Vehicle Identification via Sparse Representation

Shuang Wang, *Student Member, IEEE,* Lijuan Cui, *Student Member, IEEE,* Dianchao Liu, Robert Huck, *Member, IEEE,* Pramode Verma, *Senior Member, IEEE,* James J. Sluss, Jr., *Member, IEEE,* and Samuel Cheng, *Member, IEEE*

***Abstract*—In this paper, we propose a system using video cameras to perform vehicle identification. We tackle this problem through reconstructing an input by using multiple linear regression models and compressed sensing, which provide new ways to deal with three crucial issues in vehicle identification: *feature extraction*, *online vehicle identification database build up*, and *robustness to occlusions and misalignment*. The results show the capability of the proposed approach.**

***Index Terms*—Vehicle identification, Sparse representation**

I. Introduction

States conduct traffic monitoring for many reasons, including highway planning and design or motor vehicle enforcement. Traffic monitoring can be classified into two different types: flow monitoring and route monitoring. Flow monitoring will observe the amount of traffic flowing through an interested check point, whereas route monitoring will identify the route of an interested vehicle. Unlike flow monitoring, route monitoring generally needs to know the identity of the observed vehicle and is generally more difficult. This route monitoring capability can provide valuable information for freight logistics analysis, forecast modeling, and future transportation infrastructure planning.

For vehicle detection using a video or image sequence, the most obvious approach has been to: first compute the stationary background image, then identify the moving vehicles as those pixels in the image that differ significantly from the background, which is named background subtraction [1]. However, traffic shadows cause serious problems when doing subtraction, and slow moving or stationary traffic is difficult to detect. This led to the emergence of the adaptive background methods [1], [2]. After background subtraction, connected regions in the foreground image, namely blobs, will be associated with different vehicles and tracked over time using different algorithms, such as cross-correlation [3], mean shift [4], etc. Moreover, learning-based systems and a hidden Markov model are proposed for on-road vehicle detection and tracking in [5] and [6], [7], respectively.

The ability of vehicle detection and tracking with video will enable us to further classify or identify interested vehicles. For the video based vehicle classification, there are many techniques concentrating on this work, such as, Support Vector Machines (SVM) [8], PCA with Neural Networks (NNs) [9], a weighted k-nearest neighbor (wkNN) [10], and BP neural network [11]. Unlike classification problems that classify different vehicles into different categories, the video based vehicle identification problem is to maintain the identity of a vehicle as it travels through multiple video camera sites. In [12], Zeng *et al.* proposed a color based vehicle matching system with the highest reported true positive rate of 16.42%. However, their experimental setup was too ideal to reflect real traffic conditions. The proposed system needs to know the average time for vehicles to travel from site 1 to site 2 to reduce the number of candidate vehicles for matching. It is very likely that one cannot find a corresponding vehicle in the candidate set, since the size of candidate set for their system is typically 8 vehicles. Moreover, Kogut *et al.* [13] combined color features and the spatial organization of vehicles within platoons to improve the identification accuracy. A maximum positive match rate of 45% was reported in their work. Nevertheless, their results were based on only 22 samples, which was too small to cover different traffic conditions. In addition, given a platoon of vehicles at site 2, it is very difficult to find the corresponding platoon from site 1, since the platoons of vehicles may change significantly, when the two sites are far from each other. In this case, this algorithm will fail, since its performance ideally depends on the spatial organization of the vehicles within their platoons. Another video-based vehicle identification system achieved impressive performance by using multiple individual vehicle features, such as color, external dimensions, points of optical demarcation, etc. [14], [15]. However, this system needs specially designed hardware for top-down camera views, where each camera also needs to be calibrated manually before performing identification. Moreover, all their results were obtained by using highly overlapped vehicle databases, where the overlap rate is about 85%. Thus the performance of low overlapped data for their system is still unknown. Nevertheless, the results obtained from previous vehicle identification research [12], [13], [14], [15] are all under some given restraints, which makes it unclear how the performance of a video based identification system would be without the aforementioned restraints. Recently, Wright *et al.* proposed a face recognition algorithm [16] using sparse representation, which offers very competitive performance for face recognition. Moreover, sparse representation is also employed for scene, object and pattern classification in [17], [18], [19]. Based on the idea of sparse representation for objection classification and identification, we propose a video based vehicle identification framework in this paper. The constructed system was designed and tested under a realistic setup in contrast of the aforementioned limitations in the previous research.

The main contributions and accomplishments of our proposed system are in the following:

1) We use video cameras to capture the critical information of vehicles for the purpose of vehicle tracking when they enter the state, and use additional video cameras to track their routes. Unlike [14], [15], our system does not need specially designed hardware or the calibrated cameras. Moreover, the cameras can be placed at the side of highway, which makes our system easier to deploy.
2) We treat the problem of vehicle identification from different video sources as a signal reconstruction out of multiple linear regression models and use rising theories from an emerging signal processing area – compressive sensing to solve this problem. By employing a Bayesian formalism to compute the l^1 minimization of the sparse weights, the proposed framework provides new ways to deal with three crucial issues in vehicle identification: *feature extraction*, *online vehicle identification database building*, and *robustness to occlusion and misalignment*. For feature extraction, we use the simple down-sampled features which offer good identification performance as long as the features space is sparse enough. The theory also provides a validation scheme to decide if a newly entering vehicle has been already included in the database. Moreover, by taking advantages of down-sample based features, one can easily

Manuscript received, Dec. 2010. This work was supported by the Oklahoma Transportation Center under the funding OTCREOS9.1-16. The Associate Editor for this paper was .

