Journal of
Electronic Imaging

SPIEDigitalLibrary.org/jei

Visual traffic surveillance framework: classification to event detection

Amol Ambardekar
Mircea Nicolescu
George Bebis
Monica Nicolescu

SPIE

IS&T
imaging.org

Visual traffic surveillance framework: classification to event detection

Amol Ambardekar
Mircea Nicolescu
George Bebis
Monica Nicolescu
University of Nevada, Reno
Department of Computer Science and Engineering
Reno, Nevada 89557
E-mail: ambardek@cse.unr.edu

Abstract. *Visual traffic surveillance using computer vision techniques can be noninvasive, automated, and cost effective. Traffic surveillance systems with the ability to detect, count, and classify vehicles can be employed in gathering traffic statistics and achieving better traffic control in intelligent transportation systems. However, vehicle classification poses a difficult problem as vehicles have high intraclass variation and relatively low interclass variation. Five different object recognition techniques are investigated: principal component analysis (PCA)+difference from vehicle space, PCA+difference in vehicle space, PCA+support vector machine, linear discriminant analysis, and constellation-based modeling applied to the problem of vehicle classification. Three of the techniques that performed well were incorporated into a unified traffic surveillance system for online classification of vehicles, which uses tracking results to improve the classification accuracy. To evaluate the accuracy of the system, 31 min of traffic video containing multilane traffic intersection was processed. It was possible to achieve classification accuracy as high as 90.49% while classifying correctly tracked vehicles into four classes: cars, SUVs/vans, pickup trucks, and buses/semis. While processing a video, our system also recorded important traffic parameters such as the appearance, speed, trajectory of a vehicle, etc. This information was later used in a search assistant tool to find interesting traffic events.* © 2013 SPIE and IS&T [DOI: 10.1117/1.JEI.22.4.041112]

1 Introduction

In the last decade, we have seen a worldwide rise in the use of closed-circuit television cameras. In recent years, we are beginning to see a corresponding rise in video processing systems that can interpret the video to extract information such as actors, objects, actions, and events. The rapidly increasing capacity of digital storage and computation power and the recent innovations in video compression standards[1] have led to a strong growth of available video content. There are two major ways in which the available video content can be processed: online or offline on a need-to-know basis. For online processing of the video content, we need to have either the manpower that is expensive or the computing capability to automate the process. The same is true for offline processing when we want to process a very large amount of video content to find an exact event we are interested in. The online processing of video data is far more useful in comparison to passively recording video footage, as automated surveillance systems can detect events requiring attention and take action in real time by alerting a human supervisor. Video surveillance is a repetitive task and computers are more suited to do repetitive tasks that require limited human intervention. Therefore, automated video surveillance is attracting more attention as it can alleviate the problems faced by manual surveillance such as the lack of attention span or the increasing number of channels.

Common traffic sensors include push buttons (detecting pedestrian demand), loop detectors (detecting vehicle presence), magnetic sensors, radar sensors, and video cameras. A video camera is a promising traffic sensor because of its low cost and its potential ability to collect a large amount of information (such as the number of vehicles, vehicle speed/acceleration, vehicle class, vehicle track), from which higher-level information (such as speeding, illegal turns, one-way streets, etc.) can also be inferred. For the task of identifying and classifying vehicles using visual information, currently the most reliable approach is through the automatic number plate recognition (ANPR), which is also known as automatic license plate recognition.[2–4] Nevertheless, ANPR tends to be effective only for specialized camera views (zoomed on plates) and cannot provide wide-area observation or the measurement of the interactions between road users.

A visual traffic surveillance system needs to detect vehicles and classify them if possible. Generating vehicle trajectories from video data is also an important application and can be used in analyzing traffic flow parameters for advanced transportation management systems.[5] Efficient and robust localization of vehicles from an image sequence (video) can lead to semantic results, such as "vehicle no. 5 stopped, vehicle no. 8 is moving" or more advanced semantic results that include specific information such as "blue SUV is moving at 40.0 mph, red sedan is turning right." However, such high-level information is possible if we can not only detect vehicles but also track and classify them. Vehicle tracking provides a way to correlate detected vehicles in consecutive time frames and is useful in scenarios such as vehicle counting, stopped/speeding vehicle detection, etc. The class of a

Paper 13210SS received Apr. 16, 2013; revised manuscript received Jul. 12, 2013; accepted for publication Jul. 26, 2013; published online Aug. 28, 2013.

detected vehicle can also supply important information that can be used to make sure that certain types of vehicles do not appear in certain areas under surveillance in the case of security-related surveillance. Multicamera systems such as the one used in Ref. 6 can benefit immensely from the vehicle class information, as it can help in matching objects detected in nonoverlapping field of views from different cameras. In general, a visual traffic surveillance system with the ability to do vehicle tracking along with classification can play an important part in intelligent transportation systems.[7]

The overview of a general traffic video surveillance system, with its components, is shown in Fig. 1. The first two stages—object detection and object tracking—have achieved good accuracy in recent years. However, the same cannot be said about object classification, which is the last stage in the surveillance framework as shown in Fig. 1. Object recognition in case of still images has the problem of dealing with the clutter in the scene and a large number of classes. Object recognition in video sequences has the benefit of using background segmentation to remove clutter.[8] However, images obtained from video surveillance cameras are generally of low resolution, and in the case of traffic video surveillance, the vehicles cover very small areas of these images, making the classification problem challenging. Vehicle classes such as cars and vans are difficult to differentiate as they have similar sizes. Therefore, classification techniques that use global features such as size and shape of the detected blob do not yield satisfactory results. In the case of traffic video surveillance, the object classification stage can be implemented with a single-look classifier or a multilook classifier. The single-look classifier uses a detection region defined by a user and the objects present within the detection region are classified. This scheme has an advantage of being computationally less expensive, but it may not produce good results if the object never enters the detection region or there is a partial occlusion when the object is within the detection region. The multilook classifier processes a detected object for classification multiple times while the detected object is within the field of view of the camera, which generally results in better overall accuracy.

This work is divided into three parts. First, we implemented and compared different object recognition techniques for the purpose of vehicle classification. Second, we developed a unified visual traffic surveillance framework that can reliably detect, track, and classify vehicles by incorporating the most promising vehicle classification techniques that were discovered during our comparison stage. Third, we also developed a search assistant tool that can assist users by answering queries based on various attributes, finding instances of traffic events such as “a white pickup truck going left to right” in large amounts of traffic video.

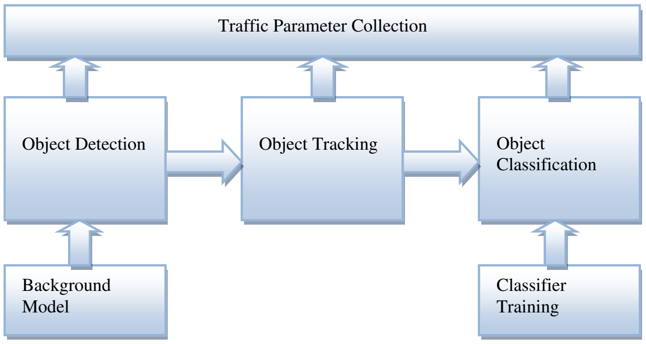


Fig. 1 Overview of a general traffic video surveillance system.

The object classification problem is the most challenging part of the visual traffic surveillance system, especially when the video resolution is low. As such, we spend a significant portion of the paper describing different object recognition techniques implemented for the purpose of vehicle classification. We implemented five different object classification techniques: principal component analysis (PCA)[9] + difference from vehicle space (DFVS), PCA+difference in vehicle space (DIVS), PCA+support vector machine (SVM),[10] linear discriminant analysis (LDA),[11] and constellation-based modeling.[12] In the first four approaches, we create a principal component space (PCS) using PCA, which we call vehicle space. In the case of PCA+DFVS, the decision is made by finding the distance from a separate vehicle space for each class, and therefore it is named distance from vehicle space. On the other hand, PCA+DIVS predicts the class of a test image after projecting a test image onto a combined vehicle space and distance from each class is calculated in vehicle space, and therefore it is named distance in vehicle space. In PCA+SVM, the features extracted using PCA are used to train an SVM classifier, which is later used to classify vehicles. PCA depends upon most expressive features (MEFs) that can be different from most discriminant features (MDFs); therefore we also implemented LDA that relies on MDFs. Some of the approaches based on PCA have been presented in the literature with some variations.[13] However, the PCA+DFVS approach as presented in this paper is a new application of PCA for the solution of vehicle classification problem. Also, for other PCA-based techniques, we use a different formulation in the classification process, along with using the entire bounding box region (detected blob) as a candidate instead of just the foreground-segmented blob as described in Ref. 13, which sometimes loses vital information due to background subtraction errors. The constellation-based approach presented in this paper relies on the techniques used in Ref. 12 and extends it for the multiclass case.

The rest of the paper is organized as follows. Section 2 provides details of various object recognition techniques that we compared for the problem of vehicle classification and also discusses implementation details of the unified traffic surveillance system. In Sec. 3, we describe five vehicle classification techniques. Section 4 explains the details of the unified visual traffic surveillance system. The results obtained while comparing vehicle classification techniques and a discussion about the performance achieved on an actual traffic video sequence using our unified traffic surveillance system are provided in Sec. 5. Section 5 also details the search assistant tool for finding events in a traffic video. Section 6 discusses our conclusions and presents future directions of work.

2 Previous Work

As discussed in the previous section, a comprehensive video traffic surveillance system needs to have capabilities to perform vehicle detection, tracking, and classification. There are also additional components such as camera calibration, vehicle pose estimation, background modeling, and foreground object detection. There have been numerous attempts made toward addressing problems in traffic surveillance using different computer vision techniques. These attempts can be broadly classified into two main categories based on the application domain: urban traffic or highway traffic. In these two domains, urban traffic poses relatively more difficult challenges. This is partly due to the easier conditions on a highway, typically with more homogeneous and constant flow than in urban areas. In addition, the distance between vehicles is larger, which reduces the amount of occlusion. Urban traffic poses two main challenges: the high density of vehicles and the low camera angle. The combination of both factors leads to a high degree of occlusion. In addition, the clutter on the streets increases the complexity of scenes. Object classification in general is a challenging problem, and vehicle classification poses another challenge as interclass variability is relatively small compared to intraclass variability. It is outside the scope of this paper to review all relevant techniques useful for a traffic surveillance system; therefore we concentrate on giving a brief overview of different traffic video surveillance systems and various vehicle classification approaches proposed in the literature.

