Structural Signatures for Passenger Vehicle Classification in Video

Ninad S. Thakoor, *Member, IEEE*, and Bir Bhanu, *Fellow, IEEE*

***Abstract*—This paper focuses on a challenging pattern recognition problem of significant industrial impact, i.e., classifying vehicles from their rear videos as observed by a camera mounted on top of a highway with vehicles traveling at high speed. To solve this problem, this paper presents a novel feature called structural signature. From a rear-view video, a structural signature recovers the vehicle side profile information, which is crucial in its classification. As a vehicle moves away from a camera, its surfaces deform differently based on their relative orientation to the camera. This information is used to extract the structure of a vehicle, which captures the relative orientation of vehicle surfaces and the road surface. This paper presents a complete system that computes structural signatures and uses them for classification of passenger vehicles into sedans, pickups, and minivans/sport utility vehicles in highway videos. It analyzes the performance of the proposed system on a large video data set.**

***Index Terms*—Image motion analysis, object recognition, vehicles.**

I. Introduction

VEHICLE classification is an important industrial application of computer vision and pattern recognition technology. Vehicle class information can be useful in traffic analysis, class-based tolling, security applications, surveillance tasks, and law enforcement. For a vehicle classification system to be useful in a real-world application, it must be robust to illumination changes, shadows, partial detections, occlusion, tracking failure, imaging system changes, camera viewpoint changes, etc. Current vehicle classification methods that rely on blob features or appearance features cannot meet these requirements. Table I summarizes related publications.

Gupte *et al.* [1] use vehicle dimensions to classify their side views in real time; however, the classification is only limited to sedans and nonsedans. With a constellation model, Ma and Grimson [3] classify sedans versus taxis and sedans versus minvans in their edge-based approach. They use oblique side views of vehicles in their work. Note that a side view of a vehicle can be easily occluded on multilane roads. Additionally, most of the cameras deployed along the road capture rear or front views of a vehicle, reducing the applicability of side-view-based techniques. Morris and Trivedi [7] also use side views of vehicles and blob features to classify vehicles. Kafai and Bhanu [8] use a hybrid dynamic Bayesian network to classify rear views of vehicles. They use features such as locations and dimensions of landmarks (e.g., license plates and tail lights) as well as their spatial relationships in the network. Detection of these high-level landmarks is challenging under varying environmental conditions.

Other work related to vehicle recognition focuses on recognizing the make and model of a vehicle. Petrovic and Cootes [9] compare various appearance features for identifying the make and model of vehicles from their frontal views. Negri *et al.* [10] use oriented contour features of frontal views to classify vehicles. Pearce and Pears [11] use a recursive partitioning scheme with Harris corner features to identify the class of a vehicle. All of these approaches use appearance information, which can widely change under varying environmental conditions. Therefore, the applicability of these approaches in real-world scenarios is limited.

All of these current methods either capture the appearance or blob structure of a vehicle. None of these methods use structural information that can be inferred from multiple views in the classification. An incremental approach by Ghosh and Bhanu [12] uses information from multiple video frames to learn the 3-D model of a vehicle, which can be used for vehicle classification. However, this approach is unsuitable for real-time implementation. In view of the state of the art, the contributions of this paper are as follows:

1) development of a vehicle classification system from the rear-view video of a vehicle, which classifies vehicles into three classes: sedan, pickup, and minivan/sport utility vehicle (SUV);
2) introduction of a novel feature called structural signature that captures side profile of a vehicle from rear-view video data;
3) integration of information from multiple video frames in the signature computation as well as in classifier decision making;
4) validation with 1664 vehicle sequences extracted from real-world videos.

A concise version of this paper appeared in [13].

This paper is organized as follows. Section II motivates the structural signature approach. The technical approach is outlined in Section III. The experimental results are presented in Section IV and this paper is concluded in Section V.

II. Motivation

Fig. 1 shows vehicle rear views from various categories. The canonical vehicle surfaces visible from a rear view are either

Manuscript received January 17, 2013; revised April 12, 2013; accepted June 1, 2013. Date of publication July 15, 2013; date of current version November 26, 2013. This work was supported in part by a grant from Federal Signal Corporation. The Associate Editor for this paper was S. S. Nedevschi.

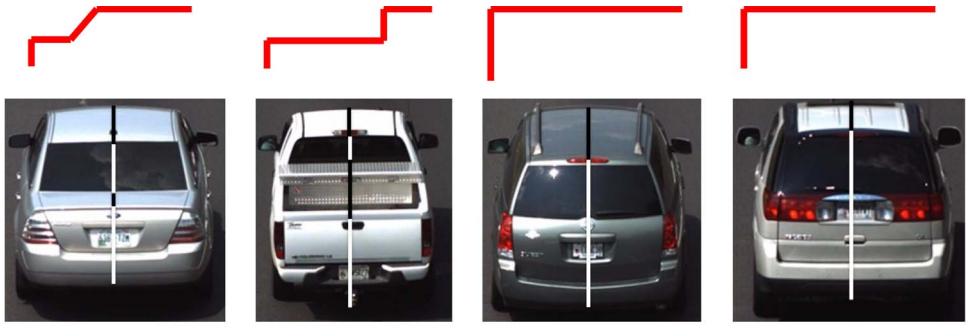
The authors are with the Center for Research in Intelligent Systems, University of California, Riverside, CA 92521 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2013.2269137

TABLE I
RELATED WORK

Publication	Principle of the Method	Vehicle Data	Classes (Accuracy)
Gupte et al. [1]	Change detection, vehicle size based classification	Highway, front	Cars/ non-cars (70%)
Avery et al. [2]	Change detection, vehicle length based classification	Highway, side	Short (passenger)/ long vehicles
Ma and Grimson [3]	Edge based features, constellation model for classification	Oblique road	Cars/Taxi, Cars/Minivan (94-99%)
Thakoor and Gao [4]	Change detection, hidden Markov models + Shape classification	Side profiles	Car/SUV/Pickup/Minivan (75-94%)
Zhang et al. [5]	Change detection, principal component analysis+support vector machines/ eigen-cars for classification	Intersection	Cars/Pickups/SUV+Vans (45-85%)
Hsieh et al. [6]	Change detection, vehicle size and linearity for classification	Highway, front	Cars/minivan/truck/van truck (∼90%)
Morris and Trivedi [7]	Change detection, classification based on morphological measurements and moments	Highway, side view	Car/SUV/Pickup/Van/Bike/Truck /Semi/Merged (∼78%)
Kafai and Bhanu [8]	Change detection, locations and dimensions of landmarks, hybrid dynamic Bayesian network	Rear view	Car/SUV/Pickup/Minivan (97%)



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

Fig. 1. (Top) Canonical structures for different vehicles. (Bottom) Rear views (black line: parallel to the road; white line: almost perpendicular to the road). (a) Sedan. (b) Pickup. (c) Minivan. (d) SUV.

almost parallel or perpendicular to the road. As seen in the top row in Fig. 1, these surface orientations alone are discriminative enough to separate sedans, pickups, and minivans/SUVs. We call these surfaces with different orientations *structural signatures*. The challenge is to capture these surface orientations reliably with minimal computational effort.

From the rear view, the structure of a vehicle can be characterized along any vertical axis. The structure is almost the same around the center of a vehicle image. We choose to encode the structural signature along the axis of bilateral symmetry of a vehicle due to the reasons given below.

