Vehicle Classification from Traffic Surveillance Videos at a Finer Granularity

Xin Chen and Chengcui Zhang

Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL 35294, USA
{chenxin, zhang}@cis.uab.edu

Abstract. This paper explores the computer vision based vehicle classification problem at a fine granularity. A framework is presented which incorporates various aspects of an Intelligent Transportation System towards vehicle classification. Given a traffic video sequence, the proposed framework first segments individual vehicles. Then vehicle segments are processed so that all vehicles are along the same direction and measured at the same scale. A filtering algorithm is applied to smooth the vehicle segment image. After these three steps of preprocessing, an ICA based algorithms is implemented to identify the features of each vehicle type. One-class SVM is used to categorize each vehicle into a certain class. Experimental results show the effectiveness of the framework.

Keywords: ICA, vehicle classification, Intelligent Transportation Systems.

1 Introduction

Due to its great practical importance, Intelligent Transportation Systems has been an active research area for years. Vehicle classification is one of the key tasks in an Intelligent Transportation System. Typically, acoustic or seismic sensors are used for such a purpose [1][8][13][14]. However, for road traffic analysis, the most available sources are traffic surveillance videos taken by fixed cameras. Since only the visual information can be reliably extracted and verified for such videos, computer vision based methods from the area of multimedia are required for video content analysis.

In order to identify vehicles, video object tracking needs to be performed before we can analyze each individual vehicle. There are a large amount of literatures on vehicle tracking based incident detection for traffic surveillance system. However, there has been relatively little work done in the field of vehicle classification. This is because it is an inherently hard problem. Some vehicle detection and tracking works even depend on classification techniques. [16] proposes a vehicle detection method with one of its step being classification i.e. a two class classification of vehicles and non-vehicles. A method called “Adaboost” is used for such a purpose.

T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4351, Part I, pp. 772 – 781, 2007.

In [15], an object tracking and classification method is proposed. Three categories of objects are differentiated – human, automobiles and background. For classification of human and automobiles, a concept called “dispersedness” is used based on the priori that human has smaller yet more complex shape than that of a vehicle. This is among one of the earliest works that address object classification from video. Most of the current work is purely dimension based (such as height and length of a vehicle) or shape based. In [7], a parameterized model is proposed to describe vehicles, in which vertices and topological structure are taken as the key features. One requirement of this method is that the image quality has to be sufficiently good to have the topological structures of vehicles exposed. However, this cannot be always satisfied in a real traffic surveillance system. Gupte et al. [3] propose a system for vehicle detection and classification. The tracked vehicles are classified into two categories: cars and non-cars. The classification is based on dimensions and is implemented at a coarse granularity. Its basic idea is to compute the length and the height of a vehicle, according to which a vehicle is classified as a car or a non-car. In order to classify vehicles at a finer granularity, we need a more sophisticated method that can detect the invariable characteristics for each vehicle type. In [6], the virtual loop assignment and direction-based estimation methods are used to identify vehicle types. Each vehicle type is represented by a 1-D signature chart. In their experiment, vehicles are classified into four categories: 7-seat van, fire engine, sedan and motor cycle. With this method, as mentioned in the paper, only a rough estimation of vehicle types based on vehicle length is possible. It cannot distinguish vehicles whose lengths are in approximately the same range, e.g. truck and bus. Another problem of this method is that only those vehicles traversing across virtual loops along the road direction can be detected. Therefore, we still need to further explore a method that can unveil the real, invariant characteristics of a type of vehicle. In this paper, we design an algorithm for vehicle classification at a finer granularity.

Principal Component Analysis (PCA) is a well-known algorithm in the field of object recognition. It used in the computer vision problem of human face recognition. The similarity between face detection and vehicle detection is that both analyze a 2-D image and try to find out the feature of the image content.