2.1 *Existing Traffic Surveillance Systems*

There are very few visual traffic surveillance systems described in the literature that perform a full array of tasks including detection, tracking, classification, and event detection. Gupte et al. developed a vehicle detection and two-type classification system by robustly tracking vehicles after camera calibration.[14] The VISTRAM system[15] classified vehicles into a small set of size-based classes and generated traffic parameters without explicit tracking, but the system did not include any type of event recognition. Kumar et al.[16] developed a parking lot monitoring system that tracked objects and classified them into six types using a known Bayesian network. The vehicle behavior at checkposts was evaluated based on a vocabulary of actions, allowing the detection of abnormal events such as loitering. A zone of influence was defined to represent potentially dangerous interactions between objects. SCOCA[17] is an intersection monitoring system that tracks and performs three-dimensional (3-D) model-based classification of objects. The speed of each vehicle is recorded along with its origin-destination information. In Ref. 18, Morris and Trivedi presented a VECTOR, which showed the ability to perform real-time vehicle classification, traffic statistic accumulation, and highway modeling for flow analysis. Buch and colleagues[19,20] presented two different approaches for vehicle classification in their visual traffic surveillance system, and the approach based on 3-D extended histogram of oriented gradients (3-D HOG) shows some of the best classification accuracy on a real traffic video. However, the approach is not suitable for real-time processing. Another approach[8] presented a traffic surveillance system that can detect, track, and classify vehicles. Although the work presented in this paper shares its goals with Ref. 8, the approach used for classification is entirely different.

2.2 *Existing Vehicle Classification Techniques*

In this section, we provide details of different algorithms used for vehicle classification. However, not all algorithms try to perform classification in a traffic video and limit their analysis to images of already segmented vehicles. The approaches for vehicle classification can be broadly categorized into four groups.

2.2.1 *3-D model–based approaches*

Three-dimensional model-based approaches have been proposed for the purpose of object detection and tracking in Refs. 8, 21, and 22. In Ref. 8, a region of interest (ROI) was extracted using statistical background modeling and extraction of foreground using background subtraction. Edges were detected using either the Sobel edge detector or the Canny edge detector. Three-dimensional wire-frames of the models in the database are projected onto the image and the best match is found based on the best matching pixel position,[23] or mathematical morphology to match the model to the edge points.[8] All the models are subjected to the matching process and the one with the highest matching score (i.e., lowest matching error) is selected as the model. In Ref. 17, Messelodi et al. generated the convex hull for 3-D vehicle models in the image and used them to estimate the ratio (a matching score) between the convex hull overlap of the model and the image normalized by the union of both areas. Similar 3-D vehicle models are matched with a motion-segmented input video in Ref. 24 for detection and in Ref. 20 for classification. In Ref. 25, Johansson et al. presented an extension to the approach that also takes into consideration the size of vehicles. These methods require camera parameters to be calibrated so that a 3-D wire-frame can be projected onto an image. They also need orientation of the vehicles, which can be retrieved from optical flow calculation.

2.2.2 *Image measurement-based approaches*

In these approaches, different features of detected objects are extracted instead of using direct images or image patches. Gupte et al.[14] proposed a system for vehicle detection and classification. They classified the tracked vehicles into two categories: cars and noncars. The classification is based on vehicle dimensions, where they compute the length and height of a vehicle and use it to distinguish cars from noncars.[14] Morris and Trivedi used 17 different region features, including seven moments for seven classes of road users.[26,27] A comparison between image-based features (e.g., pixels) and image measurement (IM) features (e.g., region area) is given. Both feature types are used with PCA and LDA as dimensionality reduction techniques. IM with LDA produced the best performance and was used for the final algorithm. The features were classified using a weighted k nearest-neighbor algorithm.

2.2.3 *PCA-based approaches*

Chunrui and Siyal developed a new segmentation technique for classification of moving vehicles.[28] They used simple correlation to get the desired match. The results shown in the paper are for the lateral view of the vehicles and no quantitative results were given. Toward this goal, a method is developed by Zhang et al.[13] In their work they used a PCA-based vehicle classification framework. They implemented two classification algorithms—eigenvehicle and PCA-SVM—to classify vehicle objects into trucks, passenger cars, vans, and pickups. These two methods exploit the distinguishing power of PCA at different granularities with different learning mechanisms. The eigenvehicle approach used in Ref. 13 is similar to the proposed approach PCA +DIVS. However, we use distance from mean image in PCA space instead of finding distance from each image from each class as done in Ref. 13. Also, they use a single-look approach and do not aggregate results using tracking.

2.2.4 *Local feature-based approaches*

Local features have certain advantages over using global features as they are better suited to handle partial occlusion. In traffic surveillance, if intersection monitoring is desired, then overlapping of passing vehicles will result in partial occlusion and errors in extracting ROIs. Scale-invariant feature transform (SIFT)[29] has shown to outperform other local features in terms of repeatability.[30] Ma and Grimson developed a vehicle classification approach using modified SIFT descriptors.[12] They used SIFT features to train the constellation models that were used to classify the vehicles. They considered two cases: cars versus vans and sedans versus taxis. They reported good results for the difficult case of classifying sedans versus taxis. However, they do not report combined classification results for sedans versus vans versus taxis, which will show the scalability of the approach. We implemented a constellation model-based approach that differs slightly from Ref. 12, but we were able to achieve similar accuracy with better computational complexity, on the same dataset as in Ref. 12.

2.2.5 *Other approaches*

Huang and Liao[31] used a hierarchical classification scheme. Initially, coarse classification identifies a moving object as a large vehicle or a small vehicle and subsequently finer classification is performed to classify the vehicle into seven categories. Ji et al. used a partial Gabor filter approach.[32] In Ref. 33, Wijnhoven and de With presented a patch-based approach that uses Gabor-filtered versions of the input images at several scales. The feature vectors were used to train an SVM classifier, which was able to produce better results than those presented in Ref. 12 for the case of cars versus vans. However, this approach is global feature based; therefore it is not best suited for cases with partial occlusion. Recently, Buch et al. presented a traffic video surveillance system that employs motion 3-D HOG to classify road users.[19] However, both approaches are computationally expensive and cannot be employed in a real-time system.

3 Vehicle Classification Techniques: a Comparison

The problem of face detection can be considered as a two-class classification when we deal with face versus nonface classification. In this research, we are interested in classifying vehicles in multiple classes, and we do so by extending the eigenface approach.[34] The components extracted from PCA are the MEFs, while LDA uses the MDFs. The constellation model is a generative model that models scale invariant features to distinguish between different classes of vehicles. As the constellation model is a part-based model, it can perform well even in the presence of partial occlusion.

3.1 *Eigenvehicle Approach (PCA + DFVS)*

In Ref. 34, PCA was used for single-class classification (i.e., face). We use it for up to three classes at the same time and therefore extend the approach by creating a separate PCS (vehicle space) for each class. We define each eigenspace as eigenvehicle.[13]

3.1.1 *Training for eigenvehicles*

For creating the PCS for each class (i.e., creating an eigenvehicle for each class), we normalize the images such that the width and height of all the images are the same. Since each sample image is a two-dimensional image $A_i \in R^{m\times n}$, we create a vector from an image by concatenating rows to create a column vector $A_i' \in R^{1\times mn}$. We consider $k = 50$ images for each class, and we build a matrix of k columns $[A_1' A_2' A_3' \ldots A_k']$ that represents the set of training samples. The length of each column is $m \times n$. Then, we can compute the mean vector μ as shown below:

$$\mu = \frac{1}{k}\sum_{i=1}^{k} A_i' \tag{1}$$

Let $\sigma_i = A_i' - \mu$ and $\sigma = [\sigma_1 \sigma_2 \sigma_3 \ldots \sigma_k]$. The covariance matrix of A' is

$$C = \frac{1}{k}\sum_{i=1}^{k} \sigma_i \sigma_i^T = \sigma\sigma^T. \tag{2}$$

The eigenvectors of C are the principal components. The eigenvectors associated with the largest eigenvalues correspond to the dimensions in the space where the data have the largest variance. In our training set, the size of C is $mn \times mn$ (3182×3182), which is not feasible to compute principal components. In Ref. 34, Turk and Pentland propose a solution to this problem, where they find the eigenvectors and eigenvalues of $\sigma^T\sigma$, instead of $\sigma\sigma^T$. Suppose v_i is an eigenvector of $\sigma^T\sigma$ and λ_i is the associated eigenvalue. Then

$$\sigma^T\sigma v_i = \lambda_i v_i \overset{\text{yields}}{\rightarrow} \sigma\sigma^T\sigma v_i = \lambda_i \sigma v_i. \tag{3}$$

The above deduction shows that σv_i is an eigenvector of $\sigma\sigma^T$. This technique reduces the computation complexity since the dimension of $\sigma^T\sigma$ is only $k\times k$ (50×50). We are able to extract the top k principal components of $\sigma\sigma^T$ by the following equation:

$$u_i = \sigma v_i. \tag{4}$$

The eigenvectors corresponding to the largest eigenvalue represent the most dominant dimensions or features of the images in a class. The length of each eigenvector is $m \times n$. Therefore, each of these eigenvectors can be rearranged as an image that we call an eigenvehicle. As we use 50 sample images from each class during the creation of eigenvehicles, we have 50 eigenvehicles for each class. However, not all the eigenvehicles need to be used during classification.