1) All types of passenger vehicles exhibit strong bilateral symmetry from the rear.
2) The axis of symmetry is robust to partial detection of a vehicle as well as to spurious regions, such as shadows being detected from the video as a part of the vehicle.
3) It is robust to illumination variation, body color change, image resolution change, etc. Thus, the axis of symmetry can be detected consistently and reliably.

Note that vehicle symmetry has been used in the past for vehicle detection [14]–[17]. However, we establish symmetry of the region of interest (ROI) instead of the entire image, reducing the computational burden.

III. TECHNICAL APPROACH

By analyzing motion of an object with time, its structure can be recovered. While this is the general principle of structure-from-motion approaches, we would like to reduce the complexity of the solution by imposing additional constraints of our problem.

A. Principle of the Technique

In a typical tolling or traffic monitoring scenario, a video camera is mounted on the top of a lane, which observes vehicles as they move away from the camera. In this scenario, we can safely assume that the velocity of vehicles is negligible in the direction perpendicular to the lane and almost constant along the lane. We will characterize geometry of this scene in a parallel projection where the projection plane is orthogonal to the road plane as well as the camera image plane. Under this projection, the road and image planes appear as lines.

As the vehicle surfaces hold a constant relationship with the road independent of the camera, we choose to analyze the surfaces with respect to their road projection instead of their image projections.

Let an object on a plane be imaged by a camera. The object is moving away from the camera. The motion of the object is negligible along the direction of the horizontal scanlines of the image. This adequately describes a vehicle moving on a road that is being captured from the rear by a camera mounted above the road. For simplicity, the object is assumed to be a cuboid. The side view of this object at time instance t is shown in Fig. 2. We analyze the horizontal and vertical faces of the cuboid, which are represented by P_1P_2 and P_1P_3, respectively.

Theorem 1: The height of the projection of the surface parallel to the road does not change with time.

Let the projection of the surface P_1P_2 at time t be $\alpha(t)d$. Since $\triangle ODP_1(t) \sim \triangle OP_0P_1'(t)$, $\triangle ODP_2(t) \sim \triangle OP_0P_2'(t)$. Using properties of similar triangles

$$\frac{h-\delta}{h} = \frac{DP_1(t)}{P_0P_1'(t)} = \frac{DP_1(t)+d}{P_0P_1'(t)+\alpha(t)d} = \frac{1}{\alpha(t)}. \quad (1)$$

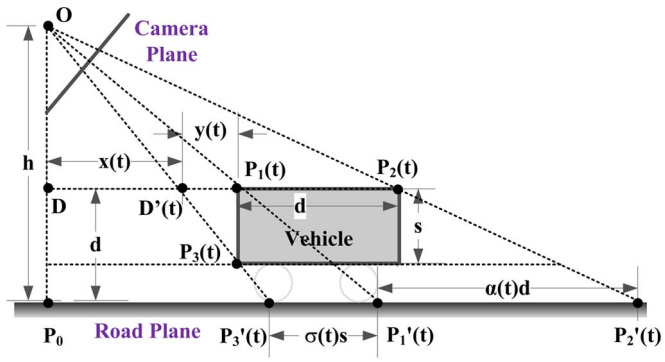


Fig. 2. Scene geometry of projection on the road.

Fig. 3. System for vehicle classification.

As $\alpha(t)$ is a constant, the height of the projection of the surface parallel to the road does not change with time. □

Theorem 2: The height of the projection of the surface perpendicular to the road changes with time.

Let the projection of the vertical line P_1P_3 at time t be $\sigma(t)s$. Since $\Delta ODP_1(t) \sim \Delta OP_0P_1'(t)$, $\Delta ODD'(t) \sim \Delta OP_0P_3'(t)$, we get

$$\frac{P_0P_3'(t)}{\sigma(t)s} = \frac{x(t)}{y(t)}. \tag{2}$$

As $\Delta ODD'(t) \sim \Delta P_3(t)P_1(t)D'(t)$

$$\frac{x(t)}{y(t)} = \frac{h-\delta}{s}. \tag{3}$$

From (2) and (3), $\sigma(t) = P_0P_3'(t)/(h-\delta)$. As $\sigma(t)$ changes with time, the height of the projection of the surface perpendicular to the road changes with time. □

Based on the properties of vertical and horizontal surfaces, we develop an approach to compute the structural signatures of vehicles, which are used in the vehicle classification system shown in Fig. 3.

B. Road-to-Camera Mapping

To find the vehicle surface projection on the road, a relationship between the camera image plane and the road plane has to be established. This relationship is a homography [18]. Homography H can be estimated with a minimum of four-point correspondence, which can be easily established with existing standardized pavement markings such as lane markings, as shown in Fig. 4.

C. Bilateral Symmetry Detection

Before the axis of bilateral symmetry of a vehicle is established, the vehicle is detected from the video with a moving-object detection approach from [19]. For each ROI selected by moving-object detection, the bilateral axis of symmetry is established. The axis is established through a voting scheme.

Given the orientation of the ROI, the axis of symmetry is assumed to be vertical, i.e., it corresponds to one of the ROI columns. To estimate the axis of symmetry, we first estimate edge magnitudes and the orientation of the image using Gabor filters. To avoid texture edges, we apply surround suppression [20] to the Gabor response. We carry out nonmaximal suppression to get the final set of edges.

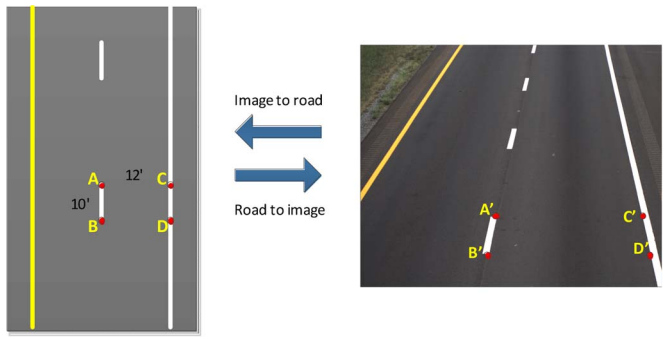


Fig. 4. Estimating road-to-image homography from lane markings.

For a candidate axis location corresponding to the jth column of ROI R with edge magnitude E and its quantized orientation O, the votes are counted as

$$V(j) = \sum_{\forall i,j^-,j^+:(i,j^+)\in R,(i,j^-)\in R} v(i,j^-,j^+) \tag{4}$$

where

$$v(i,j^-,j^+) = \begin{cases} \min(E_{i,j^-}, E_{i,j^+}), & O_{i,j^-} = O'_{i,j^+} \\ 0, & \text{otherwise.} \end{cases}$$

Additionally, $j^+ = j + \Delta$, $j^- = j - \Delta$, and $O'_{i,j} = \pi - O_{i,j}$. For the candidate axis location j, Δ takes values from 1 to $\min(j, \text{width(ROI)} - j)$. The axis of symmetry is assigned to the column with the highest number of votes. A small rectangular *template* around the axis of symmetry is selected, and it is tracked in the subsequent frames using template matching.