Independent Component Analysis (ICA) is another subspace method that has been applied to face recognition. Many works compare between ICA and PCA and show the advantages of ICA [2][4][5]. In [2], the authors applied both methods in analyzing the Coil-20 database. In [4], the authors demonstrate that ICA outperforms PCA for object recognition under varying illumination. [5] compares the effectiveness of both methods in object recognition. Since the traffic videos are taken during different time periods of the day, it is preferably that the algorithm is robust to varying illumination conditions. In this paper, we propose an ICA based vehicle classification platform.

By analyzing vehicle images with ICA, a set of features are extracted for each vehicle. These features represent the innate characteristics of the vehicle type and are fed into the classification module -- One-Class Support Vector Machine [9]. The representative features of vehicles in each vehicle type are used as training data. We build one classifier for each vehicle type which distinguishes that vehicle type from the others. In the testing phase, each set of test vehicles is tested against the classifier of each vehicle type. A test vehicle is then classified into one of the vehicle types according to the highest score it receives from each classifier.

In our experiments, we use grayscale traffic surveillance videos. It is desired that the classification is robust to the varying intensities of vehicles. For example, black and white passenger cars are expected to be classified into the same class. However, their different intensities may affect the classification result. Therefore, a filter in the preprocessing step is necessary to alleviate such problems. In this paper, a texture analysis tool is used for this purpose.

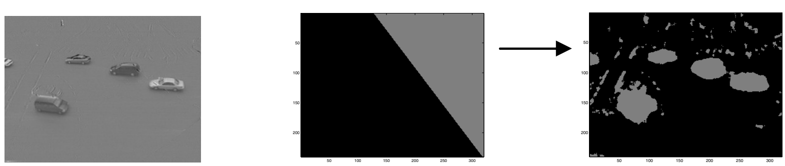
We propose an integrated system that can automatically track and categorize vehicles within a traffic surveillance video sequence. The system first tracks and segments vehicles from raw surveillance videos. Then the tracked vehicles and their features are normalized. In the final step vehicles are classified, which can provide more detailed and useful information to traffic administration. The vehicle tracking and normalization phases are based on Zhang et al.'s work [11]. In this study, improvement in the classification result by using ICA and one-class SVM is demonstrated by the experimental results at the end of this paper.

The detailed design and implementations are illustrated in the following order: Section 2 briefly introduces preprocessing module -- vehicle segmentation, adjustment and filtering. Section 3 discusses the technical details of the algorithm. Section 4 presents the system overview and the experimental results. Section 5 concludes the paper.

2 Preprocessing

2.1 Vehicle Tracking and Segmentation

For vehicle tracking and segmentation, an unsupervised video segmentation method called the Simultaneous Partition and Class Parameter Estimation (SPCPE) algorithm, coupled with a background learning algorithm, is applied to identify the vehicle objects in video sequences [10]. Figure 1 shows an example of vehicle segmentation from the initial random partition (Figure 1(a)) to the final segmentation result (Figure 1(c)).



(a) (b) (c)

Fig. 1. An example of vehicle segmentation. (a) Original frame with background removed; (b) Initial random partition; (c) Final segmentation result.

The algorithm in [10] also has the ability to track moving vehicle objects (segments) within successive video frames. By distinguishing the static objects from mobile objects in the frame, tracking information can be used to determine the trails of vehicle objects.

2.2 Vehicle Image Adjustment and Filtering

For normalization purposes, a transformation model is needed to rotate the subject cars to the same orientation and scale them to the same level. For vehicles driving toward the same direction, their rotation angles are the same. The scaling factor is determined by the shooting distance between the camera and the vehicle object. Once the rotation angle θ and the scaling factor s are available, the transformation model can be built. To preserve the co-linearity (i.e., all points lying on a line initially should still lie on a line after transformation) and the ratios of distances within the image, we use the affine transformation as our transformation function to rotate and scale vehicle objects to comparable conditions. The affine transformation is defined as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \quad (1)$$

where θ is the rotation angle and s is the scaling factor. After applying the affine transformation, we make all subject vehicles in consistent orientation and at the same scale level. This module is implemented based on Zhang et al.'s work in [11].

