Counting and Classification of Highway Vehicles by Regression Analysis

Mingpei Liang, Xinyu Huang, Chung-Hao Chen, Xin Chen, and Alade Tokuta

***Abstract*—In this paper, we describe a novel algorithm that counts and classifies highway vehicles based on regression analysis. This algorithm requires no explicit segmentation or tracking of individual vehicles, which is usually an important part of many existing algorithms. Therefore, this algorithm is particularly useful when there are severe occlusions or vehicle resolution is low, in which extracted features are highly unreliable. There are mainly two contributions in our proposed algorithm. First, a warping method is developed to detect the foreground segments that contain unclassified vehicles. The common used modeling and tracking (e.g., Kalman filtering) of individual vehicles are not required. In order to reduce vehicle distortion caused by the foreshortening effect, a nonuniform mesh grid and a projective transformation are estimated and applied during the warping process. Second, we extract a set of low-level features for each foreground segment and develop a cascaded regression approach to count and classify vehicles directly, which has not been used in the area of intelligent transportation systems. Three different regressors are designed and evaluated. Experiments show that our regression-based algorithm is accurate and robust for poor quality videos, from which many existing algorithms could fail to extract reliable features.**

***Index Terms*—Highway vehicle, image warping, cascaded regression.**

I. Introduction

VIDEO cameras could be used to record the traffic information constantly or continuously. We are thus able to analyze the traffic videos in real time and discover any information of interest. One fundamental task is to count the vehicles passing by in a given time period and classify the vehicles into different categories at the same time. The counting and classification results could be useful in many different applications. For example, they could be used to measure traffic density, traffic flow, and even emissions in terms of pollutants and greenhouse gases.

Counting and classification also could be done by other sensors such as radar, infrared, and inductive loop detectors. Although some sensors could be more accurate, they could also be intrusive and need a higher maintenance cost. For example, we may need to embed weighing sensors in road to measure vehicle weight and classify vehicle size. Comparing with other sensors, vision-based systems could be non-intrusive and could obtain much richer traffic information. However, current vision-based systems could be less accurate and more sensitive to operating conditions (e.g., weather). These problems make vision-based systems challenging and important research topics in the area of intelligent transportation systems.

A typical vision-based traffic analysis system could consist of many components such as foreground segmentation, shadow removal, feature extraction, and tracking [1]. In order to count and classify vehicles, there is often a module to detect and separate individual vehicles for each foreground segment. This module could be conducted after feature extraction or tracking. For example, if feature points could be extracted robustly across multiple image frames, it is possible to fit explicit 2D/3-D vehicle models [2], [3]. This kind of algorithms usually requires at least moderate-resolution images without severe occlusions and motion blur. In this paper, we would like to process low-quality videos by skipping this module. In our collected videos, multiple vehicles could be occluded and thus form a large foreground segment. Separation or inference of individual vehicles would be a difficult task in this case. Moreover, a 2-D vehicle shape could be strongly distorted caused by the foreshortening effect, which means the weak perspective projection used in the traditional algorithms is not a good approximation. As the video frame rate could be as low as one frame per second, the vehicle size could be reduced to less than 10×10 pixels for the next image frame. Therefore, it would also be difficult to detect and track robust feature points or edges. Fig. 1 shows several image frames in our collected videos.

There are mainly two contributions of the proposed algorithm. First, we develop a warping method to detect image foreground segments that contain unclassified vehicles. In order to reduce vehicle distortion between two consecutive image frames, we estimate a non-uniform mesh grid and a projective transformation. By using this warping method, we do not need the common used tracking or modeling of individual vehicles. We either do not need to assume the weak perspective projection that has been used in some existing algorithms (e.g., [2], [4]). In fact, the warping method could be considered as an approximation of the perspective projection when vehicle features are not reliable. Furthermore, the warping method based on the non-uniform mesh grid is designed for both curved (as shown in Figs. 1(c) and 7) and straight highway sections. To our knowledge, there are no other similar algorithms for

Manuscript received May 25, 2014; revised August 23, 2014, December 7, 2014, and March 23, 2015; accepted April 17, 2015. Date of publication May 1, 2015; date of current version September 25, 2015. The Associate Editor for this paper was H. Huang.

M. Liang, X. Huang, and A. Tokuta are with the Department of Mathematics and Computer Science, North Carolina Central University, Durham, NC 27707 USA (e-mail: mliang@nccu.edu; xinyu.huang@nccu.edu; atokuta@nccu.edu).

C.-H. Chen is with the Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529 USA (e-mail: cxchen@odu.edu).

X. Chen is with HERE, Chicago, IL 60606 USA (e-mail: xin.5.chen@here.com).

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.2015.2424917

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

Fig. 1. Examples of image frames. (a) and (b) show the severe occlusions. (c) and (d) are two consecutive image frames showing strong shape distortion caused by the foreshortening effect. Feature points or edges are often unreliable in these low resolution images.

the detection of unclassified vehicles. Secondly, we propose a cascaded regression approach to count and classify vehicles directly. A set of low-level features are extracted and form an input vector for the regression. Three different regressors that include Gaussian process regression, standard Poisson regression, and Bayesian Poisson regression are designed and evaluated.

Notice that other components of a typical traffic analysis system, such as shadow removal and foreground segmentation, are not the focus of this paper. These components could be useful for adverse weather conditions. Therefore, the counting and classification performance could be further improved when advanced algorithms for these components are added to the system. Our paper is organized as follows. Section II provides a review of related work. Our algorithm is described in details in Section III. Section IV shows the experimental results and Section V gives a conclusion.

II. Related Work

First, we provide a review of different feature extraction methods for traffic analysis. In [5], a set of Scale Invariant Feature Transform (SIFT) features [6] are extracted and matched in the follow-up image frames in order to improve tracking performance. The SIFT features are also detected, tracked, clustered in the foreground blobs in [7]. Horizontal and vertical line features are extracted in [8] to build a 3-D vehicle model assuming the vehicle is not occluded. Similarly, by predicting and matching image intensity edges, Leotta *et al.* [9] fit a generic 3-D vehicle model to multiple still images. Simultaneous tracking can also be done during the shape estimation in a video. Ma *et al.* [10] proposed a vehicle classification algorithm that uses the feature based on edge points and modified SIFT descriptors. Two classification tasks, cars versus minivans and sedans versus taxies, are tested with good performance. In [11], a 3-D extended Histograms of Oriented Gradients (HOG) feature for detection and classification of individual vehicles and pedestrians is proposed by combining 3-D interest points and HOG. 3-D vehicle models are pre-reconstructed by the methods in [12].