D. Multiframe Matching

To establish the structural signature between two frames i and j, only the templates from i to j are needed to be analyzed. First, a row-to-row correspondence between the templates is established. This can be achieved in two ways: 1) by matching the template pairs of adjacent frames and then propagating the matches and 2) by performing multiframe matching on all the templates in a single operation.

Multiframe matching offers various advantages over frame-to-frame matching as follows.

1) Multiframe matching provides more constraints on the matching as multiple frames are involved.
2) The matching can recover in subsequent frames after failure in a certain frame.
3) As errors and failures do not propagate, matches generated after each frame are independent.
4) Due to this independence, decisions such as classification after each frame can be combined to improve their accuracy.

We establish the following property, which can be used as a constraint for multiframe matching.

Theorem 3: For a vehicle moving at a constant speed, under the 1-D image projection centered at projection of the line at the

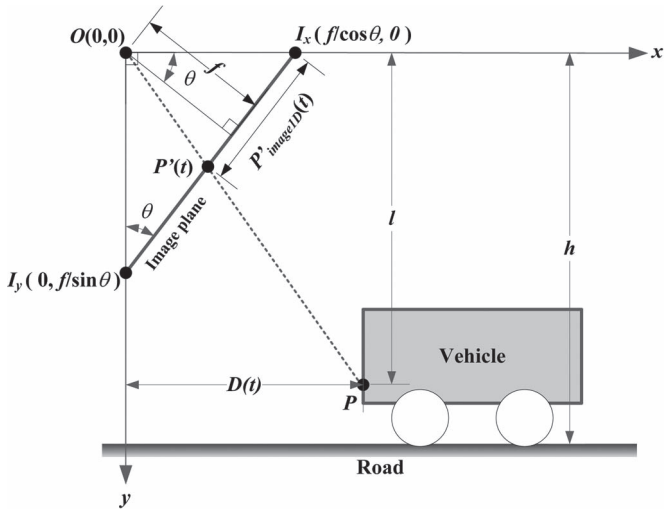


Fig. 5. Scene geometry of projection on the camera image plane.

infinity, the inverse of the projection of a feature on the vehicle linearly varies with time.

Fig. 5 shows the scene geometry. The optical center of the camera is located at O. We treat this as the origin of the 2-D camera coordinate system with the positive x-axis pointing right and parallel to the road and the positive y-axis pointing downward. The height of the camera from the road is h. The focal length of the camera is f, and the depression angle with respect to the x-axis is θ. The image plane of the camera intersects with x- and y-axes at $I_x \equiv (f/\cos\theta, 0)$ and $I_y \equiv (0, f/\sin\theta)$, respectively.

Consider point P on a vehicle visible from the camera. At time t, let the coordinates of P be $(D(t), l)$. At time t, point P is projected at $P'(t)$, which lies at the intersection of line $I_x I_y$ and line OP. Solving for this point yields

$$P'_{\text{camera2D}}(t) = \left(\frac{fD(t)}{D(t)\cos\theta + l\sin\theta}, \frac{lf}{D(t)\cos\theta + l\sin\theta} \right).$$

From the above, the projection of the line at the infinity is

$$\lim_{D(t)\to\infty} P'_{\text{camera2D}}(t) = \left(\frac{f}{cos\theta}, 0 \right) \equiv I_x.$$

In the 1-D image coordinate system with I_x as the origin, the projection is given by the distance between I_x and $P'(t)$, i.e.,

$$P'_{\text{image1D}}(t) = \frac{lf}{\cos\theta\,(D(t)\cos\theta + l\sin\theta)}.$$

Taking the inverse of the projection and differentiating with time t, we get

$$\frac{d}{dt}\frac{1}{P'_{\text{image1D}}(t)} = \frac{\cos^2\theta}{lf}\frac{d}{dt}D(t).$$

For a vehicle traveling at constant velocity ν, the above expression turns into a constant K. Thus

$$\frac{d}{dt}\frac{1}{P'_{\text{image1D}}(t)} = \frac{\nu\cos^2\theta}{lf} = K. \tag{5}$$

Thus, the inverse of $P'_{\text{image1D}}(t)$ linearly varies with time. □

The real 2-D image coordinates can be easily converted to the 1-D side-view image coordinates. First, the 1-D side-view image coordinate system is aligned along the y-axes of the 2-D image coordinates. Second, as shown in Fig. 5, the origin I_x of the 1-D side-view image coordinate system is located on the projection of the line at the infinity. Thus

$$P'_{\text{image1D}} = P'_{\text{image2D}}(y) - P_\infty$$

where P_∞ is the y-coordinate of the image projection of the line at the infinity.

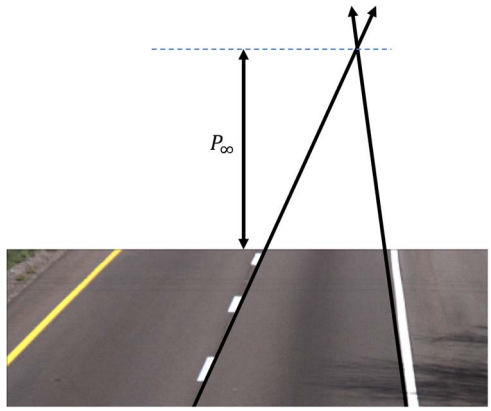


Fig. 6. Finding P_∞ with the lane markings and the vanishing point.

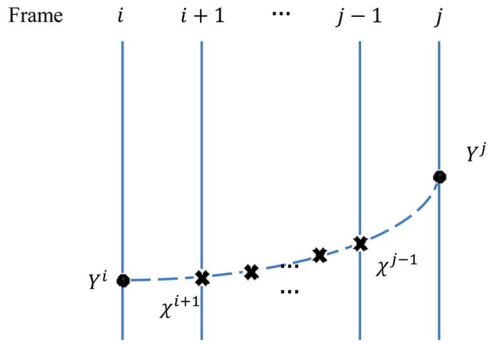


Fig. 7. Constrained multiframe matching.

Fig. 6 illustrates how the parallel lane markings and their vanishing point can be used to find P_∞. Generally, P_∞ lies outside the image and has a negative value. In the rest of this section, we will use the 1-D side-view image coordinates.

Based on Theorem 3, we develop a constrained multiframe matching approach. Consider frames i and j, where $j - i > 1$. To match 1-D image location Y^i in frame i with 1-D image location Y^j in frame j, a frame-to-frame matching cost can be written as

$$C_{FF}(Y^i, Y^j) = g\left(I^i_{\mathcal{N}}(Y^i), I^j_{\mathcal{N}}(Y^j)\right)$$

where $I_{\mathcal{N}}$ indicates neighborhood image intensities, and $g(\cdot)$ is a similarity function.

