

Preceding Vehicle Trajectory Prediction by Multi-Cue Integration

Feng Han, Yi Tan, and Jayan Eledath
Sarnoff Corporation

201 Washington Road, CN5300, Princeton, NJ 08543-5322
{fhan, ytan, jeledath}@sarnoff.com

Abstract

In this paper we describe an approach to detect and predict the driving trajectory of a preceding vehicle on highway. In particular, we focus on detecting and predicting the changing lane intention and action of the preceding vehicle. Our algorithm employs SVM for driving pattern recognition by integrating two different cues: motion cue and appearance cue, which is trained on two class feature sets extracted from examples of lane changing and lane keeping video sequences. The method is evaluated on the real-world data collected in an intelligent vehicle test-bed. The method is applied to a vision-based safety driving system, which tracks the lane, the preceding vehicle, and uses the vehicle lane-change warning to serve for other intelligent vehicle controls.

1. Introduction

Detecting and predicting vehicle-moving intention can provide valuable information for any intelligent vehicle and driver support system. However it is a challenging problem since a vehicle's move is affected by many factors, such as traffic/road conditions as well as driver's behavior. In this paper, we formulate the problem as a pattern classification problem on dynamic scenarios. Video processing based intelligent system for vehicle driving safety control has achieved significant progress in recent years, but the majority of applications are focused on analyzing either the stationary environment (e.g. the lane finding) or detecting the stationary or moving obstacles (e.g. vehicle or road sign) within the area of interest [1][9]. There are works done lately on driver's intention inferring [2][3], by using multi-modal data (e.g. road scene, CAN data, eye movement, etc.), which leads to predicting vehicle's lane changing move ahead of time. However, the work is mostly applied to a host vehicle instead of a front (target) vehicle, in which the driver can't be closely monitored and there is no specific sensor available.

In this paper we describe our work to classify the preceding vehicle driving-pattern, with emphasis on detecting and predicting the lane change intention. Our method is based on image processing for scene feature extraction, queuing lane-vehicle position data for lane-changing and lane-keeping classification. We implemented the algorithm in a vision system, which takes the feed from a monochrome camera for real-time video acquisition. The scenario of the video processing involves real-time lane following, ground plane extraction, and host-vehicle 3-D pose estimation. By using the lane geometry constraints, the algorithm detects, verifies, and tracks the preceding vehicles. The time-stamp augmented

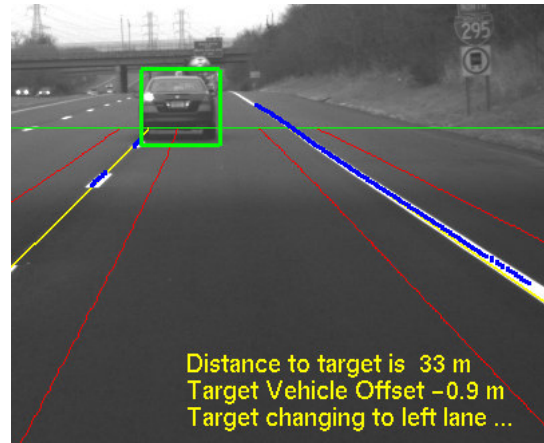


Figure 1. Vehicle lane-change warning output

3-D trajectory of a tracked vehicle is used for lane-changing recognition by Support Vector Machine (SVM). The method is tested significantly on real-world data collected on highways. Figure 1 shows an example of our system output on a scene where a preceding vehicle is detected and tracked and its lane-changing move is warned in real-time.

2. Vision system architecture

The block diagram of our vision system architecture is shown in Figure 2. In this section, we briefly describe the algorithms and functions of each module.

2.1. Lane detection and tracking

For lane feature extraction, we first calculate the Laplacian image in the regions defined around lane's boundary. Then we apply two thresholds to the result image to generate a tri-valued pixel map. Boundary features are extracted by matching the pixel map to a lane boundary template. The features are then used to fit a quadratic function. We use a Kalman filter (KF) to track the motion of the lane boundary. From frame to frame, the boundary locations are predicted by the KF, new features are extracted to refine the boundary function and update the KF states.

