclassified points in retraining. The dataset is shown in Fig.10a. In Fig.10b, part of data is selected as training data and used for training classifier. From the result shown in Fig.10c we could see three positive points and two negative points are misclassified. Suppose that the misclassified is found and used for update the training data (shown in Fig.10d), the retrained classifier (Fig.10e) is perfectly classify all dataset in Fig.10f.



(a) (b) (c)
(d) (e) (f)

Fig.10.The contribution of misclassified points in retraining
(a) All dataset. (b) Training data and decision hyperplane (c) Classification result (d) Updated training data (e) Retrained decision hyperplane (f) Classification result

Given the above conclusion, there are still two questions in performing the process of DTD updating: Are all potential misclassified points really misclassified points? in order to limit the total amount of training points in DTD (1000 pairs of training points in this paper), how to discard the previous training data to leave more room for new training data in DTD?

(1) Are all potential misclassified points really misclassified points?

Definitely not. As we mentioned in Section IV, the potential misclassified points is generated by comparison of classification result and morphological result. Both are not ground truth. But the wrong decision in mislabeling the potential misclassified points high probably happens near the edge of morphological result because, even that edge is not real road edge, we still believe that edge is similar to real edge. When some points are mislabeled, it would not have much affect on retrained decision hyperplane because the RPDM weights those points by very low values. That is also a creative point we combine RPDM with FSVMs.

(2) How to discard the previous training data to leave more room for new training data in DTD?

Among those training points in previous DTD, the point which is not support vector has no contribution in determining the decision hyperplane [16]. Without loss of generality, we set the weights of those points as 0. From the previous trained classifier we could get the support vectors and their weights in determining the decision hyperplane (See equation (4) for proof). Then, we rank those training points by their weights in increasing order

$$\bar{w}_{x_1} \le \bar{w}_{x_2} \le \ldots \le \bar{w}_{x_{1000}} \quad (13)$$

Then, we randomly choosing T pairs of training samples from the potential misclassified points and discard the old training data from $x_1, \ldots, x_T$. T is a threshold to determine the updating speed. Too large value of T would lead our algorithm to over learning on new training data while too small would make our algorithm low adaptive to environment changing. After many experiments, it is recommended to set T as 1/20 of the size of DTD (T is 50 in this paper).

V. Result

We tested our method on the various unstructured road such as paint road and rural road. There are two functional stages in the implement of proposed algorithm: initial stage and online stage. In the initial stage, the algorithm is initialized by a rough RPDM and train the initial road classifier for road detection task. In the online stage, the algorithm is activated to learn the new training points from previous result and retrain the road classifier for future detection task. we provide the results of both stages. All the following results are compared with manually annotated frames to measure the accuracy.

A. *Results in initial stage*

We just assume that the autonomous vehicle is on the road without any accurate geometrical constraint. In that case, the road can successfully be detected using initial RPDM. We compare FSVMs with SVMs in this experiment,. From the result in Fig.11 one can see, the FSVMs is robust to bad training set and gets more accurate detection rate than SVMs.

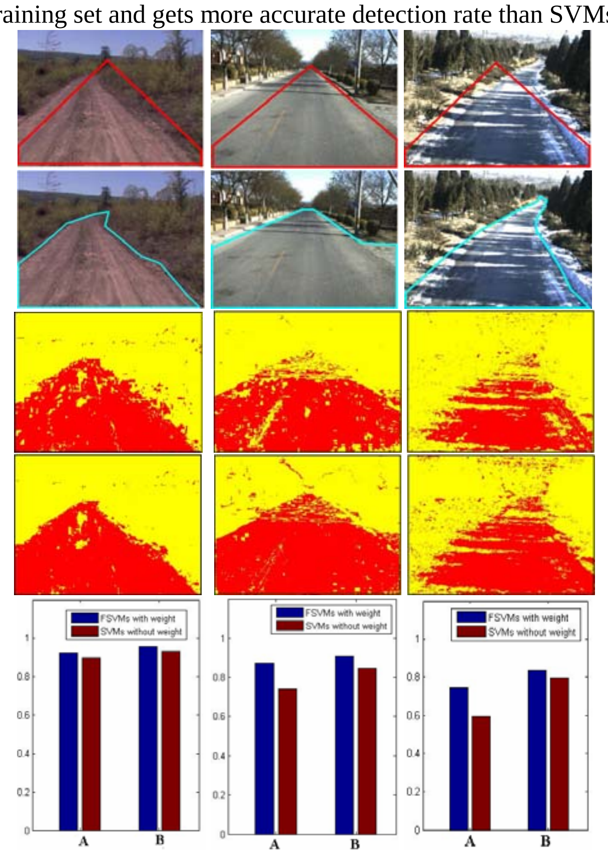


Fig.11. Initial detection results of SVMs and FSVMs

The first row shows initial RPDMs on images. The second row shows ground truth labeled by hand for detection accuracy calculation. The third row shows classification results by SVM classifier. The fourth row is the results detected by FSVMs classifier. The three charts in fifth row show the detection error rates of both classifiers. In each chart, A represents false negative rate while B represents detection error.