However, if Y^i and Y^j are matched, then they must follow (5) and should also match certain locations in frames $i+1$ to $j-1$, as shown in Fig. 7. We denote these constrained locations by $\chi^{i+1}, \chi^{i+2}, \ldots, \chi^{j-1}$. We define a multiframe matching cost as

$$C_{MF}(Y^i, Y^j) = g\Big(I^i_{\mathcal{N}}(Y^i), I^{i+1}_{\mathcal{N}}(\chi^{i+1}), \ldots, I^{j-1}_{\mathcal{N}}(\chi^{j-1}), I^j_{\mathcal{N}}(Y^j)\Big) \tag{6}$$

where intermediate locations χ^k's are given by the discrete time version of (5) as

$$\chi^k = \left(\frac{1}{Y^i} + \frac{k-i}{j-i}\left(\frac{1}{Y^j} - \frac{1}{Y^i}\right)\right)^{-1}. \tag{7}$$

The goal of multiframe matching is to establish correspondence between rows of templates T^i and T^j. In terms of the rows

$$T^i = \left[r_1^i, r_2^i, \ldots, r_N^i\right]$$
$$T^j = \left[r_1^j, r_2^j, \ldots, r_M^j\right]$$

where r_l^k indicates the lth row of the window from the kth frame. This correspondence problem can be solved with dynamic time warping (DTW) [21] by minimizing a matching cost, i.e.,

$$C_p(T^i, T^j) = \sum_{l=1}^{L} c\left(r_{n_l}^i, r_{m_l}^j\right) \tag{8}$$

where p is a warping path [21], where row $r_{n_l}^i$ corresponds with row $r_{m_l}^j$ for $l = 1, 2, \ldots, L$, and c is a row-to-row matching cost.

In our formulation, we use the multiframe cost in (6) as the row-to-row matching cost. As there is one-to-one mapping from 1-D side-view image coordinates to actual image y-coordinates and to the corresponding rows in the selected window, we rewrite (6) in terms of rows as (with a slight abuse of the notation)

$$\begin{aligned} C_{MF}(Y^i, Y^j) &= C_{MF}(y^i - P_\infty, y^j - P_\infty) \\ &= C_{MF}(y^i, y^j) = C_{MF}(r^i, r^j). \end{aligned}$$

Additionally, if the term $I_N^k(Y^k)$ returns row r^k from the window as the neighborhood of y^k, (6) can be further reduced to

$$C_{MF}(r^i, r^j) = g(r^i, r^{i+1}, \ldots, r^j).$$

We define the following similarity function between rows $r^i, r^{i+1}, \ldots, r^j$, which is the sum of element-wise variances

$$g(r^i, r^{i+1}, \ldots, r^j) = |\bar{r^2} - (\bar{r})^2| \tag{9}$$

where

$$\bar{r} = \frac{1}{j-1+1}\sum_{k=i}^{j} r^k$$
$$\bar{r^2} = \frac{1}{j-i+1}\sum_{k=i}^{j} r^k \cdot r^{k^T}.$$

Algorithm 1 shows the process of computation of cost matrix D from (9), which can then be passed to a standard DTW optimization algorithm to find optimal warping path p^*. Thus

$$p^* = \text{DTW}(D).$$

Fig. 8 shows an example of frame-to-frame and multiframe matching. The frames being matched are shown in Fig. 8(a) and (b), with the templates marked by the rectangles. The row-versus-row cost matrix is shown in Fig. 8(c) and (d). The additional constraints on the multiframe cost show up as streaks in the multiframe cost matrix, whereas the frame-to-frame cost matrix only shows a rectangular block structure.

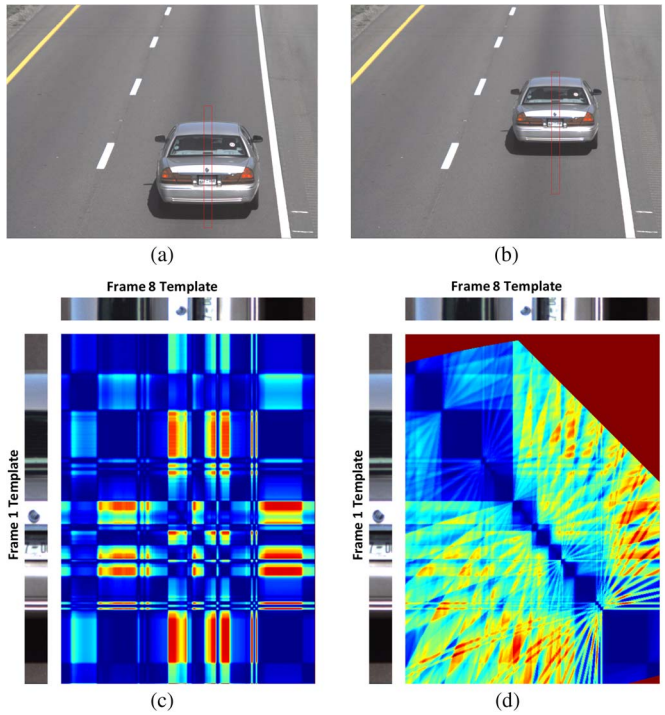


Fig. 8. Multiframe matching. (a) Frame 1. (b) Frame 8. (c) Frame-to-frame cost for frames 1–8. (d) Multiframe cost for frames 1–8. In panels (c) and (d), darker (cooler) colors indicate lower cost, whereas brighter (warmer) colors indicate higher cost.

Algorithm 1: ComputeMultiFrameCostMatrix

Input: $T^i, T^j \leftarrow$ Windows to be matched,
$T^{i+1}, \ldots, T^{j-1} \leftarrow$ Intermediate windows,
$P_\infty \leftarrow$ Projection of the line at infinity.
Output: $D \leftarrow$ Cost matrix.
for $m \leftarrow 0$ **to** $y_b^i - y_t^i$ **do**
  for $n \leftarrow 0$ **to** $y_b^j - y_t^j$ **do**
    $y^i = y_t^i + m$
    $r^i \leftarrow \text{GetRow}(T^i, y_i)$
    $Y^i \leftarrow y^i - P_\infty$
    $y^j = y_t^j + n$
    $r^j \leftarrow \text{GetRow}(T^j, y_j)$
    $Y^j \leftarrow y^j - P_\infty$
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      Find χ^k with (7).
      $y^k = \chi^k + P_\infty$.
      $r^k \leftarrow \text{GetRow}(T^k, y_k)$
    end
    $D(m,n) = g(r^i, r^{i+1}, \ldots, r^j)$
  end
end

The optimal path returned by DTW, i.e., $p^* = \{p_1^*, p_2^*, \ldots, p_L^*\}$, is in terms of the column and row numbers of matrix D, i.e., $p_l^* = (m_l, n_l)$. The path is transformed to 1-D side-view image coordinates as