2.2. Ground plane estimation

The vanishing point, which is defined as the intersection of two tracked lane boundaries, is used to estimate the pitch angle change of the camera. The estimated angle is then used to update the camera projection matrix M , which is obtained via a pre-calibration (extrinsic) process. The

incremental pitch estimation is shown in the following equation, where V_{y1} and V_{y2} are vanishing point in image y position from consecutive frames, f_y is the camera's lens focal length, and θ and $\Delta\theta$ correspond to camera pitch angle from calibration settings and incremental change from frame to frame.

$$\tan(\theta + \Delta\theta) = \frac{V_{y2} - V_{y1}}{f_y}$$

In our 3-D coordinate system, Z -axis is defined as pointing towards front; X -axis is pointing to right; and Y -axis is pointing down. We define the ground as $Y=0.0$ in 3-D space. Under this definition, all the ground points $(X, 0, Z)$ in the scene can be mapped to image $I(u, v)$ by using M : $I(u, v) = M(X, 0.0, Z)$. The inverse transformation, M^{-1} , is also computed to map image points to 3-D ground: $(X, Y=0.0, Z) = M^{-1}(u, v)$.

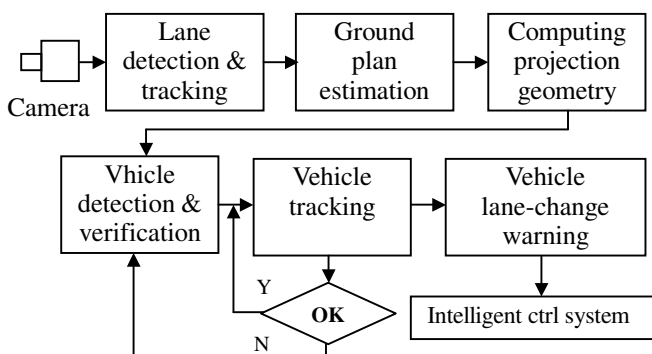


Figure 2. Vision system diagram

2.3. Vehicle detection, verification and tracking

Using the 3D constraints from the vanishing point, lane boundaries, and ground plane, a search region is defined for preceding vehicle search. Canny edge operator is applied to search region for horizontal and vertical line extraction. The line features are assembled to make candidate vehicle hypotheses, which are then confirmed by the verifier. Verifier employs a SVM with HOG (Histogram of Oriented Gradient) feature [10] for vehicle recognition. A pre-learned vehicle model database is used by the SVM. Upon passing the verification, the vehicle is handled over to a tracker, which is implemented using KLT [4] algorithm. In each frame, the central bottom of the tracked vehicle is used to calculate its 3D position, e.g. the distance (to camera) and the offset (to lane center) in the scene by the transformation M , as described in previous section.

2.4. Vehicle motion classification

To describe a vehicle's moving behavior, two sets of vehicle driving training sequences (one for changing lane and the other for staying-in lane) are collected from highway traffic to train a SVM classifier. In run-time, the 3D trajectory of the preceding vehicle is fed to the SVM for driving intention recognition. Details of the algorithm are described in section 3.

3. Moving pattern classification algorithm

We infer the change lane intention of preceding vehicle from a pattern recognition perspective; namely, given a numbers of features of preceding vehicle, how can we infer or classify its change lane intention: lane changing (either left or right) and lane keeping.

What type of feature to use is a key step in classification task. Unlike the previous work on inferring host vehicle's intention, where there are plenty of data available, we only have very limited data for the preceding vehicle. In this work, both motion cue and appearance cue of the preceding vehicle are introduced and incorporated as the feature to infer its lane changing intention.

3.1 Motion Cue: Vehicle Motion Trajectory Relative to Lane Center

After detecting the lane boundaries and tracking the preceding vehicle in other modules of the vision system, the direct feature for lane changing is the vehicle motion trajectory relative to the center of the lane. More specifically, when the preceding vehicles intends to change lanes, its relative distance to the lane center tends to take some specific patterns or trends as illustrated in Figure 4, which makes the vehicle trajectory a good feature to reveal the lane change intention.