The authors are with the School of Electrical and Computer Engineering, University of Oklahoma, Tulsa, OK, USA 74135 (email:{shuangwang, lj.cui, dianchaoliu, rchuck, pverma, sluss, samuel.cheng }@ou.edu).

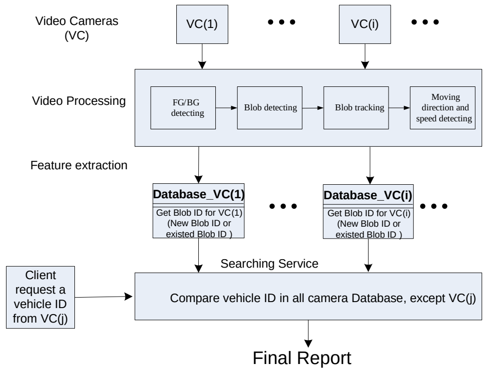

Fig. 1. The Process flow.

introduce features of newly entering vehicles into the vehicle identification database without using training algorithms, e.g. PCA [9]. Finally, Bayesian formalism provides a measure of confidence of each sparse weight.

3) Different from previous research [12], [13], [14], [15], where only about 100 vehicles were used for testing and the testing databases were highly overlapped, we conduct extensive experiments on different types of vehicles on interstate highways to verify the efficiency and accuracy of our proposed system. In our experiments, more than 1200 vehicles were used for testing and the overlap rate of the testing databases are less than 48%. The results show that the proposed framework works well on all kind of vehicles.

II. System Architecture

A. System Overview

Our vehicle identification system is able to detect, track and identify each vehicle and transmit vehicle information to a service center for further route tracking and other traffic monitoring tasks. The system includes three main components: video cameras, a service center, and clients. Video cameras are used to gather the traffic information including environment conditions, illumination conditions, and vehicle information. In addition, there are several parallel video cameras which are setup along the side of highway. These video cameras should be reliable, network accessible, of high resolution and high speed. We propose to use the Axis 223M network cameras. The service center component, the most critical part, collects the images from video cameras and employs our vehicle identification algorithm to achieve the identification results. The clients are terminals that query the identification results from the service center and produce reports of desired statistics and routing information.

B. Process Flow

The process flow for the video camera feeds used for vehicle identification is shown in Fig.1. Each video feed is sent to the service center for further processing, eg. the i-th video camera $VC(i)$ in Fig.1. At the service center, the images from the video camera are processed by the video processor module, which performs foreground/background (FG/BG) detecting, blob detecting, blob tracking, moving direction and speed detecting to extract features contributing to a unique vehicle ID. Then these vehicle IDs from different video cameras will be saved into a database with corresponding indices. When a vehicle ID, eg. the vehicle ID from $VC(j)$, is requested by the client, the given vehicle information of $VC(j)$ will be compared with other VC databases $VC(1), \dots, VC(m)$ except $VC(j)$, where m denotes the total number of VCs. If a corresponding ID is found in $VC(k), k \neq j$, it will report that this vehicle was captured in the k-th VC, otherwise, it will report -1, which means that this vehicle has not been captured by any VCs before.

III. Video Based Vehicle Identification

A. Vehicle Detecting and Tracking

Vehicle detecting and tracking is the first stage for any further identification processing. The four main components in our vehicle detecting and tracking scheme are shown as the video processing section in Fig. 1. 1) FG/BG Detecting: we adopt the approach in [2], which provides an adaptive background mixture model for real-time tracking by modeling the values of any pixel as a mixture of Gaussians. This method is robust for lighting changes, tracking through cluttered regions, slow-moving objects and so on. 2) Blob Detecting: our blob detector is implemented based on [20] to detect any newly entering object in each frame using the output from the FG/BG estimation module. 3) Blob Tracking: the Blob tracking module provides a way to track blobs from the current frame to the next frame [4]. 4) Moving Direction and Speed Detecting: it accomplished by using optical flow estimation [21], which tries to calculate the motion between two video frames at times t and $t+\tau$. In our scheme, we use the blobs with the same index in different video frames to calculate the optical flow.

The aforementioned algorithms offer high sensitivity for blob detecting and tracking, however, the false positive rate could be high due to clutter from the motion of leaves and grass. Moreover, we may only be interested in one direction of traffic flow. To tackle these issues, we utilize the following filters to exclude these unwanted blobs.