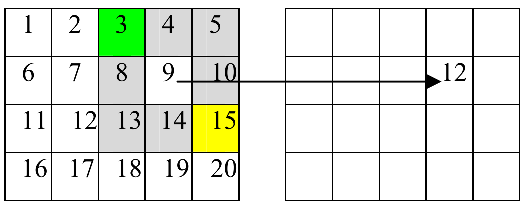


Fig. 2. Local range filtering

Although the vehicle images are transformed to grayscale images, there is still a difference with respect to intensities between bright colored (e.g. white) and dark colored (e.g. red or black) vehicles. The change of lighting conditions during the day can also cause the variations in image intensities. As mentioned in Section 1, ICA is comparatively robust in dealing with varying illuminations. Furthermore, in order to alleviate the effect of varying intensities, a filtering technique is used in the proposed framework. It calculates the local range of an image and tries to smooth out pixels within the same neighborhood. Suppose we use a 3 by 3 neighborhood window. The above figure shows the mechanism of this texture based filter.

The shaded area is the neighborhood of the pixel whose intensity value is 9. After filtering, its intensity in the corresponding position is 12 which is the difference of the maximum intensity (15) and the minimum intensity (3) of its neighborhood pixels.

Figure 3(a) is an example of a vehicle image. Figure 3(b) is the filtered image. After filtering, the outline of the vehicle is evident. In a neighborhood area, if the intensity difference is small, the whole area is smoothed to a dark patch. Otherwise, the area is highlighted such as the skeletons of vehicles. Therefore, the original color of the vehicle will not matter that much (as before); only its outline information is kept. Thus, the influence of the vehicle's original color is alleviated.

(a) Original Vehicle Image (b) Filtered Vehicle Image

Fig. 3. An example of filtered image

3 Classification

3.1 Obtain Vehicle Samples

After vehicle segmentation, the bounding boxes of vehicle segments are extracted. One factor we need to take into consideration is that the sizes of bounding boxes are different due to different vehicle sizes. This factor can affect the result of the next step – Independent Component Analysis. Therefore, we set a uniform bounding box whose size is the biggest bounding box among all samples. For those whose bounding boxes are smaller, we pad them with the mean values of their background pixels surrounding the vehicle segments. In this way, we obtain a set of training samples for each type.

Each vehicle sample is actually a 2-D image $x_i \in \Re^{m \times n}$. It can be represented as an m by n vector with m being the image height and n being the image width. We then read x_i in column-wise order, one pixel at a time, and restructure it as $x_i^{'} \in \Re^{1 \times mn}$. With k being the number of samples in the training set, we can have a matrix of k columns $X' = [x_1^{'}, x_2^{'}, ..., x_k^{'}]$. The length of each column is mn. The mean vector ω is calculated as follows:

$$\omega = \frac{1}{k}\sum_{i=1}^{k} x_i^{'} \tag{2}$$

Since ω is also a $1 \times mn$ vector, we can restore it into an m by n matrix and output it as an image. The mean “passenger car” constructed this way is shown in Figure 4. By deducting the mean vector from each vehicle image vector $x_i^{'}$, X' becomes a zero mean matrix, which is the random dataset we will analyze later.

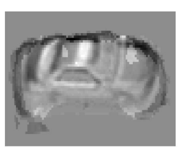


Fig. 4. The mean image of passenger car samples

3.2 Independent Component Analysis

The Independent Component Analysis (ICA) is a statistical method for revealing the underlying factors of a set of data, which are mutually independent. ICA views data as a linear mixture of sources i.e. independent components. There is little knowledge of the sources and how they are mixed. The only information we have is the observed random dataset. In order to separate the independent sources, ICA seeks an un-mixing matrix for linearly transforming to coordinates in which data are maximally statistically independent. ICA is often compared with a well known method – Principle Component Analysis (PCA) which is used to find the orthogonal bases of dataset. With PCA, data are decorrelated by being projected onto these bases. Although both ICA and PCA explore subspaces to decorrlate data, the purpose of ICA is theoretically loftier than that of PCA since ICA tries to find an un-mixing matrix such that sources are not only decorrelated but also statistically independent. Some research results have shown the advantage of ICA over PCA [2][4][5].