Region-based features are often used for traffic analysis. Gupte *et al.* [13] track regions in image frames by matching, splitting, and merging of these regions. The foreshortening effects between two consecutive image frames are not considered. In [14], assuming individual vehicles have been separated after lane and shadow detection, Hsieh *et al.* further extracted region size and vehicle "linearity" to classify vehicles into four categories (e.g., car, minivan, truck, and van truck). In [15], image regions are extracted according to high edge density areas. Shadows, symmetry measurement, and Harris corners are then used in the hypothesis classification. In [2], image regions of interest are extracted based on motion detection, then a 3-D model is fitted to the image region using a point-to-line segment distance metric.

Occlusion is a major challenging problem in the vehicle segmentation. Many methods have been proposed to deal with this problem. Features mentioned above could be considered as a set of "parts" that are tracked and grouped together [16], [17]. When the 2-D/3-D vehicle model could be fitted into image frames, it is also relatively easy to detect occlusions [2], [18]. In [19], a spatiotemporal Markov random field is proposed to detect occluded vehicles at intersections. In [20], a "cutting region" between two occluded vehicles is extracted based on the motion field of consecutive image frames. Similarly, a "cutting line" is estimated in [21] to separate two occluded vehicles based on the analysis of convex shape.

Image warping is not considered as a step or module to detect and track vehicles in [1]. There are some researches using image warping as a pre-processing step to generate a horizontal or vertical road segment to facilitate the detection and tracking (e.g., [7], [22]). Four reference points are selected to estimate a projective transformation in [7]. This transformation is applied so that all motion vectors are approximately parallel to each other. The similar idea is applied in [22] so that lanes could be detected easily. However, image warping itself has not been applied directly to detect unclassified vehicles. Moreover, only four reference points used in these algorithms are often not enough to model non-straight road segments. In our algorithm, we estimate a nonlinear mesh grid to model road segments more accurately.

The features used in many existing algorithms could be robust when image quality is ideal. However, these features based on point, line, and image region could be highly unreliable with a low image quality. Therefore, it could not be a good choice to use them for segmentation and tracking of individual vehicles and even 3-D reconstruction. In this paper, we use a set of low-level features that could be considered as weak classifiers and apply regression to count and classify vehicles directly.

Regression analysis has been applied to count people in [23], which is similar to our proposed algorithm. There are mainly two differences between two algorithms. First, the detection of unclassified vehicles is quite different from the

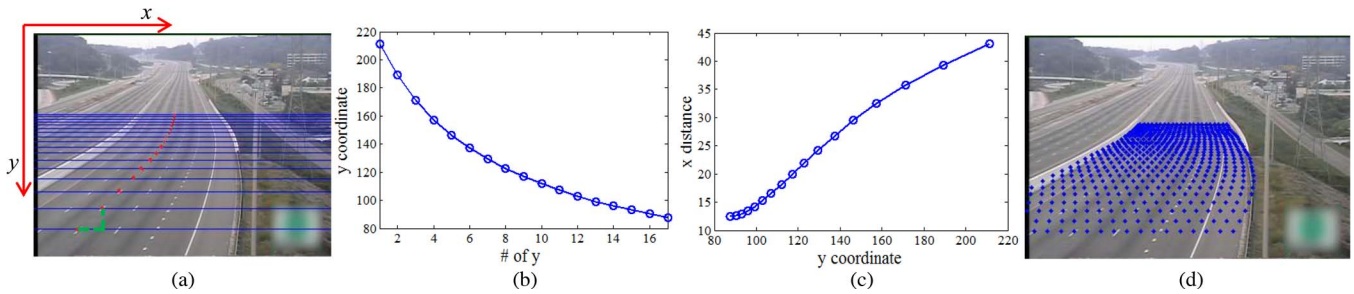

Fig. 2. (a) Red points sampled along traffic trajectory. (b) Fitted spline using vertical positions that are the y coordinates of the sampled points (i.e., horizontal axis is the number of the sampled points, and vertical axis is the corresponding y coordinate). (c) Fitted spline using y coordinates of sampled points (i.e., the vertical axis in (b)) and horizontal distances between consecutive sample points. (d) Generated dense mesh grid.

crowd segmentation described in [23]. The crowd segmentation is done by a mixture of dynamic textures. In the mixture model, the observed variable is the sampled video frames, and the hidden variable encodes the dynamics. Another hidden variable is a mixture component that is used to handle inhomogeneous videos. In the first part of our algorithm, we apply a nonlinear warping algorithm on foreground segments in the previous frame. A projective transformation is applied to reduce perspective distortion. Weighted normalized cross correlation (WNCC) is used to compare transformed patches with the corresponding patches in the current frame. Secondly, the regression frameworks of two algorithms are different. In our algorithm, we build a three-level cascaded regression framework as we have three different vehicle classes. More importantly, to our knowledge, regression analysis has not been applied before for counting and classifying highway vehicles.

III. Algorithm

Our algorithm mainly consists of four steps. The first step is the background estimation and foreground segmentation. Background estimation is a necessary pre-processing step for most vision-based systems. Background could be estimated by a simple averaging [17], [24]. The averaging method has little computational cost, however, it could not be robust to different operating conditions. To improve robustness, the background pixel could also be modeled as a single Gaussian, a mixture of Gaussians, and on [25]–[28]. Recently, Unzueta *et al.* proposed an adaptive multi-cue segmentation strategy to detect foreground pixels [29]. As our major contribution is the novel algorithm for counting and classification, we apply the averaging method to estimate background and use thresholds and morphological operations (e.g., erode and dilate) to extract the foreground segments. A region of interest is also defined and the foreground segments outside the region of interest are removed. The rest steps are described in Section III-A, III-B, and III-C, respectively.

A. Detection of Unclassified Vehicles

It would be difficult to detect and track robust features since vehicle size could be reduced to as small as 10×10. Therefore, it is necessary to consider a much larger image region that could contain more than one vehicle. Since the time between two consecutive image frames is around 1 second, it is safe to assume the relative 3-D distances among vehicles remain almost unchanged or have small changes. However, due to the foreshortening effect, their 2-D projections could be distorted significantly. Hence, the weak perspective projection assumed in [2], [4] is not a good approximation.

We first manually sample a small set of points on the dashed lane markers along the road direction. The vertical positions and horizontal distances between two consecutive points are computed. Fig. 2(a) shows the sampled points (i.e., the red points) on the background image, which partition the highway road into a set of regions based on their y coordinates. Notice that it is easy to automate this sampling step when the traffic trajectory is known. The trajectory can be estimated robustly in [30], [31]. Two sets of values are used to fit two smoothing splines. The spline fitted by vertical positions (i.e., Fig. 2(b)) is a modeling of the foreshortening effect along the road direction. The horizontal axis is the same as the number of sampled points, and the vertical axis is the corresponding y coordinates. As the horizontal direction is roughly parallel to the image plane, the spline fitted by the horizontal distances (i.e., Fig. 2(c)) is a modeling of scaling factors along x-axis. The horizontal axis of this figure is same as the vertical axis in the Fig. 2(b), and the vertical axis is the horizontal distance between two consecutive sample points, which is proportional to the scaling factor. We adopt smoothing splines for fitting as the road direction may not be straight. These two splines are used to generate a dense nonuniform mesh grid.