B. *Results in online stage*

We tested our method in two tough situations: concrete road with shadows and dust and rural road with snow and shadows (as shown in Fig.12). In order to demonstrate the necessary of online learning, we compare our online learning method to the offline learning method. Both methods start from the initial trained classifier we mentioned above. The former method can retrain itself through the online training process while the latter uses the initial trained classifier from beginning to end. From the result shown in Fig.12, our online learning method is capable of learning the novel training data and adaptive to environment changing.

VI. Conclusion

In this paper, we introduced the novel self-supervised road detection algorithm. The algorithm is able to effectively learn from previous result which makes the algorithm adaptive to environment changing. The primary innovation is that the combination of RPDM and FSVMs successfully conquers the contradiction of non-accurate model and relative accurate result. The secondary innovation is using the comparison of classification result and morphological result to get the potential misclassified points and use them for online learning. From many experiments, most of those points labeled as potential misclassified points are correct. The algorithm presented in this paper can also be seen as a novel framework for self-supervised online learning in the applications of vision-based region detection in the robot field.

References

[1] Albert S. Huang and Seth Teller, Lane Boundary and Curb Estimation with Lateral Uncertainties. Proc. Int. Conf. on Intelligent Robots and Systems, 2009. pp. 1729 – 1734.

[2] Chunzhao Guo, Mita, S, McAllester, D, Stereovision-based Road Boundary Detection for Intelligent Vehicles in Challenging Scenarios. Proc. Int. Conf. on Intelligent Robots and Systems, 2009. pp. 1729 – 1734.

[3] Peterson K. Ziglar J, Rybski P.E. Fast feature detection and stochastic parameter estimation of road shape using multiple LIDAR. Proc. Int. Conf. on Intelligent Robots and Systems, 2008. pp. 612 – 619.

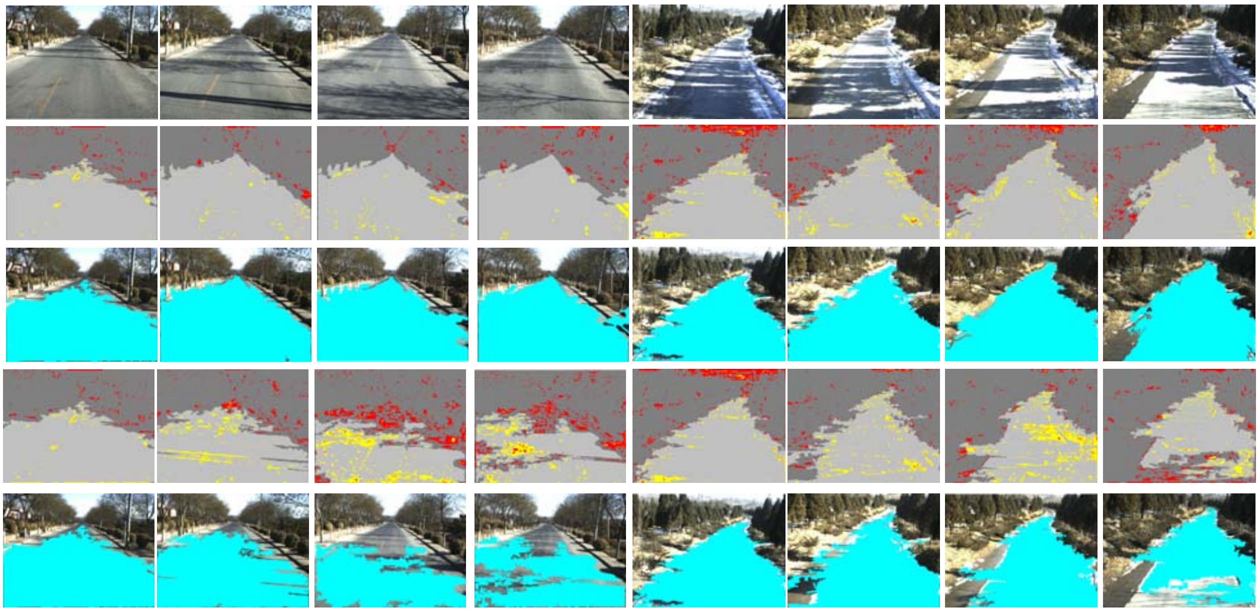
[4] Sehestedt S, Kodagoda S, Alempijevic A, Dissanayake, G. Robust lane detection in urban environments. Proc. Int. Conf. on Intelligent Robots and Systems, 2007. pp: 123 – 128.