1) Blob histogram (BH) filter: excludes blobs where the number of observations from different video frames for each given blob ID is less than τ_{BH} times, where τ_{BH} is a predetermined threshold.
2) Motion distance (MDs) filter: excludes blobs whose moving distance is less than a given threshold τ_{MDs} (in pixels).
3) Motion direction (MDr) filter: excludes blobs whose motion direction are not the same as the pre-assigned direction τ_{MDr} (right, left, up, down and etc.).

B. Vehicle Identification via Sparse Representation and Bayesian Formalism

A basic problem in vehicle identification is to determine if a newly entering vehicle has already been registered in a database or not and to find a corresponding vehicle ID if such a record exist. The core idea of the proposed vehicle identification algorithm is based on sparse representation, where a similar idea was used in [16] for face recognition.

1) Sparse Representation of a Vehicle: Before generating a sparse representation for a vehicle and finding its corresponding vehicle ID, we will first arrange the database into matrices, which are built using labeled training samples from M different vehicles. Here we assume that k_i denotes the number of training images for the i-th vehicle ID, where $i = 1, \dots, M$, and $k = k_1 + k_2 + \cdots k_M$ denotes the number of images in the database. Then, we reshape each $w \times h$ image into a column vector $\nu \in \mathbb{R}^c$, where $c = wh$; the k_i training images from the i-th vehicle ID constitute the columns of a matrix $\Phi_i = [\nu_{i,1}, \nu_{i,2}, \dots, \nu_{i,k_i}] \in \mathbb{R}^{c \times k_i}$; all k images from the database are combined to form a new matrix $\Phi = [\Phi_1, \Phi_2, \dots, \Phi_M] = [\nu_{1,1}, \nu_{1,2}, \dots, \nu_{M,k_M}] \in \mathbb{R}^{c \times k}$.

For a newly entering vehicle $u \in \mathbb{R}^c$, if sufficient training samples in the database share the same feature as the incoming vehicle, (e.g. this happens when the incoming vehicle was captured previously, let say, with a vehicle ID i), then the vehicle can be approximately represented as the linear combination of the training samples in Φ_i

$$y = \Phi_i \theta = \theta_{i,1}\nu_{i,1} + \theta_{i,2}\nu_{i,2} + \cdots + \theta_{i,k_i}\nu_{i,k_i}, \tag{1}$$

where $\theta = [\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,k_i}]^T$ and $\theta_{i,j} \in \mathbb{R}, j = 1, 2, \ldots, k_i$.

However, we do not know the identity of the incoming vehicle at the beginning. Fortunately, we can instead represent the incoming vehicle $y \in \mathbb{R}^c$ using the entire set of images in the database with relatively small increase in computation complexity. The linear combination of all the training samples is written as

$$y = \Phi x_s = [\Phi_1, \Phi_2, \ldots, \Phi_M] x_s \tag{2}$$

where with a high probability, x_s=$[0, \ldots, 0, \theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,k_i}, 0, \ldots, 0]^T \in \mathbb{R}^k$ is a coefficient vector which just has nonzero entries for those associated with the i-th vehicle ID.

In order to find x_s which can accurately determine the identity of the incoming vehicle, we need to solve the linear equation $y = \Phi x$. In general, measurement data may be noisy, so y may not be represented as the sparse combination of training samples exactly. Thus Eq. (2) will be rewritten as:

$$y = \Phi x_s + \Upsilon_z, \tag{3}$$

where $\Upsilon_z \in \mathbb{R}^c$ is noise and has a bounded energy $\|\Upsilon_z\|_2 < \varepsilon$. Nevertheless, this is a underdetermined equation and it does not have a unique solution x_s. To solve the sparse solution x_s without NP-hard, it turns out to be a l^1-norm minimization problem

$$\hat{x} = arg\min \|x\|_1 \quad \text{subject to} \quad \|\Phi x - y\|_2 \leq \varepsilon. \tag{4}$$

2) Sparse Solution via Bayesian Formalism: To find the sparse solution for the l^1-norm minimization problem, numerous methods have been proposed, such as Matching Pursuit (OMP) [22], LASSO [23], Interior-point Methods [24], SAMP [25] and Gradient Method [26]. However, the above methods only provide approximate sparse solutions and do not tell how likely the given solutions are optimum. Therefore, we will use Bayesian formalism instead which returns both a sparse solution x and the probability information indicating the uncertainly of the solution from the actual sparse x. Our approach is based on [27] by extending Tipping's Relevance Vector Machine (RVM) theory [28].