Our warping method used to detect unclassified vehicles is based on the mesh grid. For simplicity, let us assume the traffic flow is from bottom to top. It is straightforward to extend the warping method to other different traffic flows. First, we check if there are any foreground segments in the bottom region of the previous image frame. This is done by comparing the current image frame with the background image. If the area of foreground segments is larger than the minimum area of a small size vehicle (computed from the training set), then we compute a bounding box that includes these foreground segments. The mesh vertices in the bounding box are the original mesh grid. Secondly, we conduct a complete search from the bottom region to the top region. When we decrease the y coordinate, we can find a new set of vertices that is a shift of the original mesh grid along the highway road. This new set of vertices is the target mesh grid. Warping method we used here is the Thin-Plane

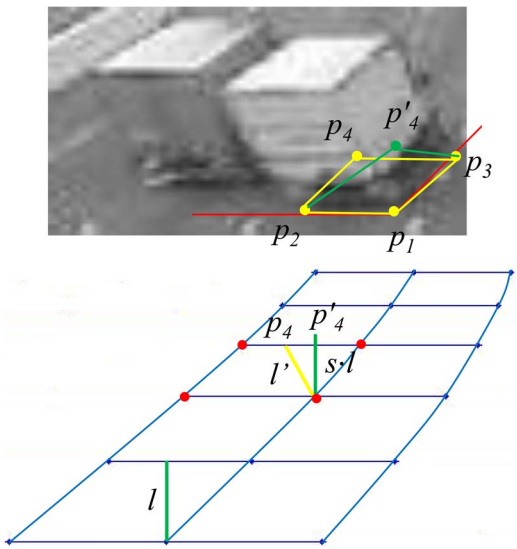

Fig. 3. Selection of four points for the projective transformation.

spline warping in [32]. Texture mapping could also be used to speed up the warping process. Image patch of the foreground mask is also warped. The warped intensity image is compared with the same image patch in the current image frame based on the WNCC,

$$\gamma = \frac{\sum_i w_i (I_p(i) - \bar{I}_p) \cdot (I_c(i) - \bar{I}_c)}{\sqrt{\sum_i w_i (I_p(i) - \bar{I}_p)^2} \cdot \sqrt{\sum_i w_i (I_c(i) - \bar{I}_c)^2}} \quad (1)$$

where $I_p(i)$ and $I_c(i)$ are corresponding pixel values in previous and current image frames, $\bar{I}_p$ and $\bar{I}_c$ are the weighted mean values of pixels in previous and current image frames, and w_i is the pixel weight in the previous image frame. The w_i is set to 1 if it is in the foreground mask and 0 otherwise. The highest matching score is selected and a convex hull is generated based on the foreground segments in the warped mask. The convex hull gives us the image region that has been classified. The foreground segments below the convex hull in the current image frame contain the unclassified vehicles. It is interesting to note that the search results could also be used to estimate the traffic density. If the moving distance along the highway direction is small, it is likely to have a heavy traffic.

Warping is defined by the mesh vertices on the road surface. If the vehicle height is large (e.g., a truck), vehicle shape could still be distorted or tilted after warping. Therefore, we apply another projective transformation to the foreground segment after warping in order to further reduce the distortion. In order to define a projective transformation, we need to define four pairs of points. As shown in Fig. 3, as the traffic flow is from bottom to top, the right and bottom part of the foreground segments are often close to the road surface. Thus, we sample three points, bottom-right p_1, bottom-left p_2, and a rightmost point p_3 at the mean of y coordinates. These three points are the same after the projective transformation.

The fourth points p_4 and p'_4 are selected based on the assumption that vertical lines on the vehicle are often parallel with the image plane of camera. Thus, the vertical lines are still vertical in the image after the perspective projection. We first select a vertical line with length l in the original mesh grid. l would be transformed to l' that is often not vertical after the warping process we described above. Since the warping process is essentially an interpolation of neighboring pixels, we could compute the coordinate of p_4 by the interpolation of four nearby mesh vertices (i.e., the red points in Fig. 3. As the length of the vertical line is $s \cdot l$, p'_4 is also computed. Here, s is the scaling factor that is derived from the spline in Fig. 2(c). Note that this 2-D projective transformation only approximately models the distortions of large vehicles caused by the perspective projection. An exact solution would require camera calibration or 3-D reconstruction. However, this could be time-consuming and needs reliable measurements and feature extraction.

Fig. 4 is an example that illustrates the processes such as warping, projective transformation, weighted normalized correlation, and detection of current unclassified vehicles.

B. Feature Extraction

As mentioned, it could be difficult to detect and track reliable features in low quality image frames. Therefore, we only use a set of low level features that could present weak linear relations to the vehicle count. These features include 1) segment area; 2) segment length along the road direction; 3) segment width; 4) segment perimeter associated with the number of pixels on the segment boundary; 5) horizontal edge length within the segment (i.e., segment boundary is not included); 6) texture coarseness.

These features are similar to the features used in [23] with two main differences. First, they are normalized (e.g., re-scaled) based on the smoothing spline estimated in Fig. 2(c). A reference line with y coordinate close to the image bottom is chosen for the normalization. Secondly, as traffic trajectory is known or has been estimated, segment length and width are not sensitive to the road direction. Thus, they are computed by projection onto the road direction and the direction perpendicular to the road direction. Longer length and width often indicate more vehicles or larger vehicles in the segment. Horizontal edge length within a segment is a feature that could distinguish a large vehicle and multiple small vehicles that may have the similar area and segment length. Texture coarseness consists of measurements of homogeneity, energy, and entropy along four orientations [23], [33]. Similar to horizontal edge length, texture coarseness could also help us estimate the different vehicle classes. In general, smoother texture tends to contain less or larger vehicles. Fig. 5 shows some of features from an image frame.

These features are then concatenated together to form a 17×1 feature vector. This feature vector is the input vector (i.e., $\mathbf{x}$) in the regression module to estimate count of large vehicles.

C. Cascaded Regression

We are interested in three different vehicle classes (i.e., large, medium, and small size) corresponding to truck/bus, SUV/minivan, and sedan, respectively. There are two reasons for using three different classes. First, we tend to combine vehicle counts and classes to estimate emissions on highways.