$$Y_l^i = m_l + y_t^i - P_\infty$$
$$Y_l^j = n_l + y_t^j - P_\infty.$$

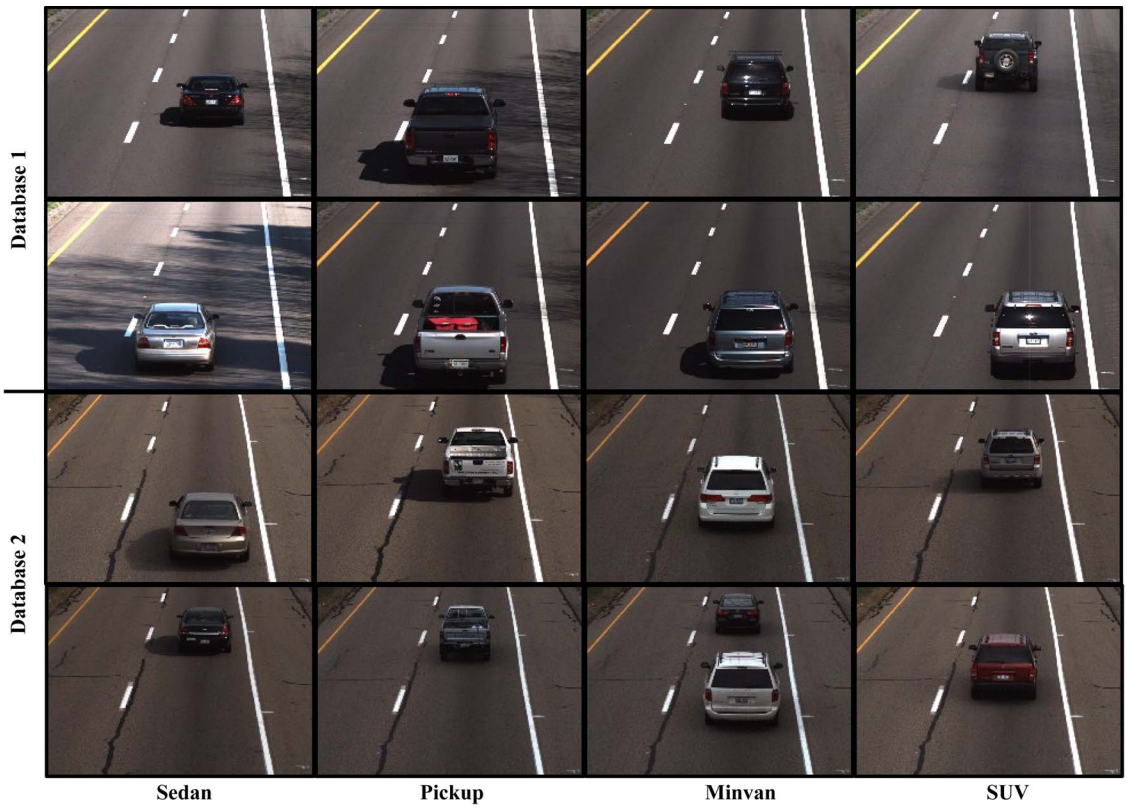


Fig. 9. Example frames from the databases used for different vehicle types.

E. Structural Signature Computation

Before structural signatures for a vehicle can be generated, candidates for vehicle surfaces are generated in the ith frame. Generally, the surface on vehicles is separated by strong horizontal edges. This property can be used to split the vehicle in N *surface elements*. Alternatively, this can be achieved by splitting the template into N *surface elements* of equal heights. Let the rows corresponding to the edges separating the surface elements be $E_1, E_2, \ldots, E_N, E_{N+1}$. Each adjacent pair of edges (E_n, E_{n+1}) forms a surface element $S_{n,n+1}$. The projection of this surface element on the road is $P_{n,n+1}$, which is computed using the camera-to-road homography. For a pair of frames (i, j), the structural signature can be computed as

$$S_{i,j} = \left(\frac{P^i_{1,2} - P^j_{1,2}}{P^i_{1,2}}, \ldots, \frac{P^i_{N-1,N} - P^j_{N-1,N}}{P^i_{N-1,N}} \right) \quad (10)$$

which represents the normalized change in the height of surface projections.

Quantities in the above equation can be found using edge locations, row-to-row mapping from frame i to frame j, and camera-to-road homography H.

IV. Experimental Results

A. Video Data and Calibration

The proposed system was validated with two video data sets. In the first data set, videos were recorded at a freeway location over several days during daytime. The data set contains 778 examples. For the second data set, videos were recorded at two freeway locations over several days during daytime, and it contains 886 vehicles. For both data sets, the camera was set up on top of a freeway lane at the height of 22 ft with depression angles of 8°–10°, capturing more than 200 ft of the lane. All videos were captured at the 1600 × 1200 resolution at 12 fps, allowing for more than 15 frames with complete view of vehicles traveling at freeway speeds (∼60 mi/h). Fig. 9 shows example frames from these videos. Table II shows the distribution of different vehicle types for both databases.

TABLE II
Description of Databases Used

Class	Database 1	Database 2	Combined
Sedan	266	315	581
Pickup	157	191	348
Minivan/SUV	355	380	735
Total	778	886	1664

For each camera view in the videos, a camera-to-road homography was established by detecting lane separation markings. These markings are typically a 10 ft white strip followed by a 30 ft gap. Using the lane width of 12 ft, two additional points were located on the solid white lane to estimate the homography.

B. Symmetry Detection

Vehicles were detected at the 400 × 300 resolution with a moving-object detection technique using a combination of frame difference and optical flow [19]. This motion-based technique can fail when two vehicles follow each other very closely as they enter the camera view by detecting them as

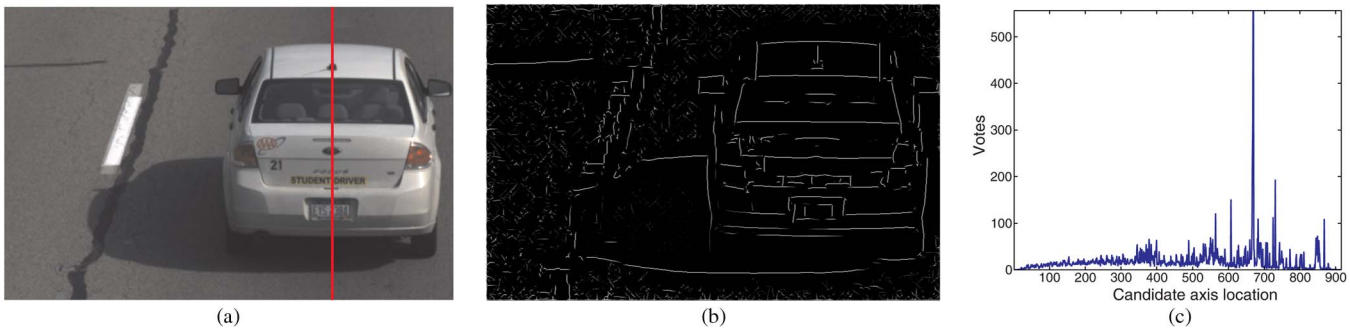


Fig. 10. Symmetry detection. (a) Vehicle ROI with shadow with detected axis of symmetry overlaid. (b) ROI with nonmaximally suppressed edges. (c) Voting.

a single vehicle. As the data were collected on a freeway in our case, this problem did not arise. An appearance-based vehicle detector (such as in [22]) can be used as an alternative when such problem is common. Detected objects that did not lie on the right lane or were not entirely in the frame were discarded. Vehicles on the left lane are discarded as the rear of a vehicle is not completely visible for most of the frames. To process the vehicles in the left lane, either the camera can be moved to cover both lanes or a separate system for the left (or any additional) lane can be deployed. The full resolution ROI was then processed to compute the structural signature. In the very first frame where the vehicle was completely visible, the axis of symmetry was established. The edge orientations and magnitudes to vote for the axis of symmetry were found with Gabor filters with orientations 0, $(\pi/4)$, $(\pi/2)$, and $(3\pi/4)$. Fig. 10(a) shows an example ROI detected by the moving-object detection system. In addition to the vehicle, the ROI includes the strong dark shadow cast by the vehicle with some spurious regions. The detected edges in the ROI are shown in Fig. 10(b). The outcome of the voting process is shown in Fig. 10(c), which clearly shows the location of the axis of symmetry. Fig. 10(a) shows the original ROI overlaid with the axis of symmetry.