3.1.2 *Classification using eigenvehicles*

Classifying a new image in one of the classes is carried out in three steps. First, we reshape A_{new} into A'_{new} such that the width and the height of the image are normalized. We then obtain $\sigma_{\text{new}} = A'_{\text{new}} - \mu$. Second, we project σ_{new} onto eigenvehicle space, i.e., the PCS created. Traditionally, this space has been called the face space. This process yields the k weights w_i where

$$w_i = u_i^T \sigma_{\text{new}}. \tag{5}$$

We choose the first l weights where $l < k$ and back-project to get an image A''_{new}

$$A''_{\text{new}} = \sum_{i=1}^{l} w_i \sigma_i + \mu. \tag{6}$$

The image A''_{new} is subtracted from the original test image A'_{new} to find the Euclidean distance, i.e., DFVS, which is essentially a back-projection error.

$$\text{DFVS} = \sqrt[2]{\sum_{i=1}^{m\times n} (A''_{\text{new}_i} - A_{\text{new}_i})^2}. \tag{7}$$

This is done for every class that yields a new A''_{new}. The class related to the PCS that results in the smallest DFVS is assigned as the class of the test image. We tried to use a different number of principal eigenvectors to assess the dependence of accuracy on the number of eigenvectors used. The detailed results are discussed in Sec. 5. This approach has an ability to perform well in the case of low interclass variability (e.g., sedans versus taxis).

3.2 *PCA+DIVS*

In this approach, we start by employing PCA as described in the eigenvehicle approach with a slight modification. We create a PCS (vehicle space) for all the training samples irrespective of the class label. Therefore, there is only one PCS as opposed to the previous approach where we created a separate PCS for each class. All training images irrespective of class label are used to calculate a covariance matrix C whose eigenvectors define a single PCS. Then, all training images in a class C ($c \in \{1, 2\}$ in the two-class case) are projected onto the PCS and weights are calculated. The mean weight vector (principal component) w^c_{mean} for each class is calculated using the first l weights that belong to the eigenvectors with the largest eigenvalues ($l < k$, where k is the total number of training sample images in all the classes combined, k^c is the number of training samples in a class c, and l will be the dimension of w^c_{mean}):

$$w^c_{\text{mean}} = \frac{1}{k^c} \sum u^T . \sigma^c_{\text{train}}. \tag{8}$$

For testing, a test image is projected on the PCS to get the weight vector (principal component) w with l dimensions, where components of w are calculated using

$$w_i = u_i^T \sigma_{\text{new}}. \tag{9}$$

We calculate the Mahalanobis distance $d^c_{\text{Mahalanobis}}$ from the mean principal component w^c_{mean} of each class.

$$d^c_{\text{Mahalanobis}} = \sqrt{(w_i - w^c_{\text{mean}})^T C^{-1} (w_i - w^c_{\text{mean}})}. \tag{10}$$

The smallest distance decides the class of the test image.

3.3 *PCA+SVM*

In this approach, we used the approach described in Sec. 3.2 to create the PCS. However, instead of finding the distance from the mean principal component of each class, we train PCA vectors using an SVM with a radial basis function (RBF) kernel.[35] The main objective of the SVM training is to find the largest possible classification margin, which indicates the minimum value of w in

$$\frac{1}{2} w^T w + E \sum \epsilon_i, \tag{11}$$

where $\varepsilon_i \geq 0$ and E is the error tolerance level. The training vectors are grouped in labeled pairs $L_i(x_i, y_i)$, where x_i is a training vector and $y_i \in \{-1, 1\}$ is the class label of x_i and are used in training the SVM that finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. We used four-fold cross-validation and tried different values for bandwidth to find the best parameters for SVM that minimize the cross-validation estimate of the test error. There are only 50 training examples of each class, and training SVM using such a small set of training examples is a challenge. To overcome this problem, we trained the SVM by using the same set of training examples repeatedly through a parameter optimization process, which tries to improve the classification accuracy by choosing the parameters (bandwidth for the RBF kernel, the error tolerance level, and the choice of support vectors, etc.). The overall accuracy of the trained SVM classifier also depends on the choice of number of eigenvectors used. If we use the first 20 values of a weight vector w_i [Eq. (9)] that correspond to the largest 20 eigenvalues, we can determine the amount of information retained by the first 20 values of weight vector by calculating the ratio

$$\text{Ratio} = \frac{\sum_{i=1}^{20} \lambda_i}{\sum_{i=1}^{100} \lambda_i}, \tag{12}$$

where λ_i is the associated eigenvalue of the PCS. The ratio was 0.862 for the case of cars versus vans, while it was 0.871 for the case of sedans versus taxis. As we use more dimensions of a weight vector, we retain more information. However, this does not always reflect directly into better SVM classification accuracy.

For testing, a test image is projected on the PCS and then the corresponding principal component is classified using the trained SVM. The choice of kernel, the size of training set, bandwidth selection, and number of eigenvectors used play a major role in the efficiency of SVM training and accuracy of the results.

3.4 *LDA*

Approaches based on PCA use the MEFs to classify new images. However, MEFs are not always the MDFs. LDA automatically selects the features that provide an effective feature space to be used for classification.[33]

To eliminate the problem of high dimensionality, we start by employing PCA as described in Sec. 3.2, where all the images irrespective of class label are projected onto a single PCS. The dimension of the PCS will be limited by the total number of training images minus the number of classes. The LDA involves calculating two matrices: the within-class scatter matrix S_W and the between-class scatter matrix S_B.

$$S_W = \sum_{i=1}^{C}\sum_{j=1}^{M_i}(y_j - \mu_i)(y_j - \mu_i)^T, \tag{13}$$

$$S_B = \sum_{i=1}^{C}(\mu_i - \mu)(\mu_i - \mu)^T, \tag{14}$$

where C is the number of classes, μ_i is the mean vector of a class i, and M_i is the number of samples within class i. The mean of all the mean vectors is represented by μ and is calculated as

$$\mu = \frac{1}{C}\sum_{i=1}^{C}\mu_i. \tag{15}$$

LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter by maximizing the following ratio: $\det|S_B|/\det|S_W|$. The advantage of using this ratio is that it has been proven in Ref. 36 that if S_w is a nonsingular matrix, then this ratio is maximized when the column vectors of the projection matrix W are the eigenvectors of $S_W^{-1}S_B$. W with dimension $C-1$ projects the training data onto a new space called fisherfaces. We use W to project all training samples onto fisherfaces. The resulting vectors are used to create a KD-tree, which is employed in finding the approximate nearest neighbors during the classification of a sample image. We use five nearest neighbors and the class with the highest number of nearest neighbors is assigned as the class of the vehicle.

3.5 *Constellation of SIFT Features*

Object recognition techniques that generally work well for object classification are not directly useful in the case of object categorization when interclass variability is low. The problem of vehicle classification is different from many other object classification problems[37] where the difference between object classes is considerable (e.g., airplane versus motorcycle). Surveillance videos pose other problems, for example, surveillance image sizes are generally small and captured images can have varying lighting conditions. Affine invariant detectors have been shown to outperform simple corner detectors in the task of object classification.[38] We tried two interest point detectors: Harris–Laplace with affine invariance and LoG with affine invariance. The number of interest points detected using these techniques is small and may not provide enough information to classify an image successfully.

In this work, we employed a constellation model-based approach that uses the same techniques as presented in Ref. 12 with a few modifications. In our implementation, we extend the approach to do the multiclass classification and use k-means clustering instead of mean-shift clustering to improve the computational complexity. Ma and Grimson used a single Gaussian to model the features and a mixture of Gaussians (MoG) to model feature positions.[12] However, in our implementation, we model both features and feature positions as independent MoGs that considers up to six Gaussians and choose the number of Gaussians that maximizes the maximum likelihood for training data.

3.6 *Fusion of Approaches*

We described five approaches that can be used in combination with each other and improve the classification accuracy. The fusion of approaches becomes more important when the number of classes increases. In Sec. 5 we present results using all these approaches showing that some of them are better suited for a certain classification task, e.g., PCA +DIVS works well for the case of cars versus vans, while PCA+DFVS works well for the case of sedans versus taxis. As explained earlier, sedans and taxis are disjoint subsets of cars. Therefore, we train two classifiers where the first classifier uses PCA+DIVS and classifies a test image into cars and vans. The test images that were classified as cars are further classified into sedans and taxis using the second classifier that employs PCA+DFVS. The fusion of different methods is thus possible and yields better results than just using a single approach.

4 Visual Traffic Surveillance Framework

In this section, we briefly discuss different parts of our traffic video surveillance framework. Figure 2 shows the components of the framework in the form of a block diagram.

Camera calibration is an important part of computer vision systems, which is responsible for finding intrinsic and extrinsic camera parameters, which in our case can be used to perform ground plane rectification, i.e., if a pixel in the image appears on the ground plane, its 3-D coordinates can be found in the world reference frame. We use the same technique as described in Ref. 8 to extract camera parameters from a traffic scene. Background modeling and foreground object detection attempt to detect the moving objects (blobs) in a traffic scene. The detected blobs are used as candidates in the vehicle detection and classification stage. To collect any meaningful information from the sequence of images, it is important that we should be able to match the objects detected in consecutive frames. This part of the system tracks the detected objects (blobs). It also tries to correct possible errors from the foreground object detection module and keeps record of the tracks and their 3-D world coordinates in each frame if available. Assuming that vehicles tend to move in the forward direction, vehicle tracking results are aggregated to infer the pose of a vehicle. The classification can greatly benefit from the knowledge of the pose. As discussed in the previous section, we implemented five different algorithms to effectively perform vehicle classification. Even though certain parts of the framework show dependence on other parts, this dependence is based on the type of algorithm chosen for a particular part. For example, if the constellation-based classifier[12] is used in the vehicle classification stage, we will need to perform edge detection. On the other hand, if vehicle speed information is not required in a particular scenario, the camera calibration stage is optional.