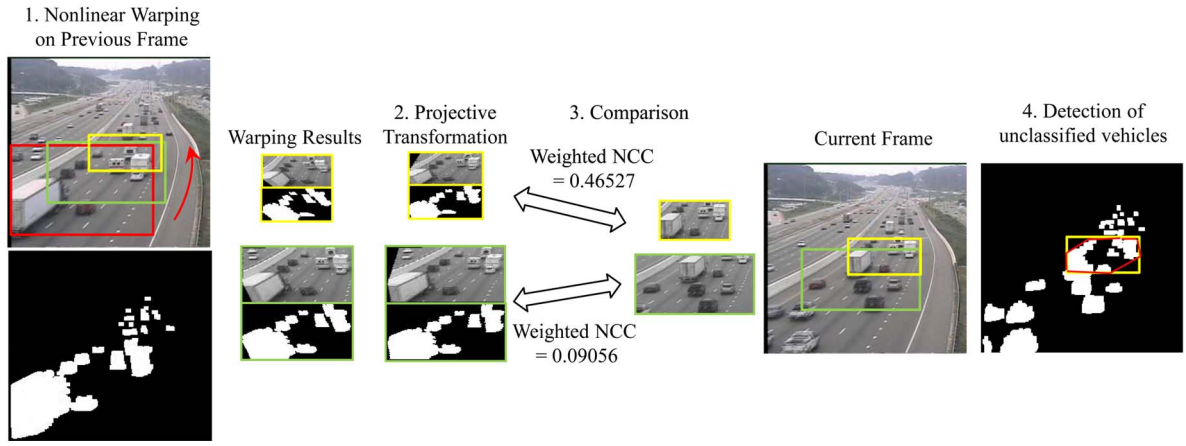

Fig. 4. An example of warping and detection processes. (From left to right) 1) Nonlinear warping based on the estimated mesh grids (e.g., warping from the red rectangle to the green and yellow rectangles, respectively); 2) projective transformation to reduce the distortions; 3) comparisons between warped results and corresponding patches in the current frame based on weighted normalized cross correlation; 4) detection of unclassified vehicles that are below the convex hull (red polygon).

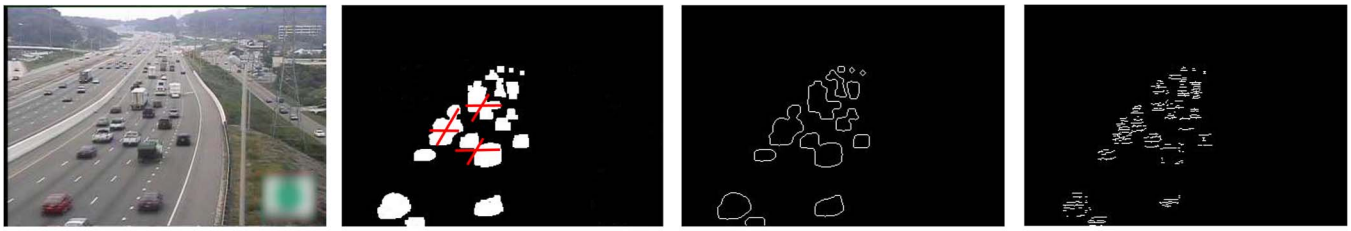

Fig. 5. Illustration of some extracted features. (a) Original image frame. (b) Foreground segments where red lines indicate segment length and width. (c) Segment perimeter. (d) Horizontal edge length within the segments.

In the area of atmospheric environment, McGaughey *et al.* applied a set of linear regressions to estimate relations between emission factors and diesel contribution for a Houston tunnel [34]. Different classes of vehicles are manually counted by observers on traffic videos that have been captured. Our algorithm is designed to automatically estimate counts and emissions in real time. Secondly, as our cameras only capture low resolution and low frame rate videos, it would be difficult to detect many different classes, and even specific models of highway vehicles. There are four different classes, cars/jeeps, light-duty trucks, medium-duty trucks (i.e., trucks with two axles and six tires), and heavy-duty trucks (trucks with three or more axles), are used in [34]. However, it is difficult for us to use these categories based on the criteria of number of axles and tires. In our image frames, many tires and axles are invisible due to occlusion and low video quality.

In statistics, regression analysis is often applied to build a mapping from an input variable to a continuous output variable. It has been widely used in the area of image processing. In the area of intelligent transportation systems, different regression algorithms have been applied to predict travel time, forecast traffic flow and detect traffic incidents, and freeway traffic states [35]–[37]. In our algorithm, we need to build a regression model from input variable, which is the 17×1 feature vector, to the count of large, medium, and small vehicles, which are our output variables. One possible solution is to assume the counts of three vehicle classes are independent with each other. Then we can apply regression for three classes separately given the same input feature vectors. However, this solution could not be accurate or robust as the assumption of independence is often not valid. Therefore, we apply a three-level cascaded regression.

The first level is the regression for the large vehicles as all the features are less affected by the counts of medium and small vehicles. The estimated count is treated as a feature and concatenated to the original feature vector. The new feature vector is used as input variable for the second level to estimate count of medium vehicles. Both counting results from these two levels are concatenated to the original feature vector and used to estimate the count of small vehicles. Fig. 6 shows the cascaded regression framework.

We evaluate three different regression methods to find out which kind of regression method could be more suitable for our application.

The simplest regression could be the linear regression that is a linear combination of the input variables. We also could use nonlinear functions of the input variables, such as polynomial functions, spline functions, and logistic sigmoid functions. These regression methods are parametric models, which could be sensitive to nonlinearities of the input data and cause overfitting when input variable is in a high-dimensional space. Therefore, we first evaluate the Gaussian process, which could be used to specify flexible nonlinear regression for more complex data set. However, the output variable by using Gaussian

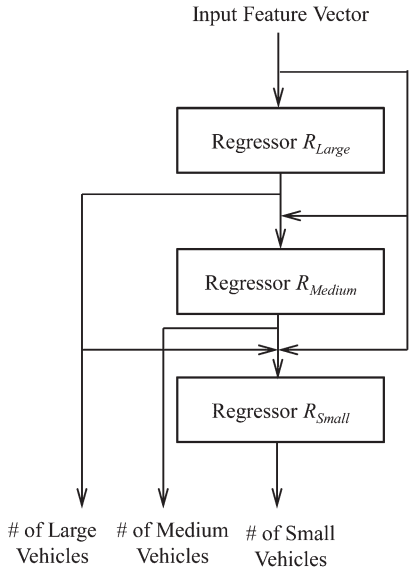

Fig. 6. Three-level cascaded regression. R_{Large}, R_{Medium}, R_{Small} are regression modules for large, medium, and small vehicles respectively.

process could be negative, which cannot happen in our application. Hence, the second regression method that we evaluate is the Poisson regression that is mainly used to model count data. For comparison purpose, we also evaluate the Bayesian Poisson regression proposed in [23].