First, we assume that x is the sum of two parts x_b and x_e (so, $x = x_b + x_e$), where $x_b \in \mathbb{R}^k$ is the vector composed of nonzero entries only at the L largest coefficients of x, and $x_e \in \mathbb{R}^k$ is the vector composed of nonzero entries only at the rest of the coefficients. Moreover, since we assume that measurements can be noisy as in Eq.(3), the vector corresponding to a vehicle y is rewritten as:

$$y = \Phi x + \Upsilon_z = \Phi x_b + \Phi x_e + \Upsilon_z = \Phi x_b + \Upsilon_e + \Upsilon_z = \Phi x_b + \Upsilon \tag{5}$$

where $\Upsilon_e = \Phi x_e$. Using the Central-Limit Theorem [29], we assume that both Υ_e and Υ_z are zero mean and approximately Gaussian distributed, then $\Upsilon = \Upsilon_e + \Upsilon_z$ can be approximated as a Gaussian noise with zero mean and unknown variance σ^2. Then the Gaussian likelihood is given by

$$p(y|x_b, \sigma^2) = (2\pi\sigma^2)^{-c/2} \exp(-\frac{1}{2\sigma^2}\|y - \Phi x_b\|^2). \tag{6}$$

Given Φ and y, the problem now is to estimate the sparse vector x_b and the noise variance σ^2. By Bayes' rule, we have

$$p(x_b, \sigma^2|y) = \frac{p(y|x_b, \sigma^2)p(x_b, \sigma^2)}{p(y)}. \tag{7}$$

Note that x_b is sparse and can be modeled by a Laplace distribution [30]. However, the Laplace prior is not conjugate to the Gaussian likelihood and thus the inference problem can not be written in closed-form [30]. Thus instead of Laplace prior, we will perform a hierarchical sparseness prior [28] which has similar properties as the Laplace prior and thus allows convenient conjugate exponential analysis on x_b. Then, based on the priors defined according to [28], the posterior can be decomposed as:

$$p(x_b, \alpha, \sigma^2|y) = p(x_b|y, \alpha, \sigma^2)p(\alpha, \sigma^2|y), \tag{8}$$

where α is a hyperparameter. The first term $p(x_b|y, \alpha, \sigma^2)$ finally can be expressed analytically as:

$$\begin{aligned} p(x_b|y, \alpha, \sigma^2) &= \frac{p(y|x_b, \sigma^2)p(x_b|\alpha)}{p(y|\alpha, \sigma^2)}, \\ &= (2\pi)^{-(k+1)/2}|\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(x_b - \mu)^T \Sigma^{-1}(x_b - \mu)\right\}, \end{aligned} \tag{9}$$

with the covariance and mean

$$\begin{aligned} \Sigma &= (\sigma^{-2}\Phi^T\Phi + A)^{-1}, \\ \mu &= \sigma^{-2}\Sigma\Phi^T y, \end{aligned} \tag{10}$$

respectively, where $A = diag(\alpha_0, \alpha_1, \ldots, \alpha_k)$. Maximizing the second term $p(\alpha, \sigma^2|y)$ is equal to maximizing the term $p(y|\alpha, \sigma^2)$ since $p(\alpha)$ and $p(\sigma^2)$ are uniform hyper-priors, which is given by:

$$\begin{aligned} p(y|\alpha, \sigma^2) &= \int p(y|x_b, \sigma^2)p(x_b|\alpha)dx_b, \\ &= (2\pi)^{-(k)/2}|\sigma^2 I + \Phi A^{-1}\Phi^T|^{-1/2} \\ &\quad \exp\left\{-\frac{1}{2}y^T(\sigma^2 I + \Phi A^{-1}\Phi^T)^{-1}y\right\}, \end{aligned} \tag{11}$$

where estimation of these hyperparameters α and σ^2 can be achieved by employing the Type-II maximum likelihood method which is also referred to as the "evidence for the hyper-parameters"[28].

3) Identification Based on Sparse Representation: Before identifying an incoming vehicle, first, we need to use the information of sparse representation to decide if the test object is a vehicle, and if the entering vehicle corresponds to one of the vehicle IDs in the database.

For each estimated sparse representation $\hat{x}$, the entries of coefficient vector $\hat{x}$ distribute in two different ways: a). most nonzero entries concentrate on one vehicle ID, and b). all nonzero entries spread widely among multiple vehicle IDs or the entire database. The first case implies that the incoming vehicle is likely to correspond to the vehicle ID on which non-zero entries concentrate; whereas the second case indicates that the feature information of the incoming vehicle is not in the database. Based on the above observation, we adopt the sparsity concentration index (SCI) to determine if a vehicle has been captured before [16]. The SCI of a coefficient vector $x \in \mathbb{R}^k$ is defined as

$$SCI(x) = \frac{M \cdot \max_i \|\delta_i(x)\|_1 / \|x\|_1 - 1}{M - 1} \quad \in [0, 1], \tag{12}$$

where $\delta_i(x) \in \mathbb{R}^k$ is a vector whose coefficients are only associated with the i-th vehicle ID of vector x.