C. Structural Signature Computation

The structural signature depends on the number of surface elements chosen. With a subset of database 1, we conducted experiments by varying the number of surface elements. Fig. 11 shows classification accuracy for these experiments. We chose the number of surface elements to be 11 as it shows maximum accuracy as well as the minimal drop-off to the closest alternatives.

The quality of structural signatures depends on the successful tracking of vehicles. We estimated smoothness of the tracking results, and the tracks that deviated too far from the smooth trajectory were identified. About 48% tracks for database 1 and 39% tracks for database 2 were identified as nonsmooth. This suggests that database 1 is more challenging for classification when compared with database 2.

Fig. 12 shows a histogram-like representation of structural signatures. These are computed for data with smooth trajectories alone. The brightness is proportional to the frequency of occurrence. Fig. 12(a) represents sedan class signatures for databases 1 and 2. As expected, values close to zero are observed for vehicle top and trunk top, which are parallel to the road. For other parts of vehicles, nonzero values are more

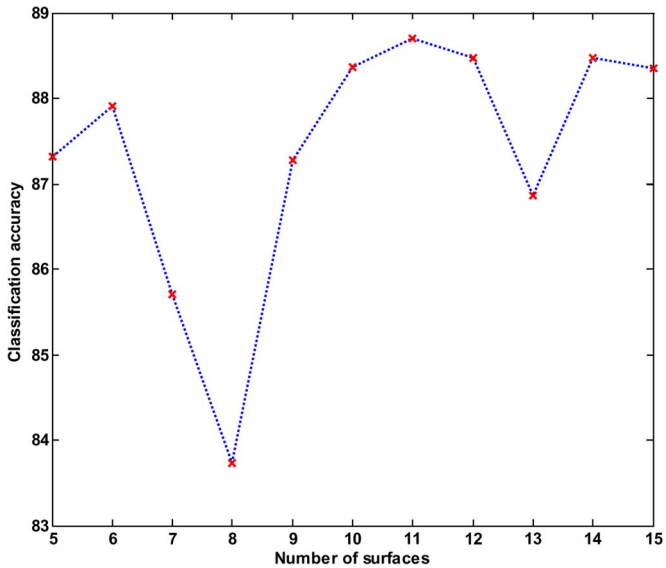


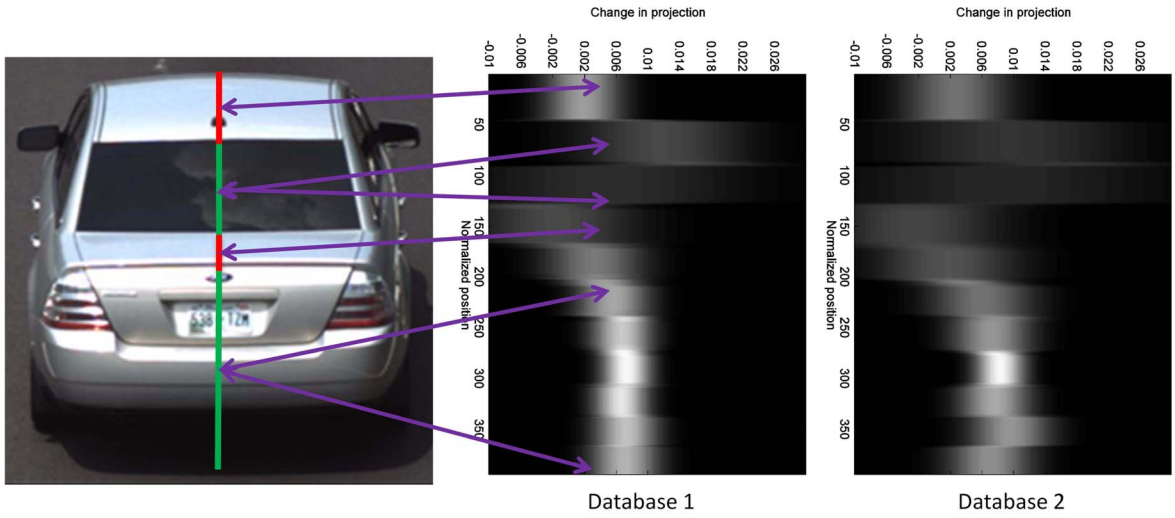
Fig. 11. Classification accuracy versus number of surface elements for the subset of database 1.

frequent in the structural signatures. Thus, structural signatures capture the canonical structure of the vehicles. It can be observed that database 2 shows higher variation compared with database 1. This is expected as database 2 was captured at two locations with different camera settings. Similar trends can be also seen for other classes in Fig. 12(b) and (c).

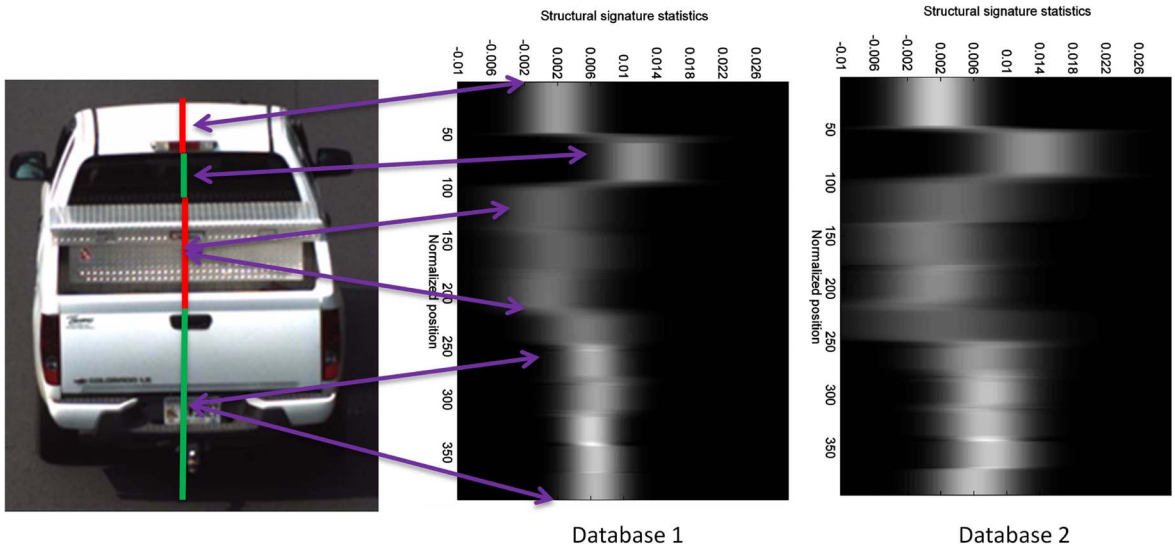
D. Classification

The structural signatures were used to train a support vector machine (SVM) classifier with a radial basis function kernel. We used the Library for Support Vector Machines (LIBSVM) for the implementation of the SVM [23]. Experiments were conducted with databases 1 and 2 and combining both with twofold cross validation to obtain classification accuracy. We also conducted experiments by using database 1 for training and database 2 for testing, and vice versa. Structural signatures were computed using frame-to-frame matching and multiframe matching for comparative purposes. We also establish the baseline performance by using structural signatures directly computed using the image coordinates, i.e., without road projection and normalization.

