Video-based Lane Detection using Boosting Principles

Raghuraman Gopalan[1], Tsai Hong[2], Mike Shneier[2], and Rama Chellappa[1]
[1] Dept. of ECE, University of Maryland, College Park, MD 20742
[2] National Institute of Standards and Technology, Gaithersburg, MD 20899
raghuram@umiacs.umd.edu, {hongt, shneier}@cme.nist.gov, rama@umiacs.umd.edu

Autonomous navigation of road vehicles is a challenging problem that has widespread applications in intelligent systems, and robotics. An integral component of such a system is to understand how the road is structured. Detection of road lane markings assumes importance in this regard, and this problem has been approached with different visual input- based inference algorithms ([1], [2]), besides other sensing modalities such as the GPS sensor and the internal vehicle-state sensors. But the challenge still remains when there is considerable amount of shadows on the road, variations in outdoor lighting conditions of the scene (transition from day to night), among others.

To address such issues, we propose a machine learning approach based on Real Adaboost [3], and train linear classifiers for both the appearance and edge cues of the training examplars. Additionally, we incorporate prior knowledge about the relative importance of the training samples by computing their weights using kernel discriminant analysis [4], before learning the classification function through boosting. The regions identified as lane markings are then analyzed for gradient direction consistency before making the final detection decision. We illustrate the effectiveness of our algorithm on challenging daylight and night-time road scenarios.

A. Real Adaboost for Lane detection

For the set of two-class problems with labeled training examples, boosting algorithms try to learn a linear classification function, by a weighted combination of several weak classifiers. Real Adaboost [3] is one such algorithm that allows for a range of classification intervals for each weak classifier rather than having a fixed threshold, and thereby gives more flexibility to handle data points with some noise. Before applying such a framework to the problem of lane detection, we make two observations [5]; (i) the lane markings have strong gradient magnitude patterns, and (ii) regions around the lane markings have similar appearance when compared with other regions in the scene. To utilize both channels of information, we train two separate layers of Real Adaboost (Figure 1a), with the first layer trained on a novel set of lines and curves (of different slopes and curvatures) to capture the edge patterns, and the second layer trained with the set of Haar-like features [6] to capture the appearance variations in the training samples. These two cues, when analyzed in tandem, provide more robustness especially in the presence of shadows, which affects the edge strengths much more than the appearance.

B. Learning prior information about training sample weights

Before learning the classification function using the boosting framework, we would like to address the following question: "How representative are the training samples to their own class, and how discriminative are they to the other class?" Normally in boosting, all training samples are initialized to the same weight, and then the boosting algorithm reweighs the samples to obtain the final classification function. But in practice, some training samples are *evidently much more* representative of a class than others. Such situations are correlated with the *quality* of training samples chosen by the user. So, intuitively it makes sense to get some prior information about the discriminative power of training samples, before performing boosting. Towards this end, we apply kernel discriminant analysis [4] on the training data, and analyze how well-placed each sample is with respect to its class mean. Let ($\mathbf{X_i}$, y_i), i∈{1,2,…,N} represent N training samples, where $\mathbf{X_i} \in R^n$ is the i^{th} training example, and $y_i \in \{0,1\}$ its class label, and let **k** be the kernel function (here we have used a RBF kernel) that transform the data $\mathbf{X_i} \in R^n$ to $\mathbf{Y_i} \in$ Ғ, where Ғ is the new feature space. The Fisher linear discriminant in Ғ is then obtained by maximizing

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \tag{1}$$

where M and N denote the between-class scatter matrix and within-class scatter matrix respectively. The most informative eigenvectors of α are then used to project $\mathbf{Y_i}$ onto $\mathbf{Z_i}$ (which lives in a lower dimensional subspace). These $\mathbf{Z_i}$'s are now analyzed to see how discriminative they are, by computing

$$\xi_i = \frac{|(Z_i - \mu_l)|}{\sum_{\forall k: y_k = y_i} |(Z_k - \mu_l)|} \tag{2}$$

where μ_l, $l \in \{0,1\}$ is the class mean corresponding to that of training sample Z_i. Then, if w_i = 1/N denote the weights of the training samples to be classified using Adaboost, the new initial weights ($\tilde{w}_i$) are given by

$$\tilde{w}_i = w_i * \exp(-\delta * \xi_i) \tag{3}$$

where δ is the factor controlling the importance of the weights learned from (2). Here, we set δ =.01 experimentally. The new set of initial weights $\tilde{w}_i$ is then normalized, and the classification functions are learnt using the Adaboost framework discussed in Section A. Adding such prior information about the training samples improves the convergence time of the boosting algorithm, and results in better detection performance (Fig 1b, 1c).

C. Post-processing and Experimental Validation

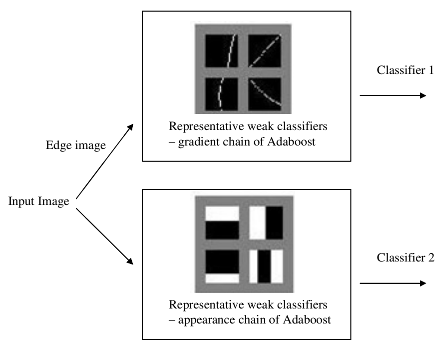
We then integrate the detection results obtained from the two boosting classifiers using an OR logic on their probability estimates. Specifically, if X is the image region under analysis, final Adaboost detection result is obtained using the following formula,

Category: Visual Processing and Pattern Recognition. Preference: Oral

$$[P_E(X) > threshold] \underset{(OR)}{\vee} [P_A(X) > threshold] \Rightarrow [P_I(X = lane\,marking) = 1] \quad (4)$$

where P_E and P_A represent the probability estimates of the edge chain, and appearance chain of Adaboost respectively, and P_I represent the detection decision of the integrated boosting classifier. We then perform post-processing by observing that the lane markings maintain roughly similar gradient orientations [5] when compared with their neighboring points on the lane. Further, it has been shown in [7] that the gradient *orientation* of the points is approximately invariant to lighting changes, which comes in very handy in presence of shadows. Hence, the boosting results (obtained from (4)) are verified for gradient orientation consistency along their neighborhood to obtain the final detection result.