Hence, for an estimated sparse representation $\hat{x}$ solved in Section III-B1, if $SCI(\hat{x}) = 1$, nonzero entries only concentrate on one vehicle ID, and if $SCI(\hat{x}) = 0$, nonzero entries are spread uniformly among all vehicle IDs. Given a threshold $\epsilon \in (0, 1)$, if $SCI(\hat{x}) \geq \epsilon$, the test vehicle will be considered as a "known" vehicle, otherwise an "unknown" vehicle will be reported. For the former case, we still need to determine the identity of the vehicle. We can achieve this by comparing the residual errors corresponding to different vehicle IDs. More precisely, denote $\hat{y}_i = \Phi\delta_i(\hat{x}_1)$ which is the approximate representation obtained by using only the entries associated with the i-th vehicle ID. Intuitively, we assign the incoming vehicle y to the ID with the best approximation. This corresponds to the minimum residual error between y and $\hat{y}_i$ given by

$$\min_{i=1,\cdots,M} r_i(y) = \min_{i=1,\cdots,M} \|y - \Phi\delta_i(\hat{x}_1)\|_2 \tag{13}$$

Now based on the sparse representation, SCI method for validation, and residual identification, the algorithm procedure for vehicle identification can be summarized in the following:

1) Input: an arranged matrix from a database with M vehicles $\Phi = [\Phi_1, \Phi_2, \ldots, \Phi_M] \in \mathbb{R}^{c \times k}$, an incoming vehicle represented by $y \in \mathbb{R}^c$, error tolerance $\varepsilon > 0$ and SCI threshold $\epsilon \in (0, 1)$.
2) Normalize the columns of Φ to have unit l^2-norm.
3) Solve the l^1-minimization problem: $\hat{x} = argmin_x \|x\|_1 \quad \text{subject to} \quad \|\Phi x - y\|_2 \leq \varepsilon$
4) Compute $SCI(\hat{x}) = \frac{M \cdot \max_i \|\delta_i(x)\|_1 / \|x\|_1 - 1}{M-1} \in [0, 1]$. If $SCI(\hat{x}) \geq \epsilon$, go to step 5, otherwise return a report that the incoming vehicle is not in the database.
5) Compute the residuals errors $r_i(y) = \|y - \Phi\delta_i(\hat{x})\|_2$, for $i = 1, \ldots, M$.
6) Output : id(y)= $argmin_{i=1,\cdots,M} r_i(y)$.

C. Identification Based on Multiple Frames

As stated in Section III-B3, for further identification, $SCI(\hat{x})$ works like a filter for each frame to sift through any unknown vehicle. However, in the vehicle identification problem, sometimes the constraint $SCI(\hat{x}) \geq \epsilon$ can not differentiate between the a known and an unknown vehicle accurately. Fortunately, it is possible to take advantage of the information from multiple frames to further improve the identification accuracy. In this paper, we propose an additional rule using the identification concentration index (ICI), which is based on multiple frame validation, to improve the vehicle identification accuracy.

Here we assume that there are F number of frames which include an incoming vehicle and $d \in \mathbb{R}^F$ is a vector to save the identified IDs from F frames. An average $\overline{SCI(\hat{x})}$ is obtained by $\overline{SCI(\hat{x})} = \frac{1}{F}\sum_{l=1}^{F} SCI(\hat{x}_l)$ and assuming that out of the F frames, D number of unique identified IDs are found. Then, considering the information from all the F frames, the proposed ICI is defined as:

$$ICI(d) = \frac{D \cdot \max(\rho_j(d))/\|d\|_0 - 1}{D - 1} \quad \in [0, 1], \tag{14}$$

where $\rho_j(d)$ counts the existing number of j-th unique IDs in d, $j = 1, \ldots, D$; if $D = 1$, $ICI(d) = 1$, since there is only one vehicle ID. Similarly, if $ICI(d) = 1$, F frames identify one vehicle ID, and if $ICI(d) = 0$, F frames are spread among all D number of IDs. Then given a threshold $\zeta \in (0, 1)$, an incoming vehicle is considered as "known "if $ICI(d) \geq \zeta$, otherwise it is considered as "unknown ".

Now we will introduce a method to combine SCI and ICI to increase the accuracy of vehicle identification. Based on the previous algorithm for vehicle identification and the proposed ICI, the algorithm procedure can be rewritten as follows:

1) Input: an arranged matrix from a database with M vehicles $\Phi = [\Phi_1, \Phi_2, \ldots, \Phi_M] \in \mathbb{R}^{c \times k}$, an incoming vehicle $y \in \mathbb{R}^c$, error tolerance $\varepsilon > 0$, SCI threshold $\epsilon \in (0, 1)$ and ICI threshold $\zeta \in (0, 1)$, where the parameters ϵ and ζ can be tuned by using an SVM classifier.
2) Normalize the columns of Φ to have unit l^2-norm.
3) For all F frames which include an incoming vehicle, solve the l^1-minimization problem: $\hat{x_l} = argmin_{x_l} \|x_l\|_1 \quad \text{subject to} \quad \|\Phi x_l - y_l\|_2 \leq \varepsilon,\ l = 1, \ldots, F$.
4) Compute $SCI(\hat{x}_l) = \frac{M \cdot \max_i \|\delta_i(x_l)\|_1 / \|x_l\|_1 - 1}{M-1} \in [0, 1]$ for each frame and then get $\overline{SCI(\hat{x})} = \frac{1}{F}\sum_{l=1}^{F} SCI(\hat{x}_l)$ for F multiple frames.
5) Compute $ICI(d) = \frac{D \cdot \max(\rho_j(d))/\|d\|_0 - 1}{D-1} \in [0, 1]$.
6) If $\overline{SCI(\hat{x})} < \epsilon$, and $ICI(d) < \zeta$, where $\epsilon \in (0, 1)$ and $\zeta \in (0, 1)$, return a report that the incoming vehicle is not in the database, otherwise go to step 7. Note that the value of ϵ and ζ are tuned by using an SVM classifier.
7) Compute the residual errors $r_i(y) = \|y - \Phi\delta_i(\hat{x})\|_2$, for $i = 1, \ldots, M$.
8) Output : id (y)= $argmin_{i=1,\cdots,M} r_i(y)$.