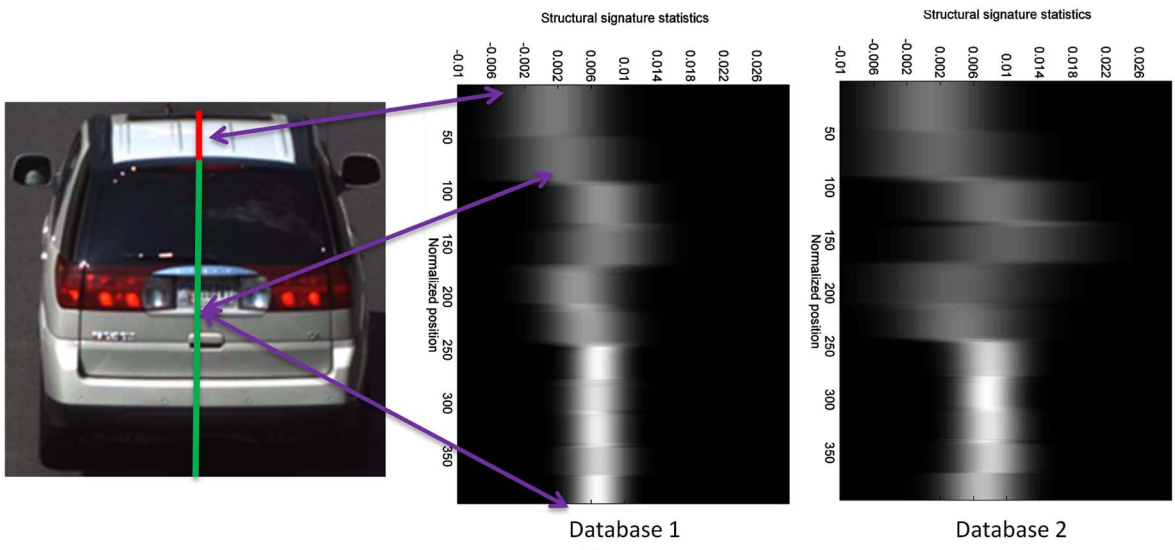
Table III shows classification accuracy when varying numbers of frames are used to compute the structural signatures. When the number of frames used is low, the amount of



(a)



(b)



(c)

Fig. 12. Structural signatures. The first column shows a representative image overlaid with a canonical structure pattern, where red indicates surfaces parallel to the road, and green indicates other surfaces. The second and third columns show a histogram-like representation of structural signatures for databases 1 and 2, respectively. Brightness is proportional to the frequency of occurrence. Arrows between the first and second columns show the correspondence between vehicle surfaces and structural signatures. Rows: (a) sedans; (b) pickups; (c) minivans/SUVs.

TABLE III
CLASSIFICATION ACCURACY

Frames used	2	3	4	5	6	7	8	9	10	11
Multiframe DB1	77.63	80.33	82.13	83.80	84.45	**85.60**	84.32	84.19	82.39	83.16
Frame to frame DB1	20.18	20.18	83.80	**85.99**	85.09	85.35	85.60	85.86	84.45	84.83
Trained with DB2 & Tested with DB1	73.25	77.31	78.56	78.33	81.26	81.26	80.59	**82.17**	81.26	80.36
Baseline DB1	63.88	71.21	74.55	74.68	**75.45**	74.68	73.91	72.62	71.47	72.24
Multiframe DB2	74.49	83.63	84.76	86.79	85.55	86.46	86.57	**87.25**	86.79	86.23
Frame to frame DB2	67.38	84.31	85.21	85.21	84.20	85.21	85.89	**86.00**	86.00	85.89
Trained with DB1 & Tested with DB2	73.26	77.89	79.56	78.53	**83.16**	80.85	81.36	79.95	82.39	80.08
Baseline DB2	62.19	74.27	75.85	78.33	77.99	78.44	**78.56**	77.99	78.44	77.09
Multiframe DB1 & DB2 combined	76.74	81.91	83.17	85.34	84.92	85.46	85.10	**85.64**	85.04	84.01
Frame to frame DB1 & DB2 combined	20.91	20.91	84.50	84.19	83.95	85.40	85.10	85.28	85.16	**85.64**
Baseline DB1 & DB2 combined	63.82	72.90	75.54	76.62	75.90	**77.10**	75.84	75.72	75.24	75.12

evidence collected is low, leading to lower quality of signatures and lower classification accuracy. However, with only five frames, the classifier performance levels out. This is possibly due to the tracking errors introduced, which negate the evidence being added. The performance of the classifier for database 1 is lower compared with database 2, which was expected due to the quality of tracking. For database 1, frame-to-frame-matching-based signatures outperform multiframe-matching-based signatures. For database 2 and combined data, multiframe matching outperforms frame-to-frame-matching-based signatures.

For the experiment where database 1 was used for training and database 2 was used for testing, the performance drops more compared with the other-way-around scenario, as database 1 shows lesser variation when compared with database 2. It is also clear that the proposed structural signatures outperform the baseline approach significantly.

E. Fusion of Classifiers

As seen in Table III, the accuracy of the classifier changes with the number of frames used to compute the signatures. This is due to the fact that, depending on the environmental conditions, the tracking performance might vary. As the classification decision can be made after each frame, a fusion scheme that combines these outcomes might be more accurate. We compare performance of decision-level fusion with simple voting and weighted voting for frame-to-frame-matching- and multiframe-matching-based signatures. In simple voting, the most frequent classifier outcome is assigned as the class and ties are considered as a classification failure. In weighted voting, the votes are weighted with training accuracy that eliminates the ties in almost all the cases. Table IV shows classification accuracy after fusion. The fusion classifier outperforms the single-frame classifier for all of our experimental scenarios. The multiframe-signature-based fusion classifier outperforms the frame-to-frame-based classifier for all the databases. This is due to the ability of multiframe matching to recover from failures.

F. Comparison With Other Methods

Vehicle blob properties such as width, height, and area are commonly used for classification of vehicles. Contours have been also used for the classification of vehicle profiles. We compare these classification methods with structural-signature-based classification in Table V. Additionally, we apply edit-distance-based classification to the structural signatures. The blob-based classifier faces difficulty as the scale of the blob varies as the vehicle moves away from the camera. Contour-based classifiers also have limited success as the rear view of a vehicle offers minimal discrimination among the classes. Although edit-distance-based classification uses structural signatures, it fails to deal with variation in the structural signatures.