[5] Zhu Wennan, Chen Qiang, Wang Hong. Lane Detection in Some Complex Conditions. Proc. Int. Conf. on Intelligent Robots and Systems, 2006. pp: 117 – 122

[6] Son, T.T., Mita, S., Takeuchi, A. Road detection using segmentation by weighted aggregation based on visual information and a posteriori probability of road regions. Proc. Int. Conf. on System, Man and Cybernetics, 2009. pp: 3018 – 3025

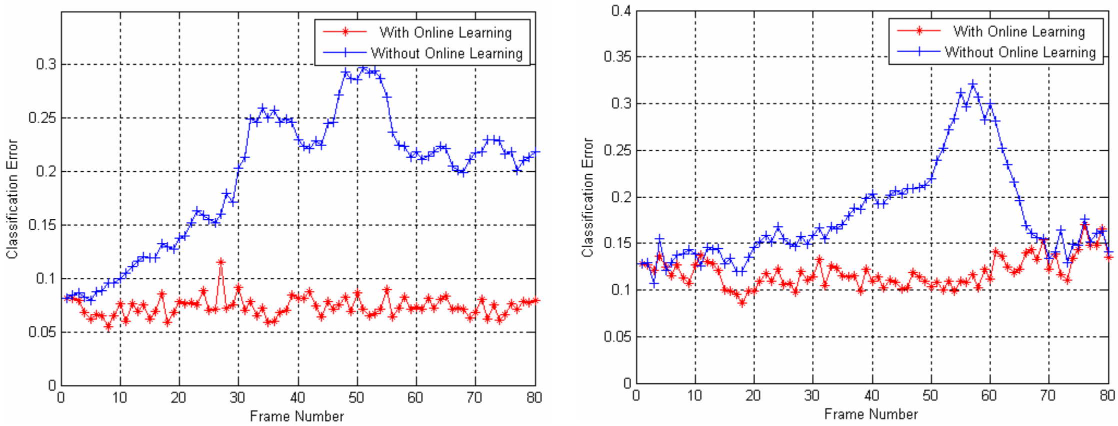
[7] D. Lieb, A. Lookingbill, and S. Thrun, “Adaptive Road Following using Self-Supervised Learning and Reverse Optical Flow”, in Online Proceedings of Robotics: Sciences and Systems, Cambridge, USA, 2005.

[8] Jian Wang, Zhong Ji, Yu-Ting Su. Unstructured road detection using hybrid features. Proc. International. Conf. on Machine Learning and Cybernetics. 2009. pp: 482 - 486



Frame 1 Frame 20 Frame 60 Frame 70 Frame 1 Frame 20 Frame 50 Frame 60

(a) Result in scene one (b) Result in scene two



(c) Detection Error rate (d) Detection Error rate

Fig.12 Results in online stage

The first rows in (a) and (b) shows the original images for detection. The second rows in (a) and (b) are classification results of online learning method. The third rows in (a) and (b) shows the morphological results after morphological operations on those classification results. The fourth rows provide the results classification result by offline learning method and the fifth rows are their morphological results. (c) is the detection errors of both methods in consecutive frames in concrete road. (d) is the detection errors of both methods in consecutive frames in rural road.

[9] Foedisch,M.Takeuchi,A. Adaptive road detection through continuous environment learning. Proc on Applied Imagery Pattern Recognition Workshop, 2004

[10] Chun-Fu Lin. Sheng-De Wang. Fuzzy support vector machines. IEEE Transactions on Neural Networks. vol. 13, no. 2, pp. 464–471, Mar. 2002.

[11] Z. Zhang. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, No.11, pages 1330-1334, 2000.Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[12] Haralick, R.M. Statistical and Structural Approaches to Texture. Proceedings of the IEEE, 67, pp. 786-804, 1979

[13] Y. Wang, E. Teoh, and D. Shen, Lane detection and tracking using B-snake, Image Vision Computing. vol. 22, no. 4, pp. 269–280, Apr. 2004.

[14] Soille, P.. Morphological Image Analysis: Principles and Applications. Springer-Verlag, 1999, pp. 173-174.

[15] Christopher J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2:121–167, 1998.