IV. Results

A. Experiment Setup

First, we separate the video images captured from two video cameras into two parts, one for training and the other for testing. During the training phase, we extracted a 10 minute length of video from video camera $VC(1)$ to build the vehicle database, which included 291 captured vehicles and a total of 13,931 images (about 48 frames per vehicle). Then another 20 minute length of video with 601 captured vehicles (about 23,920 images) were extracted from video camera $VC(2)$, where 291 vehicles were also captured by $VC(1)$ and the remaining 310 vehicles were not, and were registered into the database using our proposed algorithm. Then the SCI and ICI values were obtained after the registration and a ground truth was generated manually. This information is used to train the SVM classifier, which will be used to identify unknown vehicles based on the SCI and ICI values. During the testing phase, we used another 10 minute length of video extracted from $VC(1)$ to build the vehicle database, which includes 13,573 images for 287 vehicles. Then, another 23,838 images of 608 vehicles from $VC(2)$ were used to test the identification accuracy of our proposed algorithm. In this testing phase, 287 vehicles appeared in the database and 321 vehicles were not in the database [1].

Fig. 2. Blob detection result

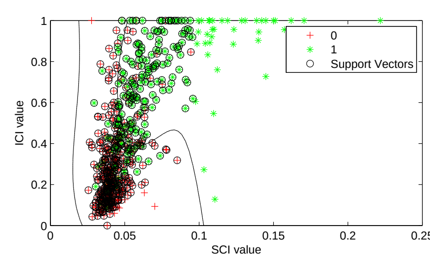

Fig. 3. Trained SVM classifier using SCI and ICI, where 1 indicates correctly identified vehicles and 0 indicates incorrectly identified vehicles

B. Vehicle Tracking and Detecting

In Fig. 2 , we present the blob detection results by using our proposed vehicle detecting and tracking algorithms. The green box

[1] A scalable solution can be achieved by restricting the number of frames per unique vehicle in building the database. That is, when a vehicle is detected as a known vehicle in the database, the newly captured frames of the vehicle might not need to be added into the database. To increase robustness of the algorithm, we could also discard some previously captured frames and replaced by the newly captured frame. Please note that the overhead to build a database as described above is very small. The complexity is dominated by the vehicle identification step. Then the central service will output a response for any input query. The selection of appropriate frames for database building depends on the real situation. Moreover, we can easily split the above-mentioned tasks into multiple pieces and process them in parallel, which can dramatically accelerate the identification process.

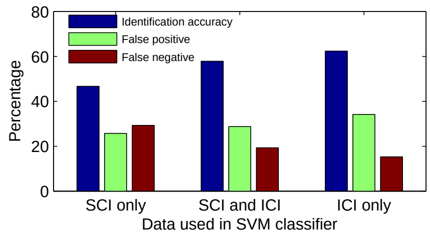

Fig. 4. Classification performance by using different kinds of data for the SVM classifier, such as SCI only, ICI only and the combination of SCI and ICI

in the video frame indicates the location and the size of a moving vehicle. Then each detected blob will be saved into the database for registration . In this experiment, we achieved greater than 90% accuracy for the blob detection. Moreover, we only focus on vehicles that travel from left to right. In Fig. 2, we can see that vehicles not moving to the right are filtered out by the MDr filter.

C. Video Based Vehicle Identification

In this section, results of the SVM classifier, identification with different feature sizes, and a detailed example of vehicle identification are presented.

1) SVM Classifier: In our experiment, we applied a sparse representation-based Identification (SRI) algorithm to each test vehicle image by solving the optimization problem in Eq. (4) with the RVM. Two dimensional training data is used in the SVM classifier which includes the SCI and ICI. Moreover, a 4th order polynomial kernel function is used in the SVM classifier. Fig. 3 shows the trained SVM classifier using the manually labeled data (SCI and ICI) obtained from the training part, where the (green) star means the vehicles appeared in both video cameras $VC(1)$ and $VC(2)$, the (red) cross means that vehicles appeared in video camera $VC(2)$ only, and the solid (black) line is the classification boundary obtained from SVM classifier.