For the brevity of the paper, we avoided providing details about each part of the framework. Interested readers may consult Ref. 8, which describes in more detail the approaches

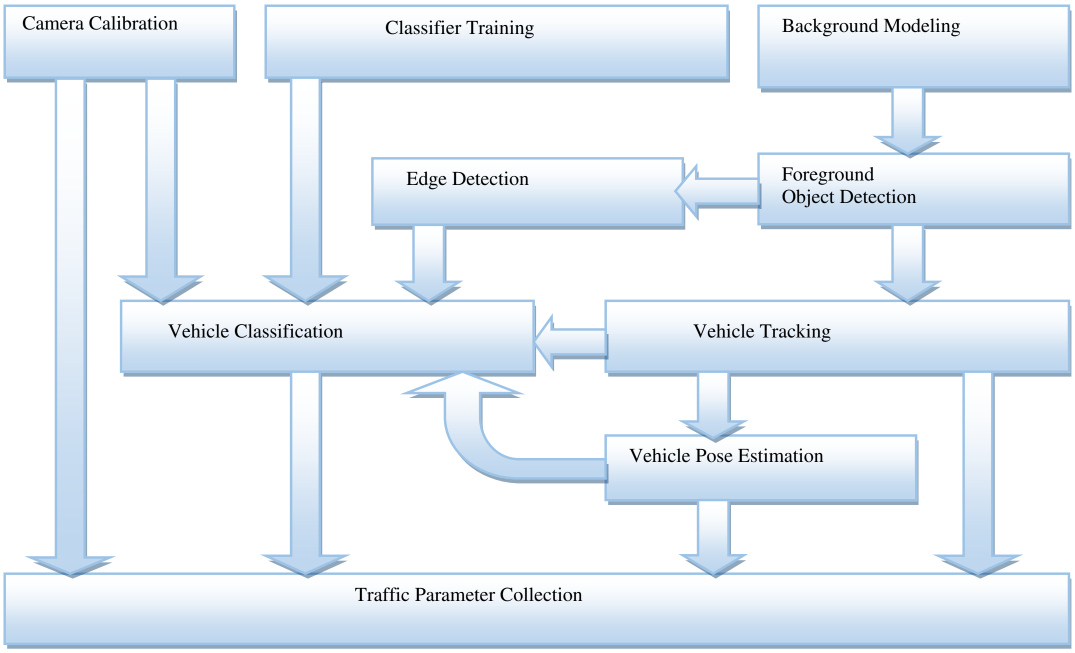


Fig. 2 Overview of our visual traffic video surveillance framework.

used for the camera calibration, vehicle tracking, and pose estimation.

5 Results

In this work we have considered five different approaches for the vehicle classification module. This section provides details about the experimental setup used during testing, the effect of different parameter choices on the results, and the comparison between different approaches.

5.1 *Comparison of Vehicle Classification Approaches*

5.1.1 *Experimental setup*

As mentioned in the previous section, vehicle classification still remains a challenging problem, and the choice of the algorithm will dictate the overall accuracy of the framework. However, the lack of standardized video sequences makes it difficult to qualitatively differentiate between different vehicle classification approaches. Therefore, we devised the first experiment that compares different vehicle classification techniques presented in the previous section. We employed a dataset used in Ref. 12, which has very limited training examples. This is important, as getting ground truth data is generally expensive and a classification technique that can be retrained by using only limited samples is important for real-world scenarios that might require retraining in the field. The dataset in Ref. 12 defines two classification tasks: cars versus vans and sedans versus taxis. Previously published results that used the same dataset do not consider a multiclass classification case. Therefore, we combined the dataset in Ref. 12 and created a more complex task of classifying three types of vehicles: sedans, vans, and taxis. Sedans and taxis are the disjoint subsets of class cars. Taxis differ from sedans in a very minor way in that they carry a characteristic taxi cab sign on top and are generally full-size sedans. The dataset provided in Ref. 12 has 50 images of each class for training and 200 images each of cars, vans, and sedans and 130 images of taxis for testing. The images in the dataset have different sizes and therefore are not suitable for PCA directly. We normalize all the images to average width and height (74 × 43 × pixels). For the case of cars versus vans, we use 50 images from each class for training and 200 images of each class for testing. For the case of sedans versus taxis, we use 50 images from each class for training and 200 images of sedans and 130 images of taxis for testing. We use the same experimental setup as in Refs. 12 and 33, so that a fair comparison is performed. In the case of sedans versus vans versus taxis, we use 50 images of each class for training and 200 images of sedans, 200 images of vans, and 130 images of taxis for testing.

5.1.2 *Results*

In this paper, we used five different approaches to classify vehicles. The dataset that we used contains the images of vehicles taken from a surveillance video camera and segmented using a tracking algorithm.[39] The images were taken such that vehicles are captured in a more general oblique view instead of side or top view. We compare our approaches with the approaches presented in Refs. 12 and 33 that use the same dataset. We observed that changing the number of eigenvectors used does not change the accuracy of PCA-based approaches (PCA+DIVS, PCA+DFVS, LDA) greatly. However, the same cannot be said about PCA + SVM, which was found to be sensitive to the number of eigenvectors used in training the SVM. We also observed that our approach PCA+DFVS outperforms all other approaches in the case sedans versus taxis, while our approach PCA+DIVS outperforms the rest in the case of cars versus vans. In the case of sedans versus vans versus taxis, the proposed fusion of approaches (PCA+DIVS and PCA+DFVS) gives the best results.

The constellation model-based approach presented in this paper gives performance benefits by using k-means clustering over mean-shift. It also has an advantage over all other approaches presented in this work in that it has an ability to handle partial occlusions owing to its reliance on local features rather than global features. Our constellation model-based approach gives results comparable to the constellation model-based approach presented in Ref. 12 for the cases of cars versus vans and sedans versus taxis. In this work, we extended the constellation model-based approach to handle the multiclass case. We can observe that the accuracy decreases while doing multiclass classification, which can be attributed to an increased number of common features as the number of classes increases.

Table 1 shows the accuracy achieved using each approach and the approaches that yielded the best results are marked in bold. The accuracy is defined as the number of vehicles classified correctly divided by the total number of vehicles in the test dataset. The first seven rows of Table 1 provide the results obtained using techniques investigated in this paper. The last two rows of Table 1 give the results obtained by the state-of-the-art techniques in vehicle classification when applied to the same dataset, as shown in Refs. 12 and 33, respectively, and use the same experimental setup as presented in this paper. However, these techniques do not extend to perform multiclass classification. The bold values in Table 1 signify the highest accuracy achieved for a particular classification task

5.2 *Visual Traffic Surveillance System*

In the previous section we presented the results of a comparison between five different object classification techniques for the purpose of vehicle classification. The effort was directed to find a technique that can produce good accuracy even when interclass variability is small (e.g., sedans versus taxis). However, the results were obtained using a dataset of images that contains well-segmented images with a vehicle covering the center of the image. This assumption cannot be made in a real-world traffic surveillance system where classification is generally toward the end of the video surveillance pipeline (see Fig. 2) and therefore it needs to deal with errors introduced in the previous stages, i.e., foreground segmentation errors or tracking errors (for example, due to partial occlusion).

We incorporated in the unified video traffic surveillance system three of the most promising techniques (PCA + DFVS, PCA+DIVS, and the constellation model with explicit shape) that were discovered during our comparison of the object recognition approaches and tested our system in terms of its vehicle classification performance directly from video.

5.2.1 *Experimental setup*

The lack of publicly available standardized video datasets to test and compare the vehicle traffic surveillance system prompted us to record our own video so that the proposed system can be quantitatively evaluated. We used a Samsung HD Camcorder fixed on a tripod, which in turn was fixed on an adjacent parking structure of two-storey height to record the traffic video with 1280 × 720-pixel resolution. We recorded 21 min of video that was used for training the vehicle classifier, and after a gap of 10 min, we recorded another 31 min of video for testing purposes. We used both video sequences for testing in order to assess the generality of our classifiers. Each video is recorded with 60 fps, but downsampled to 30 fps before processing. We also downsampled the video to 640 × 360 pixels while processing to reduce the computation time.

Figure 3 shows a snapshot from a video sequence used. It is an intersection with one major street intersecting a minor street. As we use appearance-based classifiers, we need to train different classifiers for different orientations. As most of the traffic is using the major street, we can use two classifiers: one for traffic coming toward camera, other for traffic going away from camera.

During the training phase, we process the 20 min of video using foreground object detection and tracking to find the bounding boxes for vehicles in each frame. This information and the video are used with the ground truth verification tool (GTVT)[40] to create the ground truth for the classes of all

Table 1 Comparison of approaches.