We define the central-bottom of the tracked vehicle as its position. We map both vehicle's position and the lane boundary to 3-D for trajectory computation. Vehicle's

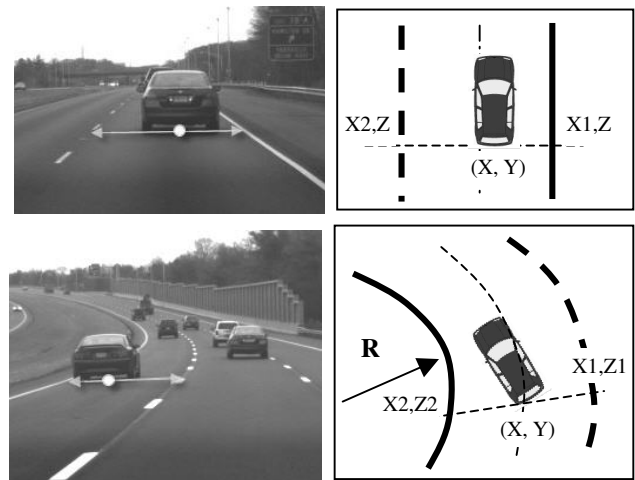


Figure 3: Vehicle trajectory computation at straight and curved lane

position in 3-D is calculated with respect to the center of the lane, as shown in Figure 3.

In straight lane case, as shown in top row in Figure 3, the bottom of the vehicle position in image (x, y) , is mapped to world coordinate (X, Y) . If we draw a horizontal line through this point, the line will intercept with lane boundaries at location $X1$ and $X2$, at the same distance Z . The lane center location is $((X1+X2)/2, Z)$. Vehicle's trajectory position, w.r.t. the lane center, is calculated as $(X-(X1+X2)/2, Z)$. In curved lane case, as shown in bottom row in Figure 3, the bottom of the vehicle position in image (x, y) , is mapped in the same manner to world coordinate (X, Y) . Since the pose of the vehicle is now parallel with lane curvature $(1/R)$, the intercepting line

now needs to be rotated $1/R^\circ$ and intercepts with lane boundaries at $(X1, Z1)$ and $(X2, Z2)$, where $Z1$ and $Z2$ are different. The lane center location is $((X1+X2)/2, (Z1+Z2)/2)$. Vehicle's position, w.r.t. the lane center, is calculated as $(X-(X1+X2)/2, Z-(Z1+Z2)/2)$.

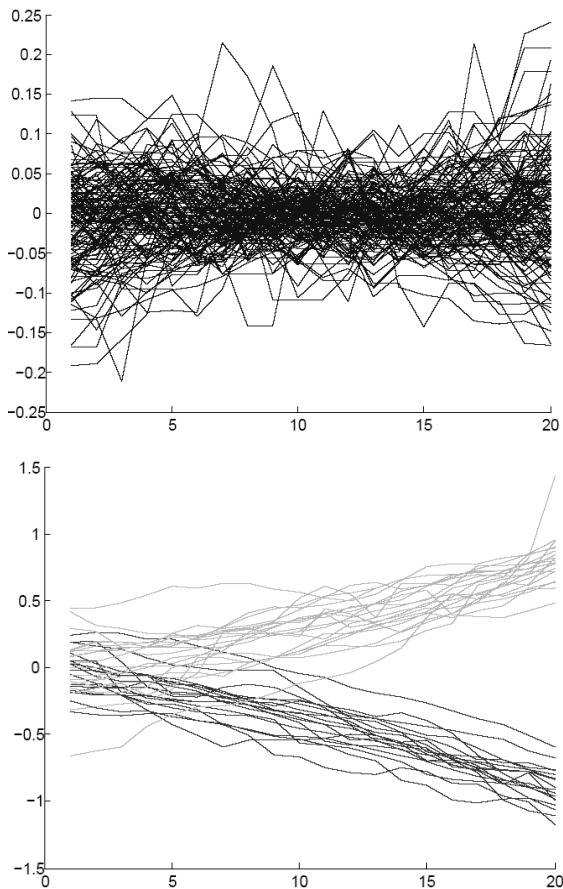


Figure 4. Typical preceding vehicle motion trajectories: top – staying in lane (scaled up to show small variations); bottom – changing to right (gray) and left (black) lane.