TABLE I
IDENTIFICATION ACCURACY USING SCI AND ICI WITH A FEATURE SIZE OF 500

SP + SVM classifier output with SCI and ICI			
# of acceptance (# of vehicles in database)		# of rejection (# of vehicles out of database)	
248 (287)		360 (321)	
# of positive	# of false positive	# of negative	# of false negative
166	82	298	62
Identification accuracy	False positive rate	False negative rate	
57.84%	28.57%	19.31%	

We compare the classification performance by using different kinds of data for SVM classifier, such as, SCI only, ICI only and the combination of SCI and ICI. Before showing the results, we will define some terminologies. In this paper, “# of acceptance” is the number of vehicles that were accepted by the SVM classifier and “# of rejection” is the number of vehicles that were rejected by the SVM classifier. The identification accuracy (IA) is defined as $\frac{\text{\# of positive reports}}{\text{\# of vehicles in the database}}$, where “# of vehicles in the database” means vehicles that appeared in both video camera $VC(1)$ and $VC(2)$. The false positive rate (FPR) is defined as $\frac{\text{\# of false positive reports}}{\text{\# of vehicles in the database}}$. The false negative rate (FNR) is defined as $\frac{\text{\# of false negative reports}}{\text{\# of vehicles out of the database}}$, where “# of vehicles out of the database” means vehicles appeared in video camera $VC(2)$ only (see Table I as an example). Fig. 4 shows IA, FPR and FNR by using different kinds of data in the SVM classifier. We can see that using ICI only in the SVM obtains the best IA and lowest FNR. However, the FPR of using ICI only is the highest. Among SCI, ICI and the combination of SCI and ICI, SCI yields the worst performance in terms of lowest IA and highest FNR. Moreover, we can see that using the combination of SCI and ICI in the SVM classifier can best leverage the performance of IA, FPR and FNR.

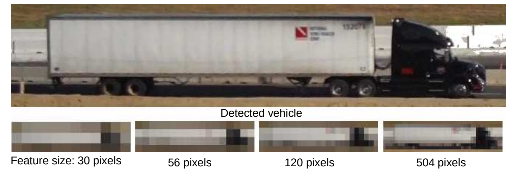

Fig. 5. The example of different feature sizes

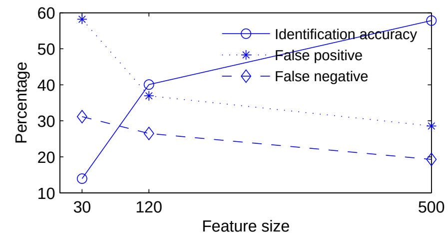

Fig. 6. Identification accuracy, false positive and false negative rate with different feature sizes.

2) Identification with Different Feature Sizes: In this section, we compare IA, FPR and FNR with different feature sizes (pixels) of 30, 120 and 504. Here, we implement a down-sampling scheme for each detected vehicle to get the feature image. The advantage of using a down-sampled feature image is that each feature image can be generated independently and requires less computation. Fig. 5 shows an example of the different feature sizes. Furthermore, in Fig. 6, we can see that IA increases as the feature size increases, while the FPR and FNR decrease as the feature size increases. Thus, to obtain a higher IA, we need a large feature image so that enough information of a given vehicle can be provided in the given feature image.

Fig. 7. An example of correct identification I

Fig. 8. An example of correct identification II

3) A Detailed Example of Identification: In this section, a detailed identification example using a sparse representation-based algorithm is presented. Fig. 7 and Fig. 8 show two correct identification results, where the database was built by using the data from video camera $VC(1)$ and the data of the querying vehicle was obtained from video camera $VC(2)$. In Fig. 7, although an occlusion (a pole) exists in the image from the database, the proposed algorithm can obtain a correct identification result. Moreover, Fig. 8 shows the case of a correct identification using misaligned images from video camera $VC(1)$ (database) and video camera $VC(2)$ (querying data). The above two cases demonstrate that the proposed algorithm is more robust with respect to occlusion and misalignment.

Fig. 9 shows an incorrect identification result, where the two vehicles from video camera $VC(1)$ and video camera $VC(2)$ are too similar to be discriminated by the proposed algorithm. However, almost any video based identification algorithm suffers from this difficulty. Thus, in this case, some additional information, such as the license plate gained from an Automatic License Plate Recognition system or Blue Tooth Traffic Detector system, could be used to increase the identification accuracy.

Vehicle from camera 1 Vehicle from camera 2

Fig. 9. An example of incorrect identification

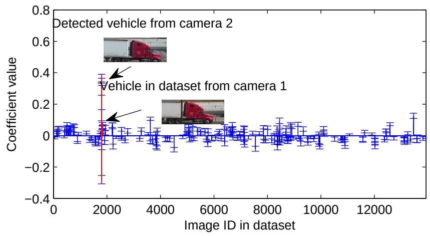

Fig. 10. Sparse coefficients for a testing vehicle

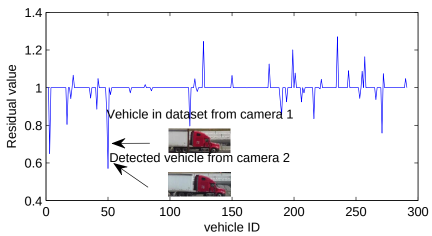

Fig. 11. Residuals for a testing vehicle

Fig. 10 and Fig. 11 show the sparse coefficients and residuals for a given test vehicle, respectively. In Fig. 10, we can see that the magnitudes of some non-zero coefficients are much larger than others. Moreover, the corresponding image IDs of these large non-zero coefficients belong to the same vehicle in the database, which is identical to the test vehicle. The error bar, another output of RVM, is also shown in Fig. 10, which can be used to measure the confidence of each coefficient. In Fig. 11, we can see that the residual between a test vehicle and a vehicle in database reach the minimum value, when the test vehicle and the vehicle in the database are identical.