	Accuracy (%)		
	Cars versus vans	**Sedans versus taxis**	**Sedans versus vans versus taxis**
PCA+DFVS (eigenvehicle)	98.5	**97.57**	95.85
PCA+DIVS	**99.25**	89.69	94.15
PCA+SVM	94.50	91.92	
LDA	96	95.15	90.00
Constellation model (implicit shape)	96.25	89.39	85.66
Constellation model (explicit shape)	97	89.09	86.04
A fusion of approaches			**96.42**
Constellation model[12]	98.5	95.86	
Patch-based object classification[33]	**99.25**	95.25	

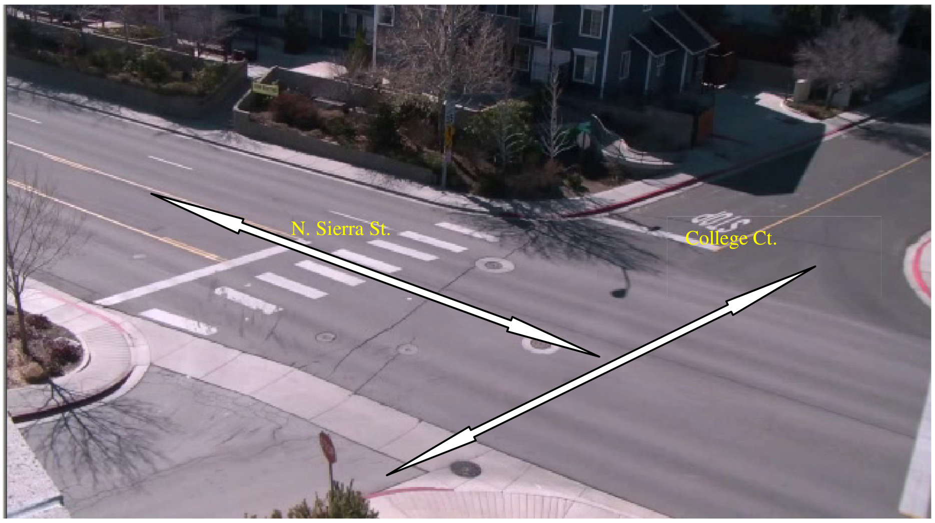


Fig. 3 A snapshot from video sequence.

detected vehicles. For the purpose of this experiment, we considered four classes: cars, SUVs/vans, pickup trucks, buses/semis. We refrained from using the class motorcycle as 20 min of the video that was used for training had only a single example of it; therefore, enough training samples could not be extracted from it. We use a two-stage classification process. In the first stage, the blobs are classified on the basis of size and very big blobs are classified as buses/semis. In the second stage, we use the appearance-based classifier to distinguish between cars, SUVs/vans, and pickup trucks. Table 2 provides details about the ground truth established for the training video using the GTVT.[40] In the training video, there were 101 cars, 123 SUVs/vans, 64 pickup trucks, 1 bus, and 1 motorcycle. The vehicles that are not motorcycles or buses/semis and clearly cannot be classified into cars or pickup trucks are classified as SUVs/vans while establishing the ground truth. That means we classify vehicles such as crossovers, small trucks (e.g., U-Haul), etc. as SUVs. We also establish the ground truth for tracking such as we record the number of tracks (vehicles) that were using the major street and were tracked correctly, the number of turning vehicles, and the tracking failures. Table 3 illustrates the tracking results for the training video, which show that the tracking accuracy of our video surveillance system is 94.51% for this video sequence.

Table 2 Classification ground truth for the training video.

Class	Number of vehicles
Cars	101
SUVs/vans	123
Pickup trucks	64
Bus	1
Motorcycles	1
Total	291

Table 4 provides details about the ground truth established for the testing video. For testing there were 200 cars, 188 SUVs/vans, 69 pickup trucks, 5 buses/semis, and 2 motorcycles. The vehicles that are not motorcycles and buses/semis are classified as SUVs/vans while establishing the ground truth. We also establish the ground truth for tracking such as we record the number of tracks (vehicles) that were using major street and were tracked correctly, number of turning vehicles, and tracking failures. Table 5 provides details about the tracking results on the testing video, for which we report a tracking accuracy of 94.18%.

In Refs. 26 and 27, Morris and Trivedi improved the classification accuracy by >10 when combining classification with tracking. Therefore, we do not delineate a detection region where classification is done (single-look

Table 3 Tracking ground truth/results for the training video.

Correctly tracked and not turning	262
Correctly tracked and turning	13
Tracking failures	16

Table 4 Classification ground truth for the testing video.

Class	Number of vehicles
Cars	200
SUVs/vans	188
Pickup trucks	69
Bus	5
Motorcycles	2
Total	464

Table 5 Tracking ground truth/results for the testing video.

Correctly tracked and not turning	389
Correctly tracked and turning	48
Tracking failure	27

classification), but instead classification is done for every frame and for all the instances of detected objects using the major street and the results are aggregated using tracking (multilook classification). In Refs. 26 and 27, a highway-type of traffic was used to evaluate the approach and the camera was set up such that only side views are visible. This assumption leads to relatively low perspective distortion and small changes in the size and appearance of a vehicle as it moves from one side of the scene to the other. However, in a general-purpose system, this assumption cannot be made, and Fig. 4 shows examples of such change in appearance for a vehicle as it moves across the scene. Therefore, our system divides the scene in multiple regions and uses different classifiers for better classification accuracy. Figure 5 shows that the scene is divided into three regions and each region has two classifiers for two vehicle orientations. Therefore, we train a total of six separate classifiers. The video sequence used contains oblique-view vehicles and hence is more general than frontal or side views. If only frontal or side views are present in a particular scene, the user has a choice to define a lower number of separate classification regions and limit the number of classifiers.

After defining classification regions and establishing the ground truth using GTVT,[40] we randomly select 200 images (100 images per orientation/pose) of each class (three classes: cars, SUVs/vans, and pickup trucks) for each classification region. Figure 6 gives an example of the training samples used, where it can be seen that the vehicle images are of different sizes. We normalize them to the average width and height before using them in a classifier training algorithm.

For region 1, there are two classifiers: classifier1a for vehicles coming toward camera and classifier1b for vehicles going away from camera. In a similar manner, we define classifier2a, classifier2b, classifier3a, and classifier3b. Therefore, we trained six classifiers (two for each region) using three different classification techniques that performed the best in the evaluation performed in the previous section. The classifiers used are PCA+DFVS, PCA+DIVS, and the constellation model with explicit shape. After training the classifiers, we tested it on the same training images to see how well the classifier can distinguish between the vehicles. Finally, we incorporated the trained classifiers in our video traffic surveillance system to determine the classification

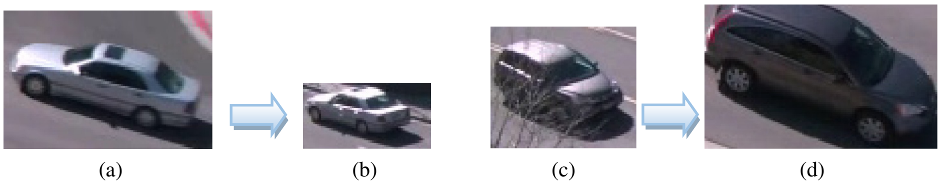


Fig. 4 Change in size and appearance of a vehicle as it moves across the scene. (a) Vehicle A entering the scene from right. (b) Vehicle A exiting the scene. (c) Vehicle B entering the scene from left. (d) Vehicle B exiting the scene.

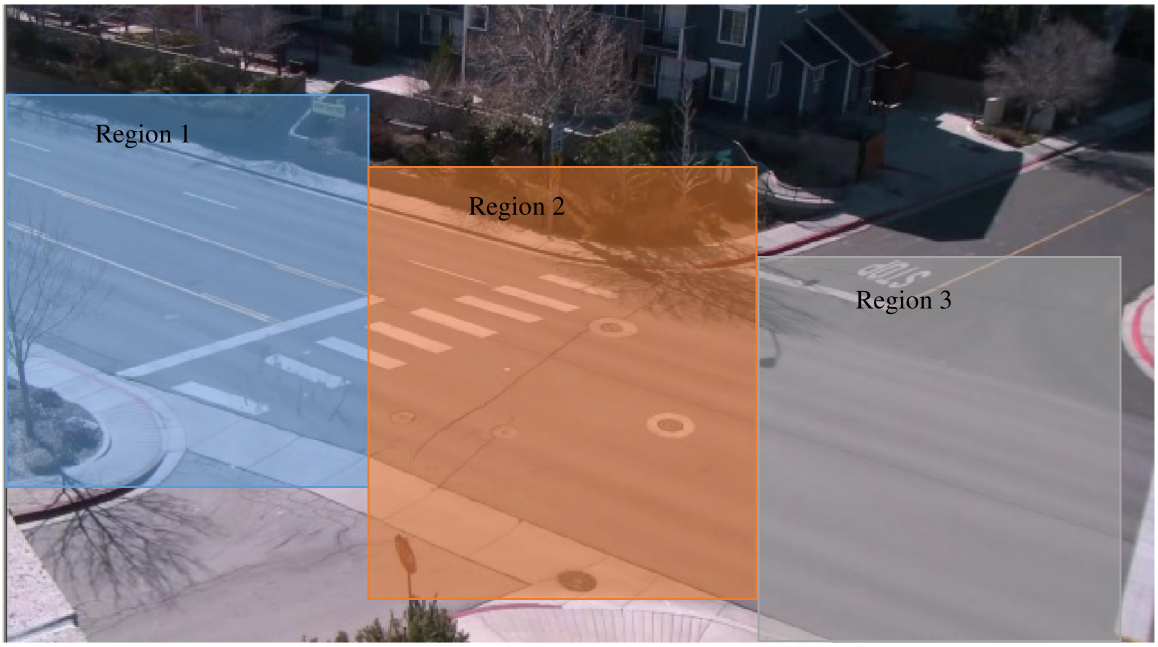


Fig. 5 Classification regions. Blue represents region 1. Red represents region 2. Green represents region 3.

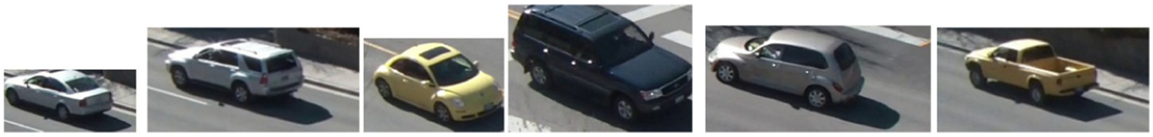


Fig. 6 An example of training samples used.

accuracy in a test video. The rest of this section discusses the results obtained and compares them with other systems in the literature.

5.3 *Results and Comparison*

As explained in the previous section, we tried three different classification techniques: PCA+DFVS, PCA+DIVS, and the constellation model with explicit shape.

5.3.1 *Results using PCA+DFVS*

After training the classifiers, we tested the accuracy using the same images that were used for training. Table 6 gives details of the accuracy achieved by each of the six classifiers while using the PCA+DFVS technique.

We used the trained classifiers in a unified traffic surveillance system to evaluate the vehicle classification framework. If we consider all the vehicles that were recorded in the ground truth, Table 7 gives the confusion matrix for the video sequence used for training. For this and all subsequent confusion matrix tables, the rows define the ground truth, whereas columns define the detected class. When the track appears for a very few frames (generally <6) due to turning or tracking failures, no class value is assigned. The overall classification accuracy on the training video was 85.88%. If we only considered vehicles that were detected and tracked correctly and not turning, then the accuracy improves to 91.60%. Table 8 gives the confusion matrix for correctly tracked nonturning vehicles.