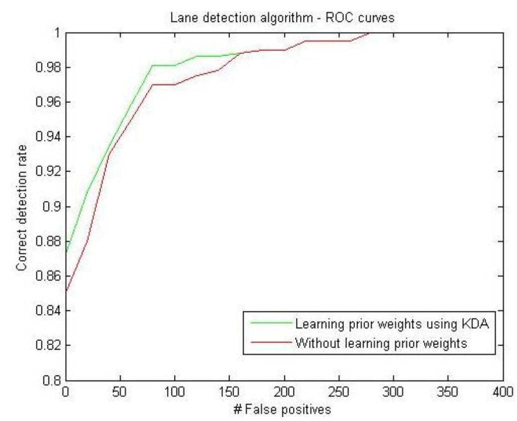
We tested the lane detection algorithm on video sequences collected at NIST, during both day and night-time. The boosting algorithm, implemented in MATLAB®, was trained separately with examples under daylight and night-time to learn the appropriate classification functions. The Receiver operating characteristic (ROC) curves (given in Fig 1c) was obtained for a set of 100 road images collected under different settings, with the correct lane marking regions appropriately hand-marked. Some sample lane detection results are also provided in Fig 1d-1f. It can be seen that the performance of the detection algorithm is quite robust even under changing lighting conditions of the scene. We believe this is primarily due to the choice of the feature description that captures both the appearance and edge patterns of the lane markings.



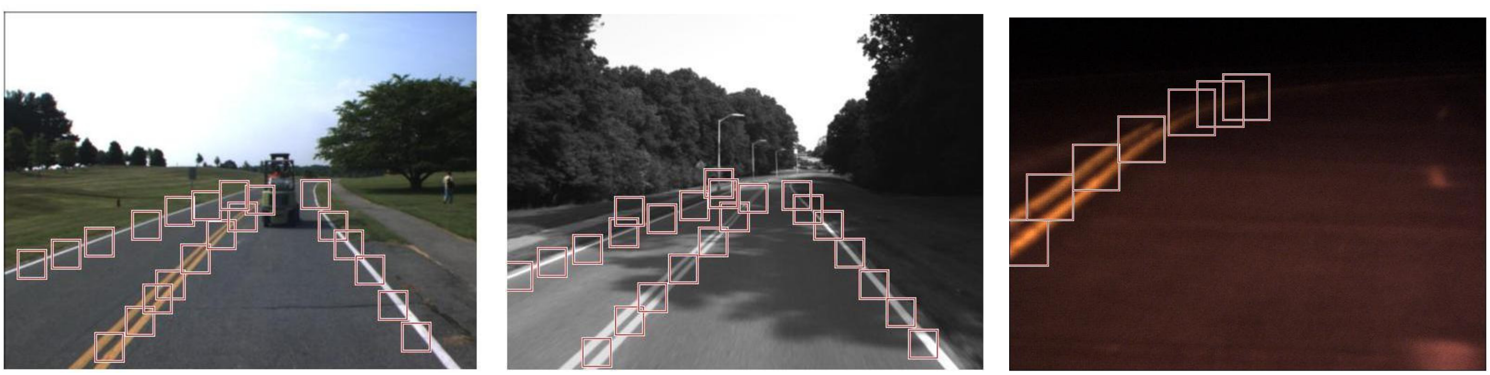
a

Nested cascade stages	< Convergence time (in minutes), # weak learners required to satisfy true detection – false alarm criteria of the stage>	
	With prior learning of training weights	Without prior learning of training weights
1	<5, 2>	<5.3, 2>
2	<5.9, 5>	<6.2, 5>
3	<7.3, 5>	<7.7, 6>
4	<8.5, 10>	<8.9, 11>
5	<10, 10>	<10.4, 12>
6	<12.7, 18>	<12.9, 20>
7	<15, 20>	<15.3, 21>
8	<17.9, 20>	<17.9, 20>
9	<18.2, 50>	<18.3, 50>
10	<18.5, 50>	<18.6, 50>
11	<19, 50>	<19, 50>

b



c



d e f

Fig 1: (a) Training stage of Adaboost, (b) Comparison of convergence times, with and without learning weights using KDA, (c) ROC curves of the Lane detection algorithm, (d)-(f) Sample lane detection results

REFERENCES

[1]. J.C. McCall, and M.M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system and evaluation", IEEE Transactions on Intelligent Transportation Systems, Vol 7(1), 2006.

[2]. Z. Kim, "Robust lane detection and tracking in challenging scenarios", IEEE Transactions on Intelligent Transportation Systems, Vol 9(1), 2008.

[3]. R.E. Schapire, and Y. Singer, "Improved boosting algorithms using confidence-rated predictions", Machine Learning, Vol 37(3), 1999.

[4]. J. Yang et al, "KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 27(2), 2005.

[5]. E.D. Dickmanns, "Dynamic vision for perception and control of motion", Springer 2007.

[6]. C.P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection", Sixth International Conference on Computer Vision, 1998.

[7]. A.J. Fitch et al, "Orientation correlation", British Machine Vision Conference, 2002.