Gaussian Process: The regression model with Gaussian noise is given by

$$y = f(\mathbf{x}) + \epsilon \tag{2}$$

where ϵ is a random noise variable that is independent for each observation, y is the count of a vehicle class, and $\mathbf{x}$ denotes the feature vector. The Gaussian likelihood for the count is given by

$$p(\mathbf{y}|f(\mathbf{x})) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I) \tag{3}$$

where σ^2 is the variance of noise. Based on the definition of a Gaussian process, the Gaussian prior has the zero mean and a Gram matrix $\mathbf{K}$ as the covariance

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{K}). \tag{4}$$

The kernel function $k(\mathbf{x}, \mathbf{x}')$ that determines $\mathbf{K}$ is given by adding the squared exponential function, a linear term, and a constant.

$$k(\mathbf{x}, \mathbf{x}') = \theta_1 \exp\left(-\frac{\theta_2}{2}|\mathbf{x} - \mathbf{x}'|^2\right) + \theta_3 \mathbf{x}^T \mathbf{x}' + \theta_4 \tag{5}$$

where $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$ are the hyperparameters and optimized by maximizing the log likelihood $p(\mathbf{y}|\theta)$. The linear and constant terms in the kernel function are used to model the linear relation between input and output. The squared exponential function could model the local nonlinearities that are caused by occlusions and segmentation errors.

Suppose there is a new input x_*, let us define $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$ and $\mathbf{y} = (y_1, \ldots, y_N)^T$ (N is the size of training set), then the predictive distribution is given by

Fig. 7. Sample images captured at different time intervals.

$$p(y_*|\mathbf{X}, \mathbf{y}, x_*) = \mathcal{N}(m, \sigma_*^2) \tag{6}$$

where

$$m = \mathbf{k}(x_*, \mathbf{X})(\mathbf{K} + \sigma^2 I)^{-1}\mathbf{y}$$
$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*, \mathbf{X})^T(\mathbf{K} + \sigma^2 I)^{-1}\mathbf{k}(\mathbf{X}, \mathbf{x}_*)$$

Poisson Regression: As the vehicle count is a nonnegative integer, the typical regression choice is the Poisson regression where $y \sim \text{Poisson}(\mu)$. The canonical link is the log,

$$\log(\mu) = \eta = \mathbf{w}^T \phi(\mathbf{x}). \tag{7}$$

Iterative reweighted least squares could be applied to fit this model. The update formula is $\mathbf{w} = (\mathbf{\Phi} \boldsymbol{R} \mathbf{\Phi})^{-1} \mathbf{\Phi}^{\boldsymbol{T}} \mathbf{R} \mathbf{z}$, where $\mathbf{R} = \text{Diag}(\mu_i)$ is the weight matrix, $z_i = \eta_i + (y_i - \mu_i)/\mu_i$, and $\mathbf{\Phi}$ is the design matrix.

Bayesian Poisson Regression: A Bayesian model for count regression is proposed in [23]. After approximations on the posterior distribution $p(\mathbf{w}|\mathbf{X}, \mathbf{y})$, the prediction distribution could be modeled by a negative binomial distribution,

$$p(y_*|\mathbf{X}, \mathbf{y}, x_*) = \mathcal{NB}(e^{\mu_\eta}, \sigma_\eta^2) \tag{8}$$

with mean and variance

$$\mu_\eta = \mathbf{k}(x_*, \mathbf{X})(\mathbf{K} + \Sigma_{\mathbf{y}})^{-1}\mathbf{t}$$
$$\sigma_\eta^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*, \mathbf{X})^T(\mathbf{K} + \Sigma_{\mathbf{y}})^{-1}\mathbf{k}(\mathbf{X}, \mathbf{x}_*)$$

where $\Sigma_{\mathbf{y}} = \text{Diag}(1/(y_1 + c), \ldots, 1/(y_N + c))$ and $\mathbf{t} = \log(\mathbf{y} + c) - c\Sigma_{\mathbf{y}}\mathbf{1}$. The kernel function we used here is same as the kernel in (5).

IV. Experiments

We remotely collected close to 70-minutes videos from a local transportation department at different highway locations at different time intervals. Image size is 352 × 240 and frame

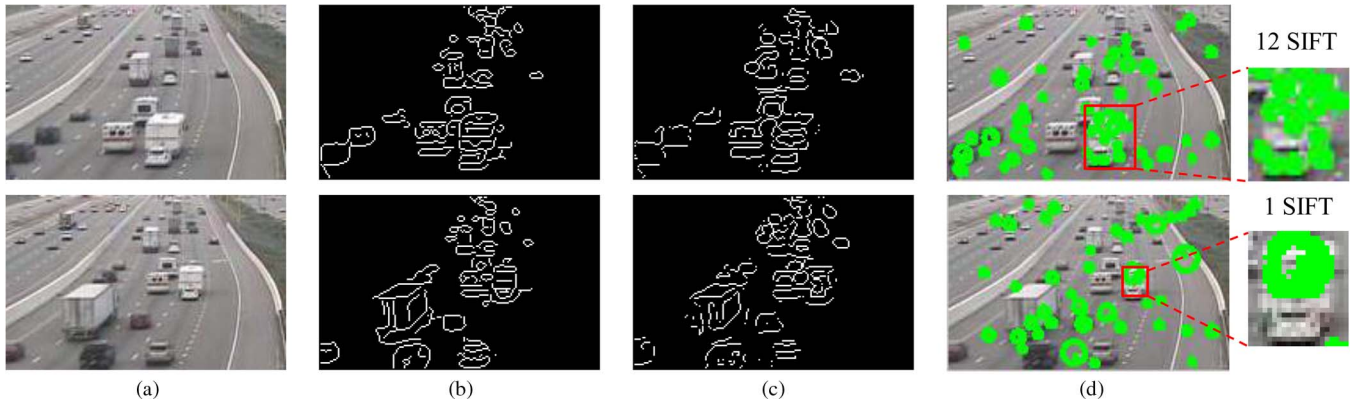

Fig. 8. Illustration of unreliable contour and SIFT features. (a) Two consecutive image frames. (b) Contours from the Canny method. (c) Contours from the Laplacian of Gaussian method. (d) The SIFT feature points. The enlarged regions are from the same vehicles, which contain 12 SIFT features and 1 SIFT feature, respectively.

Fig. 9. Five consecutive image frames and their corresponding foreground segments. Yellow segments contain unclassified vehicles and white segments contain classified vehicles that have been classified in previous frame. All the segments are extracted from the pre-defined region of interest.

rate is only around 1 fps that may be caused by a slow network transfer. In each image frame, we manually counted the new entered vehicles according to their classes. These manually counted results are used as the ground truth. A region of interest (e.g., the region close to the camera) is selected for each highway road. Fig. 7 shows sample images captured at different time intervals.

Our algorithm is implemented using Matlab and can be run in real time. The most time consuming part is the warping algorithm, which is 0.16 ms on average. The prediction based on the regression is around 3.4×10^{-4} ms per foreground segment.