It is important that the classifiers also perform well on a video sequence that contains vehicle image samples previously not seen. We used 31 min of a separate video sequence for testing the performance of our traffic surveillance system. We achieved a classification accuracy of 81.90%, if we consider all the vehicles in the ground truth. Table 9 gives the confusion matrix for this case. Table 10 provides the confusion matrix in the case when only nonturning and correctly tracked vehicles are considered. The classification accuracy in this case was 90.49%.

5.3.2 *Results using PCA+DIVS*

As discussed in the previous section, PCA+DIVS has produced remarkable results on the dataset provided in Ref. 12. We trained the required six classifiers as discussed in Sec. 3. Their performance on the training samples was evaluated before incorporating them into a unified traffic surveillance system. Table 11 gives the accuracy achieved by these classifiers.

We used the trained classifiers in the unified traffic surveillance system to evaluate the vehicle classification

Table 6 Accuracy on training samples (PCA+DFVS).

Name of the classifier	Accuracy (%)
Classifier1a	99.66
Classifier2a	100
Classifier3a	100
Classifier1b	100
Classifier2b	100
Classifier3b	100

Table 7 Confusion matrix for all vehicles in the video sequence used for training (PCA+DFVS).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	5	86	9	1	0
SUVs/vans	8	5	106	1	4
Pickups	5	2	2	54	1
Buses/semis	0	0	0	0	1

Table 8 Confusion matrix for all correctly tracked nonturning vehicles in the video sequence used for training (PCA+DFVS).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	0	83	8	0	0
SUVs/vans	1	5	104	0	1
Pickups	2	1	2	52	1
Buses/semis	0	0	0	0	1

Table 9 Confusion matrix for all vehicles in the test video sequence (PCA+DFVS).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	11	158	22	9	0
SUVs/vans	9	6	162	7	4
Pickups	4	2	3	56	4
Buses/semis	0	0	0	1	4

Table 10 Confusion matrix for all correctly tracked nonturning vehicles in the test video sequence (PCA+DFVS).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	0	149	19	3	0
SUVs/vans	0	4	151	3	2
Pickups	0	1	2	48	1
Buses/semis	0	0	0	1	4

Table 11 Accuracy on training samples (PCA+DIVS).

Name of the classifier	Accuracy (%)
Classifier1a	98.66
Classifier2a	98.33
Classifier3a	98.33
Classifier1b	99.66
Classifier2b	95
Classifier3b	94

framework. If we consider all the vehicles that were recorded in the ground truth, Table 12 gives the confusion matrix for the video sequence used for training. The overall classification accuracy on the training video was 81.44%. The accuracy increases to 87.40% if we only consider vehicles that were detected and tracked correctly and not turning. Table 13 gives the confusion matrix for correctly tracked nonturning vehicles.

We used the same classifiers to evaluate the performance on the test video sequence. If we consider all the vehicles that were recorded in the ground truth, Table 14 gives the confusion matrix for the test video sequence. The overall classification accuracy on the training video was 73.49%. If we only consider vehicles that were detected and tracked correctly and not turning, then the accuracy increases to 82.51%. Table 15 gives the confusion matrix for correctly tracked nonturning vehicles.

Table 12 Confusion matrix for all vehicles in the video sequence used for training (PCA+DIVS).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	6	77	13	5	0
SUVs/vans	8	4	102	6	4
Pickups	4	1	1	57	1
Buses/semis	0	0	0	0	1

Table 13 Confusion matrix for all correctly tracked nonturning vehicles in the video sequence used for training (PCA+DIVS).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	0	74	13	4	0
SUVs/vans	1	4	100	5	1
Pickups	1	1	1	54	1
Buses/semis	0	0	0	0	1

Table 14 Confusion matrix for all vehicles in the test video sequence (PCA+DIVS).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	11	141	15	33	0
SUVs/vans	9	1	142	29	7
Pickups	4	3	2	54	6
Buses/semis	0	0	0	1	4

5.3.3 *Results using the constellation model (explicit shape)*

Algorithms such as PCA+DFVS and PCA+DIVS use global image features, i.e., they consider an entire image patch for classification. The constellation model uses local features such as SIFT and is inherently more capable of handling partial occlusion. Therefore, we also implemented the constellation-based model (with explicit shape model) to evaluate its usefulness in the unified traffic surveillance system. We kept the same parameters as used in Sec. 3 so that the robustness of the algorithm can be judged. Initially, we trained six classifiers and tested them by classifying the training images. Table 16 gives the accuracy achieved by each classifier.

We incorporated the trained classifiers into our traffic surveillance system and evaluated the performance of the classifier on the video sequence that was used for creating the training samples.

Table 15 Confusion matrix for all correctly tracked nonturning vehicles in the test video sequence (PCA+DIVS).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	0	134	14	23	0
SUVs/vans	0	1	137	19	2
Pickups	0	2	2	46	2
Buses/semis	0	0	0	1	4

Table 16 Accuracy on training samples (constellation model).

Name of the classifier	Accuracy (%)
Classifier1a	90
Classifier2a	96
Classifier3a	81.66
Classifier1b	90
Classifier2b	75.33
Classifier3b	97.33

If we consider all the vehicles that were recorded in the ground truth, Table 17 gives the confusion matrix for the video sequence used for training. The overall classification accuracy on the training video was 72.85%. We achieved the accuracy of 77.86% only considering vehicles that were detected and tracked correctly and not turning. Table 18 gives the confusion matrix for correctly tracked nonturning vehicles.

The constellation model–based approach did not work as expected. Therefore, we did not perform further experimentation using this approach. The accuracy of the algorithm highly depends on different parameters such as the number of clusters, the edge detector's thresholds, etc. However, further experimentation with different thresholds may improve the results and can produce results comparable to the previous two approaches, i.e., PCA+DFVS and PCA+DIVS.

Table 17 Confusion matrix for all vehicles in the video sequence used for training (constellation model).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	5	85	5	6	0
SUVs/vans	10	5	69	36	4
Pickups	4	0	2	57	1
Buses/semis	0	0	0	0	1

Table 18 Confusion matrix for all correctly tracked nonturning vehicles in the video sequence used for training (constellation model).

	None	Cars	SUVs/vans	Pickups	Buses/semis
Cars	0	81	5	5	0
SUVs/vans	3	5	68	34	1
Pickups	1	0	2	54	1
Buses/semis	0	0	0	0	1

Table 19 Average time to process a sample image.

Classifier	Average time (ms)
PCA+DFVS	6.63
PCA+DIVS	0.792
Constellation model (explicit shape)	680

5.3.4 *Time complexity*

In this section we compare the three approaches (PCA +DFVS, PCA+DIVS, and the constellation model with explicit shape) discussed previously in Sec. 3. We observed that PCA+DFVS outperformed all other approaches. However, for the purpose of video surveillance, it is important that the algorithm should be computationally inexpensive. We profiled these approaches to find the average time required to process a training image. Table 19 shows the average time to process an image in the training dataset using all three approaches considered.

It can be observed from Table 19 that PCA+DIVS is computationally inexpensive, while the constellation model–based approach that requires the extraction of SIFT features for every edge point detected is computationally expensive and not suitable for real-time video surveillance applications. By using technologies such as compute unified device architecture (CUDA) or general-purpose computation on graphics processing unit (GPGPU),[41] the required time for the constellation model–based approach can be reduced to certain extent.

5.3.5 *Comparison with other approaches*

In this section, we compare the results obtained by our traffic surveillance system that employs the vehicle classification framework to other approaches in the literature and highlight the contributions of our work. Table 20 lists some of the most promising approaches for video surveillance. A more exhaustive list of such approaches is available in a review paper by Buch et al.[42]

In this work we presented a unified traffic surveillance system that appears to outperform many of the approaches described in the literature. It is difficult to perform a direct quantitative comparison as various approaches use different video sequences to validate their results. In Table 20 we limited the listing to the most promising approaches that use video sequences of at least 30 min. The lack of publicly available traffic videos with ground truth is a problem often faced by researchers in video surveillance. For a small video of 1 min with 30 fps and having on average four objects present per frame, the task of creating ground truth requires labeling 7200 individual objects. The next section discusses a tool that can help in finding interesting events in a traffic video.

5.4 *Search Assistant Tool*

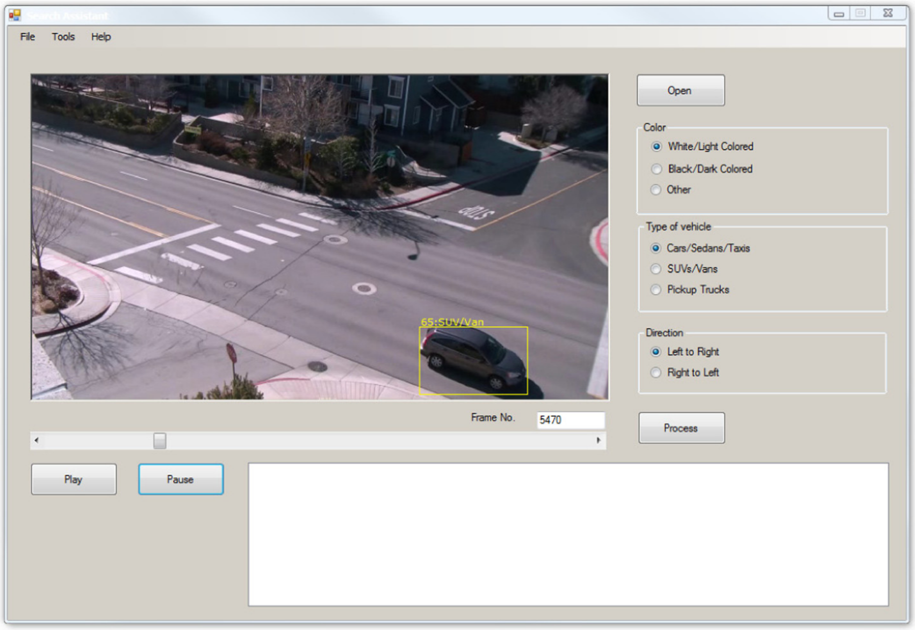
As the number of cameras performing video surveillance increases, it is becoming important that automated systems for surveillance be employed to process the video data online in real-time. However, limitations of computing power and

Table 20 Comparison with other approaches (the last row corresponds to our system).