TABLE IV
CLASSIFICATION ACCURACY WITH DIFFERENT FUSION SCHEMES

	Single best (Frame used)	Fusion (Frames fused)	Weighted (Frames fused)
Multiframe DB1	85.60 (7)	87.53 (2-11)	88.17 (2-11)
Frame to frame DB1	85.99 (5)	86.35 (4-11)	86.76 (3-11)
Multiframe DB2	87.25 (9)	88.71 (3-11)	89.05 (2-11)
Frame to frame DB2	86.00 (9)	87.92 (3-11)	88.04 (3-11)
Multiframe DB1 & DB2 combined	85.64 (9)	87.5 (3-11)	87.8 (2-11)
Frame to frame DB1 & DB2 combined	85.64 (11)	86.48 (5-11)	86.78 (5-11)

TABLE V
COMPARISON WITH OTHER METHODS

Method	DB1	DB2	DB1 & DB2 combined
Blob features	54.65	41.00	45.80
Contour curvature	49.61	45.49	47.96
Structural Signatures + Edit distance	65.7	61.29	63.46
Structural Signatures + SVM (Single Best Classifier)	85.99	87.25	85.64
Structural Signatures + SVM (Multiclassifier fusion)	88.17	89.05	87.80

TABLE VI
CONFUSION MATRIX

Decision→ Class↓	Sedan	Pickup	Minivan/ SUV	%Accuracy
Sedan	502	10	69	86.4
Pickup	27	273	48	78.45
Minivan/SUV	36	13	686	93.33

G. Discussion

Table VI gives the confusion matrix for the weighted fusion classifier for the combined database. Pickups have the lowest classification accuracy as their beds can carry items that can deform their structural signature, leading to incorrect classification. On the other hand, minivans and SUVs have the simplest structures, and this results in the highest accuracy. Additionally, some sedans such as hatchbacks and some pickup trucks that carry camper shells have structural signatures similar to minivans/SUVs, which result in misclassifications.

V. Conclusion

We have presented structural-signature features for the classification of rear-view videos of vehicles. The approach used information from multiple video frames to infer the vehicle structure. This is unlike the state-of-the-art approaches, which use either blob features or appearance features from frame to frame.

The structural signatures are independent of the appearance, which makes them less susceptible to illumination changes and imaging system variations. Use of the road projection allows significant variations in camera angles. Incorporating symmetry makes our system robust against shadows, partial detections, and occlusions. The proposed system uses computationally inexpensive techniques, such as change detection, edge-voting-based symmetry, and template tracking to realize the structural signatures.

Our OpenCV-based C ++ implementation of the classification system runs at 20 fps on a computer with Intel Core i7-2600 central processing unit at 3.4 GHz. Further optimizations will make real-time implementation viable on general-purpose computation platforms with graphics processing units and digital signal processors.

While sedans, pickups, minivans, and SUVs form the majority of the vehicles on the road, including the heavy good vehicles, buses, and motorcycles in the classifier would give a complete classification solution. In the future, we plan to amend the structural signature classifier with a size-based classifier to achieve this.

References

[1] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 37–47, Mar. 2002.

[2] R. Avery, Y. Wang, and G. Scott Rutherford, "Length-based vehicle classification using images from uncalibrated video cameras," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2004, pp. 737–742.

[3] X. Ma and W. Grimson, "Edge-based rich representation for vehicle classification," in *Proc. IEEE ICCV*, 2005, pp. 1185–1192.

[4] N. Thakoor and J. Gao, "Automatic video object shape extraction and its classification with camera in motion," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, vol. 3, pp. III-437–III-440.

[5] C. Zhang, X. Chen, and W. B. Chen, "A PCA-based vehicle classification framework," in *Proc. 22nd Int. Conf. Data Eng. Workshops*, 2006, p. 17.

[6] J.-W. Hsieh, S.-H. Yu, Y.-S. Chen, and W.-F. Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 2, pp. 175–187, Jun. 2006.

[7] B. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 425–437, Sep. 2008.

[8] M. Kafai and B. Bhanu, "Dynamic Bayesian networks for vehicle classification in video," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 100–109, Feb. 2012.

[9] V. S. Petrovic and T. F. Cootes, "Analysis of features for rigid structure vehicle type recognition," in *Proc. BMVC*, 2004, pp. 587–596.

[10] P. Negri, X. Clady, M. Milgram, and R. Poulenard, "An oriented-contour point based voting algorithm for vehicle type classification," in *Proc. ICPR*, 2006, pp. 574–577.

[11] G. Pearce and N. Pears, "Automatic make and model recognition from frontal images of cars," in *Proc. IEEE AVSS*, 2011, pp. 373–378.

[12] N. Ghosh and B. Bhanu, "Incremental unsupervised three-dimensional vehicle model learning from video," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 423–440, Jun. 2010.

[13] N. Thakoor and B. Bhanu, "Structural signatures for passenger vehicle classification in video," in *Proc. ICPR*, 2012, pp. 926–929.

[14] A. Kuehnle, "Symmetry-based vehicle location for AHS," in *Proc. SPIE*, vol. 2902, *Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, A. C. Chachich and M. J. de Vries, Eds., 1997, no. 19, pp. 19–27.

[15] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G. Toulminet, "A cooperative approach to vision-based vehicle detection," in *Proc. IEEE Intell. Transp. Syst.*, 2001, pp. 207–212.

[16] A. Broggi, P. Cerri, and P. Antonello, "Multi-resolution vehicle detection using artificial vision," in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 310–314.

[17] J. Arrospide, L. Salgado, and J. Marinas, "HOG-like gradient-based descriptor for visual vehicle detection," in *Proc. IEEE IV Symp.*, 2012, pp. 223–228.

[18] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[19] N. Thakoor and J. X. Gao, "Automatic video object extraction with camera in motion," *Int. J. Image Graph.*, vol. 8, no. 4, pp. 573–600, Oct. 2008.

[20] C. Grigorescu, N. Petkov, and M. Westenberg, "Contour detection based on nonclassical receptive field inhibition," *IEEE Trans. Image Process.*, vol. 12, no. 7, pp. 729–739, Jul. 2003.

[21] M. Müuller, "Dynamic time warping," in *Information Retrieval for Music and Motion*. New York, NY, USA: Springer-Verlag, 2007, pp. 69–84.

[22] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *Proc. IEEE ITSC*, 2012, pp. 975–982.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, Apr. 2011.

Ninad S. Thakoor (S'04–M'10) received the B.E. degree in electronics and telecommunication engineering from the University of Mumbai, Mumbai, India, in 2001 and the M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Arlington, TX, USA, in 2004 and 2009, respectively.

He is a Postdoctoral Researcher with the Center for Research in Intelligent Systems, University of California, Riverside, CA, USA. His research interests include vehicle recognition, stereo disparity segmentation, and structure-and-motion segmentation.

Bir Bhanu (S'72–M'82–SM'87–F'95) received the S.M. and E.E. degrees in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA, USA; the M.B.A. degree from the University of California, Irvine, CA, USA; and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA.

He is a Distinguished Professor of electrical engineering and the Founding Director of the interdisciplinary Center for Research in Intelligent Systems and the Visualization and Intelligent Systems Laboratory (VISLab), University of California, Riverside (UCR), CA. In addition, he serves as the Director of the National Science Foundation graduate research and training program in video bioinformatics with UCR. He was the Founding Professor of electrical engineering with UCR and served as its first Chair (1991–1994). He has been the cooperative Professor of computer science and engineering (since 1991), of bioengineering (since 2006), of mechanical engineering (since 2008), and the Director of VISLab (since 1991). Prior to joining UCR in 1991, he was a Senior Honeywell Fellow with Honeywell Inc. His research interests include computer vision, pattern recognition and data mining, machine learning, artificial intelligence, image processing, image and video database, graphics and visualization, robotics, human–computer interactions, and biological, medical, military and intelligence applications.

Dr. Bhanu is a Fellow of the American Association for the Advancement of Science, the International Association for Pattern Recognition (IAPR), and the International Society for Optics and Photonics (SPIE).