As mentioned in the related work, different features (e.g., contours and feature points) could be extracted for traffic analysis [2], [3], [7]–[11], [17]. These features could be used to fit into a 2-D/3-D vehicle model or be tracked over image frames. Fig. 8 shows the extraction of two kinds of contour features (the contours in the background have been removed) and the SIFT feature points. As most contours are mixed together and not distinguishable, they cannot be easily used to fit into one vehicle model. As the vehicle resolution is small, the number of the SIFT features is very limited even we set the peak threshold of the DoG scale space to the minimum. These SIFT features are also highly inconsistent over image frames, which makes the modeling and tracking difficult.

For instance, SIFT features need to be matched and tracked in the foreground segments in [7]. Motion vectors are then grouped by a hierarchical clustering algorithm. As shown in the enlarged regions in Fig. 8(d), the same vehicle in two consecutive image frames contains 12 SIFT features and 1 SIFT features, respectively. Therefore, it would be very difficult to match and track these SIFT features.

We believe that other similar feature extraction methods, such as the Harris corner, would have the same problem. As a result, many existing algorithms (e.g., [2], [3], [7]–[11], [17]) are not suitable for low quality videos and could easily fail to extract features so that counting and classification are highly inaccurate.

Many existing algorithms also need to segment individual vehicles either before or after tracking. This segmentation itself could be a difficult problem when vehicle resolution is small and severe occlusions present. As a result, in many existing algorithms (e.g., [7], [13], [14], [20], [21]), we can only find segmentation of around two occluded vehicles.

Fig. 9 shows the five image frames and their corresponding foreground segments. Because of the low frame rate, vehicles often become very small after one image frame and are almost invisible after two image frames. There are severe occlusions in foreground segments. Each foreground segment could easily contain many vehicles. The assumption of the weak perspective

TABLE I
PERFORMANCE COMPARISON BY USING THREE REGRESSION METHODS AND DIFFERENT FEATURES. BASED ON ALL FEATURES, THE BEST CLASSIFICATION RATES FOR LARGE, MEDIUM, AND SMALL VEHICLES ARE 92.7%, 63.4%, AND 79.9%, RESPECTIVELY, WHICH ARE CORRESPONDING TO MEAN ABSOLUTE ERRORS 0.146, 0.732, AND 0.401

Feature	Regression Method	Mean Absolute Error (Standard Deviation)		
		Large	Medium	Small
All	Gaussian Process	1.028 (0.384)	1.788 (1.084)	1.171 (0.6853)
All	Poisson	**0.146** (0.278)	0.752 (0.298)	**0.401** (0.281)
All	Bayesian Poisson	0.196 (0.292)	**0.732** (0.654)	0.730 (0.439)
Area	Gaussian Process	0.98 (0.236)	1.64 (1.170)	1.57 (1.011)
Area	Poisson	0.143 (0.278)	1.210 (0.298)	**0.467** (0.281)
Area	Bayesian Poisson	**0.080** (0.294)	**0.827** (0.709)	1.213 (0.633)
Segment	Gaussian Process	1.020 (0.161)	1.83 (1.167)	1.82 (0)
Segment	Poisson	**0.153** (0.195)	**0.917** (0.200)	**0.488** (0.229)
Segment	Bayesian Poisson	0.27 (0.267)	1.14 (0.884)	1.03 (0.506)
Edge	Gaussian Process	0.98 (0.224)	1.87 (1.167)	1.6 (0.994)
Edge	Poisson	**0.143** (0.320)	1.260 (0.221)	**0.477** (0.719)
Edge	Bayesian Poisson	0.25 (0.59)	**1.040** (0.713)	1.03 (0.761)
Texture	Gaussian Process	0.91 (0.274)	1.60 (1.174)	1.51 (1.024)
Texture	Poisson	**0.157** (0.457)	**0.910** (0.439)	0.965 (0.559)
Texture	Bayesian Poisson	0.375 (0.564)	1.010 (0.719)	**0.485** (0.465)

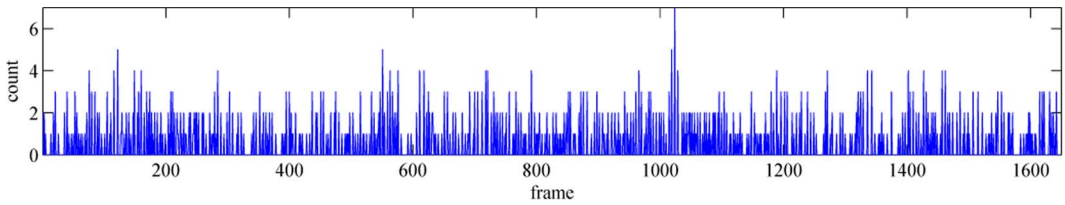

Fig. 10. The ground truth of small size vehicles in a traffic video (around 30 minutes).

projection that has been used in some existing algorithms is also not valid here.

Three different regression methods are evaluated, in which training and testing data are completely separated. The absolute error for each vehicle class ($\text{err} = \frac{1}{N}\sum|\hat{c}_i - c_i|$, where $\hat{c}_i$ and c_i are the estimated and true counts in the ith foreground segment, and N is the number of foreground segments) is computed. Table I shows the counting results for all the features and different categories of features. In order to obtain an accurate evaluation, the image frames that contain no or few vehicles are removed in this experiment as these image frames are not challenging and could greatly reduce the absolute errors. For the selected image frames, the number of small and medium size vehicles in one foreground segment could reach up to 11 and the number of large size vehicles could reach up to 4. On average, there are around 4 small and medium vehicles and 2 large vehicles in each foreground segment. We can find that the performance by using all the features is better than the performance only using one kind of features. Our algorithm is accurate and robust to count large size vehicles. For example, the mean absolute error using all features and the Poisson regression is 0.146 per foreground segment, which means the algorithm could miscount 15 large size vehicles for every 100 foreground segments that contain around 200 large size vehicles. For small and medium size vehicles, our algorithm is less accurate. For example, the mean absolute error using all features and the Bayesian Poisson is 0.732 per foreground segment. Therefore, based on all features, the best classification rates for large, medium, and small vehicles are 92.7%, 63.4%, and 79.9%, respectively, which are corresponding to mean absolute errors 0.146, 0.732, and 0.401. As shown in Fig. 6, the counting results for small and medium vehicles also reply on the counting results of large vehicles. Therefore, the errors made in the first-level regression would be propagated to the second and third level regression.

If the cascaded regression framework is not applied for small and medium vehicles (i.e., the counts of different vehicles in one segment are mutually independent), the counting and classification performance is further reduced. For example, if the Poisson regression and all features are applied in this experiment, the mean absolute errors could be increased from 0.752 (0.298) to 0.893 (0.429) for medium size vehicles and from 0.401 (0.281) to 0.820 (0.231) for small size vehicles, where the numbers in the parentheses are the corresponding standard deviations.