Urban/ highway	Real time	Ref	Algorithm	Performance	Comments
Urban	Yes	17	3-D convex hull matching for eight vehicle classes	91.5% classification accuracy on correctly detected vehicles, overall 82.8%	Restricted view with delineated detection region. High mounted camera to limit partial occlusion. Vehicle size-based classification cannot handle classes such as SUVs and pickup trucks. No appearance-based information was used. Requires camera calibration for classifier to work.
Urban	Yes	43	3-D model-based vehicle tracking	65% classification accuracy	15-min test video was used.
Urban	Yes	20	3-D model matching against Gaussian mixture model (GMM) silhouettes	89.8% classification accuracy on correctly detected tracks	1-h video was used. It cannot handle classes such as pickup trucks versus SUVs that have similar sizes. Requires camera calibration for classifier to work.
Urban	No	19	Appearance model: 3-D HOG to classify road users	92.1% classification accuracy on correctly detected tracks	1-h video was used. Requires camera calibration for classifier to work. Computationally demanding.
Highway	Yes	27	Single Gaussian background model with size-based vehicle classifier. Uses LDA+ K-nearest neighbor (KNN).	88.4% classification accuracy on 2-h video	Only side view of vehicles, with very high mounted camera. Highway scenes generally produce better results than urban traffic scenes.
Urban Our approach with PCA+DFVS	Yes		Classification using PCA +DFVS and result aggregation using tracking	Classification accuracy of 91.6% on a training video of 20 min, and 90.49% classification accuracy on a test video of 31 min	Does not assume a particular camera angle. Camera calibration is not required for successful classification. Can handle multilane traffic with cases of partial occlusion. Also performs traffic parameter collection and records parameters such as speed and appearance of a vehicle.

requirements for processing the video on a need-to-know basis will still require tools that can process the recorded videos to find relevant events. In the case of traffic surveillance, it may be relevant to search for events based on certain visual attributes such as "a white truck coming toward the camera" or "black car moving away from the camera." However, searching for such events manually will be not only time consuming but also expensive. Therefore, we describe a tool that can automate this process by helping the user to find instances of an event based on visual attributes.

Content-based video indexing and retrieval have a wide range of applications such as quick browsing of video folders, remote instruction, digital museums, news event analysis, intelligent management of web videos (useful video



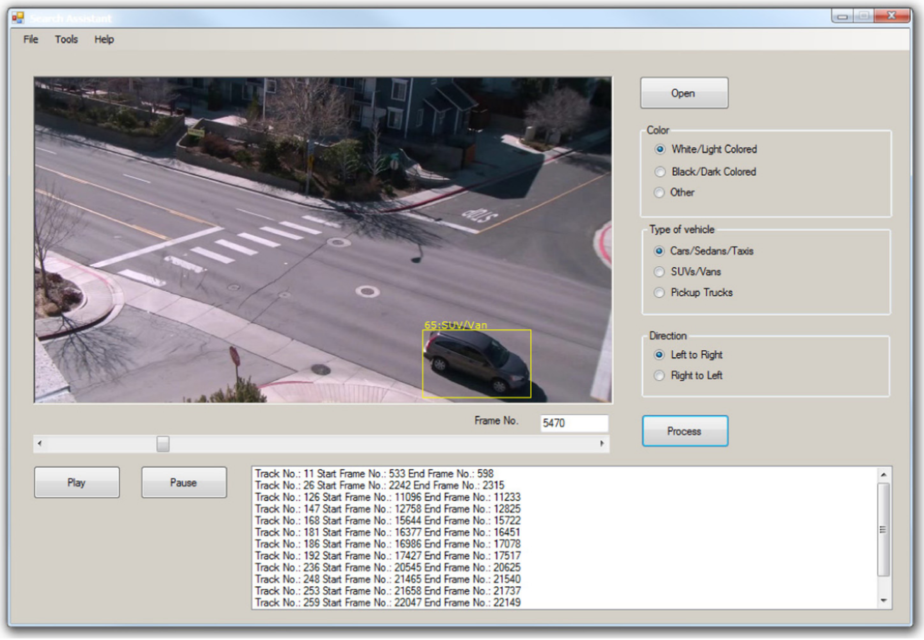
Color
● White/Light Colored
○ Black/Dark Colored
○ Other

Type of vehicle
● Cars/Sedans/Taxis
○ SUVs/Vans
○ Pickup Trucks

Direction
● Left to Right
○ Right to Left

Frame No. 5470

Fig. 7 Screenshot of search assistant tool (SAT).



Color
● White/Light Colored
○ Black/Dark Colored
○ Other

Type of vehicle
● Cars/Sedans/Taxis
○ SUVs/Vans
○ Pickup Trucks

Direction
● Left to Right
○ Right to Left

Frame No. 5470

```
Track No.: 11 Start Frame No.: 533 End Frame No.: 598
Track No.: 26 Start Frame No.: 2242 End Frame No.: 2315
Track No.: 126 Start Frame No.: 11096 End Frame No.: 11233
Track No.: 147 Start Frame No.: 12758 End Frame No.: 12825
Track No.: 168 Start Frame No.: 15644 End Frame No.: 15722
Track No.: 181 Start Frame No.: 16377 End Frame No.: 16451
Track No.: 186 Start Frame No.: 16986 End Frame No.: 17078
Track No.: 192 Start Frame No.: 17427 End Frame No.: 17517
Track No.: 236 Start Frame No.: 20545 End Frame No.: 20625
Track No.: 248 Start Frame No.: 21465 End Frame No.: 21540
Track No.: 253 Start Frame No.: 21658 End Frame No.: 21737
Track No.: 259 Start Frame No.: 22047 End Frame No.: 22149
```

Fig. 8 Screenshot of SAT after processing a query.

search and harmful video tracing), and video surveillance. Interested readers can refer to Ref. 44 for a survey on visual content-based video indexing and retrieval.

In the case of the search assistant tool (SAT) proposed in this work, we start by processing a given traffic video using the traffic surveillance approach described in Sec. 3 and record the traffic parameters. These recorded traffic parameters are then used as an input to the SAT. Figure 7 shows a screenshot of SAT after loading a traffic video and the relevant track parameter information. Figure 8 shows a screenshot of SAT after processing a query, "White cars moving left to right." The listbox at the bottom is populated by all instances (as video subsequences within the large video being queried) found for this event, which can be shown by clicking on each of them. A direct application of this tool can be envisioned in the context of law enforcement, where a witness may only remember a few visual attributes of a suspect. Assuming that a very large surveillance video is available, this SAT can be used to query the video based on those attributes in order to obtain just the relevant instances (small video sequences) to be presented to the witness for further investigation.

6 Conclusion

In this work we describe an integrated system for video traffic surveillance that can robustly and efficiently detect, track, and classify vehicles. We have investigated and compared five different approaches for vehicle classification. Using the PCA+DFVS (eigenvehicle) approach, we were able to achieve an accuracy of 97.57% in the challenging case of sedans versus taxis, which is higher than any published results using this dataset. PCA+DIVS outperformed all other approaches investigated in this paper in the case of cars versus vans. We also extended the constellation model approach[12] for classifying all three vehicle classes at the same time. LDA performed reliably, but did not produce the best results in any of the cases we experimented on. PCA+SVM showed its utility in the task of vehicle classification, but it was observed that the method is sensitive to the choice of training parameters and the number of eigenvectors used. Overall, PCA+DFVS approach achieves good results. However, the constellation model-based approach can be configured to work better in the presence of partial occlusion and minor rotations. We also presented a fusion approach that combines two classifiers and achieves improvements over using just one approach. We report an accuracy of 96.42% in case of sedans versus vans versus taxis using a fusion of approaches. We can use the SIFT-PCA features to train the constellation models. Also, features other than SIFT, such as LoG affine regions, can be used for modeling. The performance of the constellation model deteriorates as we extend it to multiple classes. A boosting algorithm can be used to choose the appropriate features for training.

After incorporating the vehicle classification framework into our unified traffic surveillance system, we achieved a classification accuracy as high as 90.49% on a 31-min test video sequence and 91.6% on a training video of 20 min. This was achieved by combining the classification results with the tracking results. We considered four vehicle classes: cars, SUVs/vans, pickup trucks, and buses/semis. We evaluated and compared three different classification techniques: PCA+DFVS, PCA+DIVS, and the constellation model, and found that PCA+DFVS produces the best results, while PCA+DIVS is the fastest of all three. Our classification framework performed considerably well considering that we impose no viewpoint restriction. Also, we were able to achieve a frame rate of about 6 fps while using PCA +DIVS. The SAT discussed in the previous section can be used in conjunction with the traffic surveillance system to extract instances of events described through visual attributes such as "a white pickup truck moving left to right."

The current classification framework requires that initial training examples must be provided by establishing the ground truth manually. GTVT[40] greatly reduces efforts required for establishing the ground truth, but it still is a cumbersome process. The same can be said about initial camera calibration, which is important for finding information such as the area of a blob in 3-D world coordinates and the speed of a vehicle. The camera calibration can be automated by using the Hough transform[45] to find lane structures on the street surface. The process of establishing the ground truth can be automated by using a 3-D model-based approach,[8] with strong thresholds (low false positives) to train an appearance-based approach such as PCA+DFVS for better accuracy. We tested our algorithm on a test sequence of 31 min, which is considerably longer than many other approaches that use video sequences as short as 2 min.[46] However, for a real-world scenario, it is important that the classification framework will need to be updated periodically to cope with illumination changes. The approach based on PCA is best suited for this purpose as incremental PCA[47] can be implemented to handle the changes gradually rather than retraining a model from scratch. We can also combine image feature-based approaches with appearance-based methods to achieve better performance.