3.2 Appearance Cue: Vehicle Appearance Change Relative to Tracking Template

When the preceding vehicle intends to change lane, its appearance of the rear part also changes due to the rotation. This makes appearance change also a strong cue to predict the lane change intention. To represent the appearance change, we use the tracking results.

We use KLT tracker for tracking the preceding vehicle. This method minimizes the following error between the pre-established template and the sub-region in the current frame:

$$E(p) = \sum_x [I(W(X; p + \Delta p)) - T(x)]^2, (1)$$

Where warping parameter $p=(p_1, p_2, p_3, p_4, p_5, p_6)$ represents the transformation from the template to the sub-region in the image, $W(x;p)$ is the warping function, $T(x)$ is the online template.

Therefore the appearance relationship between the current tracked preceding vehicle and the template is encoded in the warping parameters, which can be used as

the representation for the appearance change during the lane changing process.

3.3 SVM Classifier

Since the vehicle may change lane at different lateral offset from the lane center, one decision can't be obtained by simple thresholding. In this paper we choose the support vector machine [5,6] as the classifying function. The Support Vector Machine (SVM) is a statistical learning method based on the structure risk minimization principle. Its efficiency has been proven in many pattern recognition applications [5,7]. In the binary classification case, the objective of the SVM is to find a best separating hyperplane with a maximum margin.

The form of a SVM classifier is:

$$y = \text{sign}(\sum_{i=1}^N y_i a_i K(x, x_i) + b), (2)$$

where x is the feature vector of an observation example, $y \in \{+1, -1\}$ is a class label, x_i is the feature vector of the i^{th} training sample, N is the number of training samples, and $K(x, x_i)$ is the kernel function. The weight $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$ and constant b are computed through the learning process.

One distinct advantage of this type classifier over traditional neural networks is that support vector machines achieve better generalization performance. While neural networks such as multiple layer perceptrons (MLPs) can produce low error rate on training data, there is no guarantee that this will translate into good performance on test data. Multiple layer perceptrons minimize the mean squared error over the training data (empirical risk minimization) where support vector machines use an additional principle called structural risk minimization [6]. The purpose of structural risk minimization is to give an upper bound on the expected generalization error.

Compared with the popular Adaboost classifiers, SVM is slower in the test stage. However, the training of SVM is much faster than that of Adaboost classifiers.

4. Experiments

4.1 Training

Our training image sequences are collected by cooperation of two cars, in which the preceding car intently frequently changes lane. From these sequences, we cropped ~200 clips. Each of these clips corresponds to one lane-change occurrence and they are used as the positive training samples. We also cropped ~300 clips when the preceding vehicle staying in the lane and use them as initial negative training samples.

One issue with using SVM for lane change detection was that lane changes do not have fixed time length. Lane changes vary anywhere between 1 to 5 seconds and thus the corresponding clips having different number of frames. Direct temporal mapping between the data and SVM classification is not possible. Longer lane changes see a smooth transition in features values like straight motion

trajectory; whereas shorter ones have a relatively abrupt transition. So we normalize the entire lane-changing feature by either interpolation or extrapolation.

One very important issue in the classifier training for one object class is how to select effective negative training samples. As negative training samples include all kinds of images, a prohibitively large set is needed in order to be representative, which would also require in feasible amount of computation in training. To alleviate this problem, a bootstrapping method, proposed by Sung and Poggio [8], is used to incrementally train the classifier as illustrated in Figure 5.

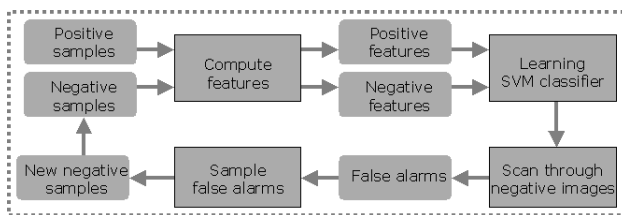


Figure 5. The bootstrap training schematic.

Even the preceding vehicle may change to either left lane or right lane; we can train only one classifier for changing to left lane since after reverse the signs of all the feature values, this classifier can be used for changing to right lane. Therefore, we convert all the positive training samples corresponding to right-lane change to left-lane change and train one classifier.