D. Discussion

In this paper, we conducted extensive experiments to test the proposed vehicle identification system. Our proposed system can achieve about 57.84% identification accuracy, which is much better than the previous results of 16.42% and 45% reported by [12] and [13], respectively. Although, the identification accuracy found in our experiment was lower than [15], our system did not require specially designed hardware and top-down camera views as [15], which makes our system easier for deployment. Moreover, vehicle databases used in our experiment have a lower overlap rate of 48% and more than 1200 vehicles, which was 10 times larger than the number of the previous research [12], [13], [14], [15]. Thus, our experimental results may reflect a more realistic traffic condition, e.g. a higher probability that two vehicles are not identical but have similar or the same color, shape, etc (see Fig. 9). In our experiment, we found that most false positive and false negative reports were caused by two vehicles which are too similar to be discriminated between.

References

[1] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," *In Proc. of the Thirteenth Conference on Uncertainty in Artificial Intelli-gence(UAI)*, 1997.

[2] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. Citeseer, 1999, pp. 246–252.

[3] K. Karmann and A. von Brandt, "Moving object recognition using an adaptive background memory," *Time-varying image processing and moving object recognition*, vol. 2, pp. 289–296, 1990.

[4] D. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2. Citeseer, 2000, pp. 142–149.

[5] S. Sivaraman and M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. Intell. Transport. Syst.*, vol. 11, no. 2, pp. 267–276, 2010.

[6] A. Jazayeri, H. Cai, J. Zheng, and M. Tuceryan, "Motion based vehicle identification in car video," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 493–499.

[7] ——, "Vehicle detection and tracking in car video based on motion model," *IEEE Trans. Intell. Transport. Syst.*, no. 99, pp. 1–13.

[8] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using gabor filters and support vector machines," in *International Conference on Digital Signal Processing*, vol. 2. Citeseer, 2002, pp. 1019–1022.

[9] N. Matthews, P. An, D. Charnley, and C. Harris, "Vehicle detection and recognition in greyscale imagery," *Control Engineering Practice*, vol. 4, no. 4, pp. 473–479, 1996.

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

[11] Z. Qin, "Method of vehicle classification based on video," in *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, 2008, pp. 162–164.

[12] N. Zeng and J. Crisman, "Vehicle matching using color," in *IEEE Conference on Intelligent Transportation System*, 1997, pp. 206–211.

[13] G. Kogut and M. Trivedi, "Maintaining the identity of multiple vehicles as they travel through a video network," in *IEEE Workshop on Multi-Object Tracking*, 2001, pp. 29–34.

[14] A. MacCarley, "Video-Based Vehicle Signature Analysis and Tracking Phase 1: Verification of Concept Preliminary Testing," *UCB-ITS-PWP, PATH, University of California, Berkeley, CA*, 1998.

[15] C. MacCarley, "Video-based vehicle signature analysis and tracking system phase 2: algorithm development and preliminary testing," *UCB-ITS-PWP, PATH, University of California, Berkeley, CA*, 2001.

[16] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 210–227, 2008.

[17] L. Li, H. Su, E. Xing, and L. Fei-Fei, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," *Neural Information Processing Systems (NIPS), Vancouver, Canada*, 2010.

[18] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 99, pp. 1–1.

[19] J. Thiagarajan, K. Ramamurthy, and A. Spanias, "Sparse representations for pattern classification using learned dictionaries," in *Proceedings of Twentyeighth SGAI International Conference on Artificial Intelligence*. Springer, 2008.

[20] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti, and R. Bolle, "Appearance models for occlusion handling," *Image and Vision Computing*, vol. 24, no. 11, pp. 1233–1243, 2006.

[21] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 490–498, 1989.

[22] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[23] P. Zhao and B. Yu, "On model selection consistency of Lasso," *The Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[24] F. Potra and S. Wright, "Interior-point methods," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 281–302, 2000.

[25] T. Do, L. Gan, N. Nguyen, and T. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*. Citeseer, 2008, pp. 581–587.

[26] J. Tropp and S. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[27] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[28] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Machine learning research*, pp. 211–244, 2001.

[29] A. Papoulis, S. Pillai, and S. Unnikrishna, *Probability, random variables, and stochastic processes*. McGraw-Hill New York, 2002.

[30] J. Bernado and A. Smith, "Bayesian theory," 1994.