Acknowledgments

This work has been supported by the Office of Naval Research, under grant number N00014-09-1-1121.

References

1. E. Jaspers and J. Groenenboom, "Quantification of the optimal video-coding complexity for cost-efficient storage," in *Digest of Tech. Papers of the Int. Conf. on Consumer Electronics*, Las Vegas, Nevada, pp. 123–124 (2005).
2. "CRS, computer recognition systems," http://www.vysionics.com/ (18 August 2013).
3. "Kapsch TrafficCom," http://www.kapsch.net/en/ktc/ (4 April 2012).
4. HP Autonomy, "Virage," http://www.virage.com/ (4 April 2012).
5. New Hampshire DOT, "Advanced transportation management system," 2011, http://www.nh.gov/dot/media/nr2011/nr102011tms.htm (10 April 2012).
6. D. Ang, Y. Shen, and P. Duraisamy, "Video analytics for multi-camera traffic surveillance," in *Proc. Second Int. Workshop on Computational Transportation Science*, Seattle, Washington, pp. 125–130 (2009).
7. USDOT, "Intelligent transportation systems research," http://www.fhwa.dot.gov/research/ (18 August 2013).
8. A. Ambardekar, M. Nicolescu, and G. Bebis, "Efficient vehicle tracking and classification for an automated traffic surveillance system," in *Int. Conf. on Signal and Image Processing*, Kailua-Kona, Hawaii, pp. 1–6 (2008).
9. I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag Inc., New York (2002).
10. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, New York (2006).
11. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Recognition*, Wiley-Interscience, New Jersey (2000).
12. X. Ma and W. Grimson, "Edge-based rich representation for vehicle classification," in *Int. Conf. on Computer Vision*, Beijing, China, pp. 1185–1192 (2006).
13. C. Zhang, X. Chen, and W. Stork, "A PCA-based vehicle classification framework," in *Proc. of Int. Conf. on Data Engineering Workshops*, Atlanta, Georgia (2006).
14. S. Gupte et al., "Detection and classification of vehicles," *IEEE Trans. Intell. Transp. Syst.* **3**(1), 37–47 (2002).
15. Z. Zhu et al., "VISTRAM: a realtime vision system for automatic traffic monitoring," *Image Vis. Comput.* **18**(10), 781–794 (2000).
16. P. Kumar et al., "Framework for real-time behavior interpretation from traffic video," *IEEE Trans. Intell. Transp. Syst.* **6**(1), 43–53 (2005).
17. S. Messelodi et al., "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern Anal. Appl.* **8**(1/2), 17–31 (2005).
18. B. Morris and M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Trans. Intell. Transp. Syst.* **9**(3), 425–437 (2008).
19. N. Buch, J. Orwell, and S. Velastin, "Three-dimensional extended histograms of oriented gradients (3-DHOG) for classification of road users in urban scenes," in *Proc. of British Machine Vision Conf.*, London, UK, pp. 1–11 (2009).
20. N. Buch, J. Orwell, and S. Velastin, "Urban road user detection and classification using 3-D wireframe models," *IET Comput. Vis.* **4**(2), 105–116 (2010).
21. H. Kollnig and H. Nagel, "3D pose estimation by directly matching polyhedral models to gray value gradients," *Int. J. Comput. Vis.* **23**(3), 283–302 (1997).
22. J. Lou et al., "3-D model-based vehicle tracking," *IEEE Trans. Image Process.* **14**(10), 1561–1569 (2005).
23. R. Wijnhoven and P. de With, "3D wire-frame object modeling experiments for video surveillance," in *Int. Symp. on Information Theory*, Benelux, pp. 101–106 (2006).
24. X. Song and R. Nevatia, "Detection and tracking of moving vehicles in crowded scenes," in *Proc. IEEE WMVC*, Austin, Texas (2007).
25. B. Johansson et al., "Combining shadow detection and simulation for estimation of vehicle size and position," *Pattern Recognit. Lett.* **30**(8), 751–759 (2009).
26. B. Morris and M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," in *Proc. IEEE ITSC*, Toronto, Ontario, Canada, pp. 1078–1083 (2006).
27. B. Morris and M. Trivedi, "Improved vehicle classification in long traffic video streams," in *Proc. IEEE Int. Conf. AVSS*, Sydney, Australia (2006)).
28. Z. Chunrui and M. Siyal, "A new segmentation technique for classification of moving vehicles," in *Proc. of Vehicular Technology Conf.*, Tokyo, Japan, pp. 323–326 (2000).
29. D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
30. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors,"*IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1615–1630 (2003).
31. C. Huang and W. Liao, "A vision-based vehicle identification system," in *Proc. Conf. on Pattern Recognition*, Cambridge, England, Vol. 4, pp. 364–367 (2004).
32. P. Ji, L. Jin, and X. Li, "Vision-based vehicle type classification using partial Gabor filter bank," in *Proc. of the Int. Conf. on Automation and Logistics* (2007).
33. R. Wijnhoven and P. de With, "Experiments with patch-based object classification," in *IEEE Conf. on Advanced Video and Signal Based Surveillance* , London, UK, pp. 105–110 (2007).
34. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.* **3**(1), 71–86 (1991).
35. B. Schölkopf et al., "Estimating the support of a high-dimensional distribution," *Neural Comput.* **13**(7), 1443–1471 (2001).
36. R. Fisher, "The statistical utilization of multiple measurements," *Ann. Eugen.* **8**(4), 376–386 (1938).
37. R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Madison, Wisconsin, Vol. 2, pp. 264–271 (2003).
38. K. Mikolajczyk et al., "A comparison of affine region detectors," *Int. J. Comput. Vis.* **65**(1), 43–72 (2005).
39. J. Migdal and W. Grimson, "Background subtraction using Markov thresholds," in *IEEE Workshop on Motion and Video Computing*, Breckenridge, Colorado, pp. 58–65 (2005).
40. A. Ambardekar, M. Nicolescu, and S. Dascalu, "Ground truth verification tool (GTVT) for video surveillance systems," in *Proc. of Advances in Computer Human Interactions*, Cancun, Mexico, pp. 354–359 (2009).
41. NVIDIA, "CUDA parallel programming made easy," http://www.nvidia.com/object/cuda_home_new.html (18 August 2013).
42. N. Buch, S. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.* **12**(3), 920–939 (2011).
43. A. Ottik and H. Nagel, "Initialization of model-based vehicle tracking in video sequences of inner city intersections," *Int. J. Comput. Vis.* **80**(2), 211–225 (2008).
44. W. Hu et al., "A survey of visual content-based video indexing and retrieval," *IEEE Trans. Syst., Man, Cybern.* **41**(6), 797–819 (2011).
45. L. Shapiro and G. Stockman, *Computer Vision*, Prentice-Hall Inc., Saddle River, New Jersey (2001).
46. P. Nguyen and H. Le, "A multimodal particle-filter-based motorcycle tracking system," in *Proc. of the 10th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence*, Hanoi, Vietnam, 819–828 (2008).
47. Y. Li, "On incremental and robust subspace learning," *Pattern Recognit.* **37**(7), 1509–1518 (2004).

Amol Ambardekar graduated from University of Mumbai, India (2002), with a BE degree in electronics engineering. He completed his MS in applied physics from East Carolina University (2005). He received another MS (2007) and a PhD degree (2012) both in computer science from University of Nevada, Reno (UNR). He is currently a research software development engineer at Microsoft Corporation. His research interests include object recognition and tracking, vision-based surveillance, and human–robot interaction. From 2010 to 2012, for three consecutive years, he received Outstanding International Graduate Student Award from UNR. He is a member of IEEE.

Mircea Nicolescu received a BS degree from the Polytechnic University Bucharest, Romania, in 1995, an MS degree from the University of Southern California (USC) in 1999, and a PhD degree from USC in 2003, all in computer science. He is currently an associate professor of computer science at UNR and codirector of the Computer Vision Laboratory (CVL). His research interests include visual motion analysis, perceptual organization, vision-based surveillance, and activity recognition. In 1999 and 2003, he received the USC Academic Achievements Award, and in 2002, the Best Student Paper Award at the International Conference on Pattern Recognition in Quebec City, Canada. He is a member of the IEEE Computer Society.

George Bebis received a BS degree in mathematics and an MS degree in computer science from the University of Crete, Greece, in 1987 and 1991, respectively, and a PhD degree in electrical and computer engineering from the University of Central Florida, Orlando, in 1996. Currently, he is a professor in the Department of Computer Science and Engineering at UNR, director of the UNR CVL, and visiting professor at King Saud University. His research interests include computer vision, image processing, pattern recognition, machine learning, and evolutionary computing. His research has been funded by National Science Foundation, NASA, Office of Naval Research, National Institute of Justice (NIJ), and Ford Motor Company. He is an associate editor of the *Machine Vision and Applications* journal and serves on the editorial boards of the *International Journal on Artificial Intelligence Tools* and the *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization* journal. He has served on the program committees of various national and international conferences and has organized and chaired several conference sessions. In 2002, he received the Lemelson Award for innovation and entrepreneurship.

Monica Nicolescu is an associate professor with the Computer Science and Engineering Department at UNR and is the director of the UNR Robotics Research Lab. She earned her PhD degree in computer science from USC (2003) at the Center for Robotics and Embedded Systems. She obtained her MS degree in computer science from USC (1999) and a BS in computer science at the Polytechnic University Bucharest, Romania (1995). Her research interests are in the areas of human–robot interaction, robot control, learning, and multirobot systems. Her research has been supported by the National Science Foundation, the Office of Naval Research, the Department of Energy and Nevada Nanotech Systems. In 2006, she was a recipient of the NSF Early Career Development Award (CAREER) for her work on robot learning by demonstration.