The performance of Gaussian process is worse than other two regression methods. Standard Poisson regression is slightly better than the Bayesian Poisson regression proposed in [23]. One of possible reasons could be that vehicle counts (e.g., up to 11) generally are much less than the pedestrian counts (e.g., up to 50). There are no enough local nonlinearities so that squared exponential kernel function is not very useful comparing with the linear kernel.

Figs. 10 and 11 show the count estimates using standard Poisson regression and ground truth of small size vehicles over a moderate traffic video. The video length is close to

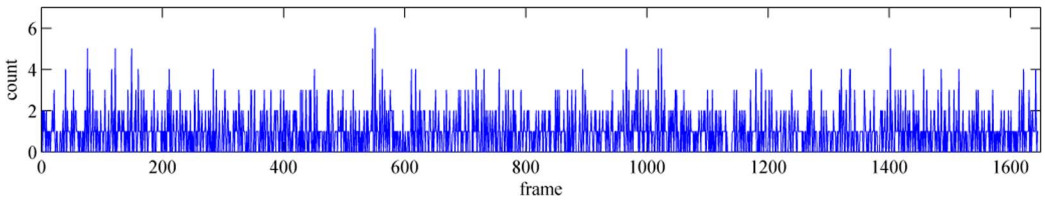

Fig. 11. Our estimation of small size vehicles in the same traffic video used in Fig. 10.

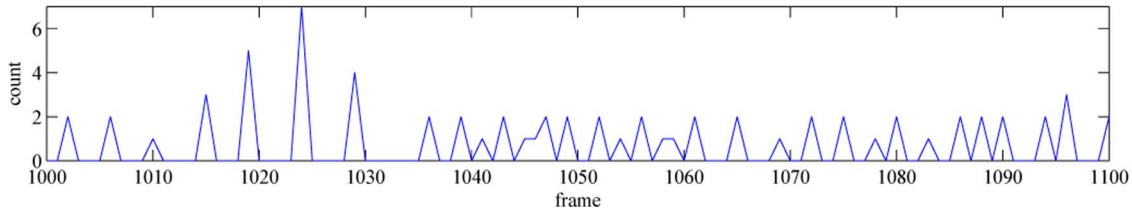

Fig. 12. The ground truth between 1000th and 1100th image frame in Fig. 10.

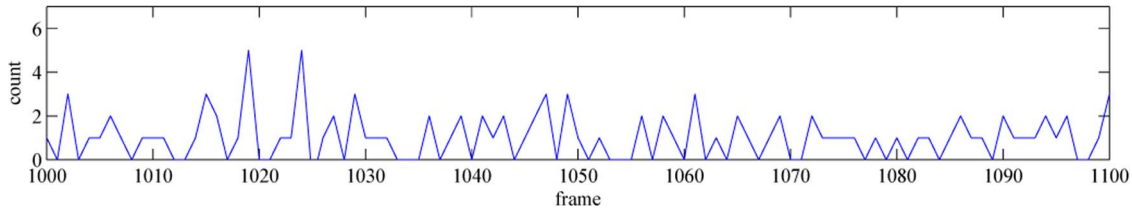

Fig. 13. Our estimation between 1000th and 1100th image frame in Fig. 11.

30 minutes. We can find that these two distributions are similar. This indicates that our estimation can be used to approximate the traffic density distribution. In terms of each image frame, it is also common that there is a difference of 1 or 2 vehicles between the ground truth and our estimation. We believe that it could be difficult to further reduce the errors unless image resolution and frame rate are increased. Figs. 12 and 13 show the results between 1000th and 1100th image frame of Figs. 10 and 11.

V. Discussion

In many segmentation based algorithms, the boundary of an individual vehicle or the boundaries shared by multiple vehicles need to be estimated. However, the boundary estimation could be considered as a more difficult problem than the vehicle counting itself. One advantage of using regression analysis is that the vehicle boundary estimation is not required and the counting problem can be addressed directly. The major factor that affects our algorithm performance is the feature vector extracted from the foreground segment. Normalization of these features based on the smoothing splines is an important step to reduce the effects from perspective projection. Without this step, the small size vehicles close to the camera could have some similar features with the large size vehicles far away from the camera.

Another advantage of using regression analysis is that prediction stage of regression is often very fast. All the regression methods we used could make predictions in real time. It is also possible to choose other supervised learning techniques to learn the relation between the feature vector and the vehicle count, such as a neural network. For sufficiently large number of hidden units, a two-layer neural network could have a similar performance of Gaussian process.

Our algorithm is mainly designed to count and classify highway vehicles. Without finding individual vehicles, it is not easy to extend our algorithm to other applications, such as detection of complex events for urban traffic. This could be one limitation of the algorithm.

Our algorithm is currently trained and evaluated at different time intervals during daytime. However, our algorithm still cannot handle many different weather conditions, such as the "transition" weather condition presented in [29]. This is another limitation of the algorithm. For example, our experiments are currently conducted when small shadow areas present. However, features could be strongly affected by large shadow areas. In order to improve robustness, it could be useful to add shadow removal to our framework. Our algorithm also could not be applied during nighttime. One reason is that vehicle features could not quite different during daytime and nighttime. In order to partially solve the problem, we could train multiple regression models based on different time intervals.

VI. Conclusion

In this paper, we present a counting and classification algorithm for highway vehicles. Unlike many existing algorithms,

our algorithm requires no explicit segmentation of individual vehicles. Our algorithm also does not rely on tracking of robust features. Given a set of low level features, we apply a cascaded regression model to count and classify vehicles directly. We have tested our algorithm on low quality videos that last more than one hour. We show that our algorithm can deal with the traffic with severe occlusions and very low vehicle resolutions. Our algorithm is suitable for vision based systems that are non-intrusive and can be mounted many places near highways. Our algorithm could be further applied for estimation of traffic density and vehicle emissions. Looking into the future, there are many areas that could be improved. One immediate step is to apply more sophisticated algorithms for background estimation and shadow removal.

REFERENCES

[1] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.

[2] J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank, "3-D model-based vehicle tracking," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1561–1569, Oct. 2005.

[3] C. C. C. Pang, W. W. L. Lam, and N. H. C. Yung, "A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 441–459, Sep. 2007.

[4] B. Johansson, J. Wiklund, P.-E. Forssén, and G. Granlund, "Combining shadow detection and simulation for estimation of vehicle size and position," *Pattern Recognit. Lett.*, vol. 30, no. 8, pp. 751–759, Jun. 2009.

[5] T. Gao, Z. Liu, W. Gao, and J. Zhang, "Moving vehicle tracking based on sift active particle choosing," in *Advances in Neuro-Information Processing*. Berlin, Germany: Springer-Verlag, 2009, pp. 695–702.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[7] G. Jun, J. Aggarwal, and M. Gokmen, "Tracking and segmentation of highway vehicles in cluttered and crowded scenes," in *Proc. IEEE WACV*, 2008, pp. 1–6.