4.2 Testing

We test our algorithm on a two-hour-long sequence captured on freeway. In the testing stage, we keep a 20-frame long feature buffer for the tracked preceding vehicle to store its motion trajectory and appearance change. Once this feature buffer is classified as changing lane, a reminding is displayed as illustrated in Figure 1.

To show the performance improvement achieved by

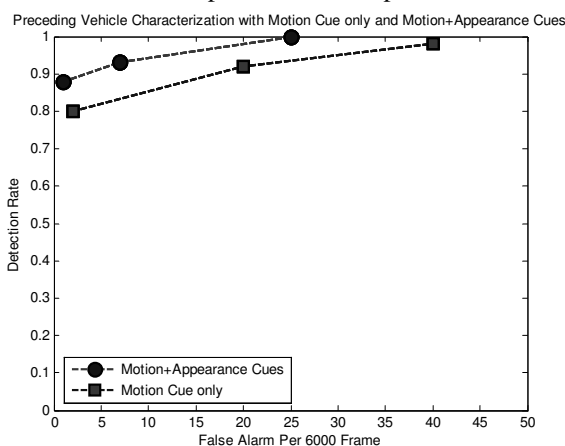


Figure 6. ROC curves for the system.

incorporating the two different cues, we compare the performance of the change-lane warning module by turning on and off the appearance change cue. In Figure 6, we show the two ROC curves corresponding to turning on

and off the appearance change cue when inferring the change lane intention of the preceding vehicle. From these comparisons, we can clearly see that the incorporation of the two cues improves the performance.

5. Summary and Conclusion

In this paper, we discussed in details the method we have proposed for inferring the change lane intention of a preceding vehicle. The algorithm has been extensively tested on real-world data. The experiment results show that the method works reliably using multiple image cues, which are extracted from lane-vehicle tracking process. The whole system achieves about 10 Hz running speed on a PC with Intel 1.8 GHz Duo-Core CPU.

Acknowledgements

We wish to thank Dr. Farooq Ibrahim of Visteon Corporation for his assistance during data collection and valuable discussions during the course of this work.

References

- [1] V. Kastinaki, M. Zervakis and K. Kalaitzakis: "A survey of video processing techniques for traffic application," *Image and Vision Computing*, vol. 21, no. 4, pp. 359-381, 2003.
- [2] J.C. McCall, D. Wipf, M.M. Trivedi, and B. Rao: "Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning," *IEEE CVPR Workshop: Machine Vision for Intelligent Vehicles*, vol. 3, pp59-67, 2005
- [3] D.D. Salvucci: "Inferring driver intent: A case study in lane-change detection," *Proceedings of the Human Factors Ergonomics Society 48th Annual Meeting*, 2004
- [4] I. Matthews, T. Ishikawa and S. Baker: "The Template Update Problem", *IEEE Transactions on Pattern Analysis and Intelligence*, Vol. 26, No. 8, 2004.
- [5] E. Osuna, R. Freund, and F. Girosi: "Training support vector machines: an application to face detection", *Proceedings of the IEEE Conference on Computer Vision and Patter Recognition*, pp. 130-136, 1997.
- [6] V. Vapnik: "The nature of statistical learning theory", *New York: Springer-Verlag*, 1995.
- [7] B. Heisele, T. Serre, S. Prentice, and T. Poggio: "Hierarchical classification and feature reduction for fast face detection with support vector machines", *Pattern Recognition*, 36(9):2007-2017, Sep 2003.
- [8] K. K. Sung and T. Poggio : "Example-based learning for view-based human face detection", *IEEE Transactions on Pattern Analysis and Intelligence*, vol. 20, no. 1, pp. 39-51, 1998.
- [9] M. Betke, E. Haritaoglu, and L. S. Davis: "Real-time multiple vehicle detection and tracking from a moving vehicle", *Machine Vision and Applications*, vol. 12, No. 2, pages 69-83, Aug. 2000.
- [10] Navneet Dalal and Bill Triggs: "Histograms of Oriented Gradients for Human Detection", *Proceedings of the IEEE Conference on Computer Vision and Patter Recognition*, 2005.