[8] Z. Kim and J. Malik, "Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 524–531.

[9] M. J. Leotta and J. L. Mundy, "Vehicle surveillance with a generic, adaptive, 3D vehicle model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1457–1469, 2011.

[10] X. Ma and W. E. L. Grimson, "Edge-based rich representation for vehicle classification," in *Proc. Tenth IEEE ICCV*, 2005, vol. 2, pp. 1185–1192.

[11] N. Buch, J. Orwell, and S. A. Velastin, "3D extended histogram of oriented gradients (3dhog) for classification of road users in urban scenes," in *Proc. BMVC*, 2009, pp. 1–11.

[12] S. Messelodi, C. M. Modena, and M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern Anal. Appl.*, vol. 8, no. 1/2, pp. 17–31, 2005.

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

[14] 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.

[15] D. Alonso, L. Salgado, and M. Nieto, "Robust vehicle detection through multidimensional classification for on board video based systems," in *Proc. IEEE ICIP*, 2007, vol. 4, pp. IV-321–IV-324.

[16] C. Gentile, O. Camps, and M. Sznaier, "Segmentation for robust tracking in the presence of severe occlusion," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 166–178, Feb. 2004.

[17] N. K. Kanhere, S. J. Pundlik, and S. T. Birchfield, "Vehicle segmentation and tracking from a low-angle off-axis camera," in *Proc. IEEE CVPR*, 2005, vol. 2, pp. 1152–1157.

[18] X. Song and R. Nevatia, "A model-based vehicle segmentation method for tracking," in *Proc. 10th IEEE ICCV*, 2005, vol. 2, pp. 1124–1131.

[19] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 108–118, Jun. 2000.

[20] C.-L. Huang and W.-C. Liao, "A vision-based vehicle identification system," in *Proc. 17th ICPR*, 2004, vol. 4, pp. 364–367.

[21] W. Zhang, Q. J. Wu, X. Yang, and X. Fang, "Multilevel framework to detect and handle vehicle occlusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 161–174, Mar. 2008.

[22] R. Jiang, M. Terauchi, R. Klette, S. Wang, and T. Vaudrey, *Low-Level Image Processing for Lane Detection and Tracking*. Berlin, Germany: Springer-Verlag, 2010.

[23] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, Apr. 2012.

[24] N. K. Kanhere and S. T. Birchfield, "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 148–160, Mar. 2008.

[25] P. Kumar, S. Ranganath, and H. Weimin, "Bayesian network based computer vision algorithm for traffic monitoring using video," in *Proc. IEEE Intell. Transp. Syst.*, 2003, vol. 1, pp. 897–902.

[26] B. Morris and M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," in *Proc. IEEE ITSC*, 2006, pp. 1078–1083.

[27] J.-S. Hu and T.-M. Su, "Robust background subtraction with shadow and highlight removal for indoor surveillance," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, p. 108, Jan. 2007.

[28] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Computer Vision—ECCV*. Berlin, Germany: Springer-Verlag, 2000, pp. 751–767.

[29] L. Unzueta *et al.*, "Adaptive multicue background subtraction for robust vehicle counting and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 527–540, Jun. 2012.

[30] S. Atev, G. Miller, and N. P. Papanikolopoulos, "Clustering of vehicle trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 647–657, Sep. 2010.

[31] K. Kim, D. Lee, and I. Essa, "Gaussian process regression flow for analysis of motion trajectories," in *Proc. IEEE ICCV*, 2011, pp. 1164–1171.

[32] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[33] A. N. Marana, L. d. F. Costa, R. Lotufo, and S. Velastin, "On the efficacy of texture analysis for crowd monitoring," in *Proc. SIBGRAPI*, 1998, pp. 354–361.

[34] G. R. McGaughey *et al.*, "Analysis of motor vehicle emissions in a Houston tunnel during the Texas Air Quality Study 2000," *Atmosp. Environ.*, vol. 38, no. 20, pp. 3363–3372, Jun. 2004.

[35] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.

[36] S. Tang and H. Gao, "Traffic-incident detection-algorithm based on nonparametric regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 38–42, Mar. 2005.

[37] L. Li, X. Chen, and L. Zhang, "Multimodel ensemble for freeway traffic state estimations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, Jun. 2014.

Mingpei Liang received the B.S. degree in electrical engineering from Beijing University, Beijing, China. He is currently working toward the master's degree in computer science with North Carolina Central University, Durham, NC, USA, under the supervision of Dr. X. Huang and Dr. A. Tokuta. He is currently with NetBrain Technologies Inc., Boston, MA, USA. His research interests include computer vision and image processing.

Xinyu Huang received the B.S. degree in mechanical engineering from Huazhong University of Science and Technology, Wuhan, China, the M.S. degree in computer science from Eastern Kentucky University, Richmond, KY, USA, and the Ph.D. degree in computer science from the University of Kentucky, Lexington, KY. He is currently an Assistant Professor with the Department of Mathematics and Computer Science, North Carolina Central University, Durham, NC, USA. His research interests include computer vision, image processing, pattern recognition, and machine learning.

Chung-Hao Chen received the B.S. and M.S. degrees in computer science and information engineering from Fu Jen Catholic University, New Taipei City, Taiwan, in 1997 and 2001, respectively, and the Ph.D. degree in electrical engineering from The University of Tennessee, Knoxville, TN, USA, in 2009. In 2009, he joined the Department of Mathematics and Computer Science, North Carolina Central University, Durham, NC, USA, as an Assistant Professor and retained this position until 2011. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA, USA. His research interests include object tracking, robotics, and image processing.

Xin Chen received the B.S. degree in computer science from the University of Science and Technology of China, Hefei, China, and the M.S. and Ph.D. degrees in computer science and engineering from the University of Notre Dame, Notre Dame, IN, USA. He is a Senior Research Scientist and an Engineering Manager at HERE (a Nokia company), Chicago, IL, USA. He is also the University Cooperation Ambassador for the Nokia Intellectual Property Business Unit. Currently, he is an Adjunct Professor at the Department of Computer Science, Illinois Institute of Technology, and the Department of Electrical Engineering and Computer Science, Northwestern University.

Alade Tokuta received the B.S. and M.S. degrees in electrical engineering from Duke University, Durham, NC, USA, the E.E. degree in electrical engineering from Columbia University, New York, NY, and the Ph.D. degree in electrical engineering and computer science from the University of Florida, Gainesville, FL, USA. Currently, he is a Professor at the Department of Mathematics and Computer Science, North Carolina Central University, Durham. His research interests include robotics, computer image synthesis/vision, networking, and algorithm design.