In ICA model, the random dataset is denoted as:

$$X' = AS \tag{3}$$

where X' contains k observed data points $[x_1, x_2, \ldots, x_k]$. In our case, x_i is a vehicle image represented by a vector. k is the number of training samples in the training set. A is the mixing matrix and S is the matrix containing the independent components that are mixed by A to represent the observed dataset X'. All we observe is the random dataset X'. A and S must be estimated according to X'. In our experiment, a fixed point version of this algorithm – FastICA [12] is used. Our assumption is that the independent components have nongaussian distributions. After estimating A, its inverse W can be computed and the independent components S is obtained by the following equation:

$$S = WX' \tag{4}$$

The length of each independent component is mn. Similarly to how we construct the mean image, we can reconstruct this vector into a 2-D image. For vehicle classification, the independent components in S are used as the bases for a low-dimensional representation. For each sample vehicle image in the training set, the following equation is used to compute its weight vector consisting of the weight of each independent component in representing that vehicle image.

$$\beta = S^T X' \tag{5}$$

The rows of β are weight vectors of vehicle images in the training set. These weight vectors are normalized to the scale of [0, 1] to avoid bias.

3.3 One-Class Support Vector Machine

One-Class classification is a kind of supervised learning mechanism. It tries to assess whether a test point is likely to belong to the distribution underlying the training data. In our case, a training set is composed of a set of vehicles of the same type. One-Class SVM has so far been studied in the context of SVMs. The objective is to create a binary-valued function that is positive in those regions of input space where the data predominantly lies and negative elsewhere.

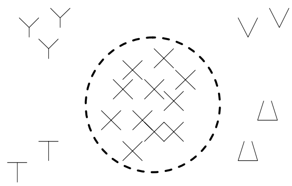


Fig. 5. One-Class classification

The idea is to model the dense region as a “ball”. Vehicles that belong to the class are inside the “ball” and the others are outside. This is shown in Figure 5 with the crosses representing the data that belongs to the positive class. If the origin of the “ball” is $\vec{\alpha}$ and the radius is r, a point $\vec{p}_i$ is inside the “ball” iff $\left\| \vec{p_i} - \vec{\alpha} \right\| \leq r$. In our case, a point is the weight vector that represents the features of a vehicle. This “ball” is actually a hyper-sphere. The goal is to keep this hyper-sphere as “pure” as possible and include as many vehicles that belong to this class as possible. Details can be found in Schölkopf’s One-Class SVM [11].

The process of classifying a new (unknown) vehicle x_{new} to one of the classes (known vehicles) proceeds in three steps:

1. Train a set of One-class SVM classifiers with the weight vectors of the sample images in the training sets. A classifier is generated for each vehicle type.
2. Reshape x_{new} into x'_{new} and obtain $\sigma_{new} = x'_{new} - \omega$. Transform σ_{new} with the independent components of the training set and obtain the feature vector β_{new} (weight vector) by Equation 5.

Test β_{new} against each classifier generated in the first step and obtain a set of scores which indicates the possibility of x_{new} belonging to each vehicle type. Finally, x_{new} will be classified into the vehicle type from which it receives the highest score.

In our experiment, there are three training sets, one for each type of vehicles: passenger car, pick-up and van. Each type of vehicles is represented by a set of weight vectors and trained by One-class SVM. Then we use the trained One-class SVM classifiers to classify new vehicles.

4 Experimental Results

From vehicle tracking and segmentation to vehicle classification, we now have an integrated system that can automatically track and classify vehicles in traffic surveillance videos. A real-life traffic video sequence with 67635 frames is used to analyze the performance of the proposed vehicle classification algorithm. The video sequence is obtained from a high way survelliance camera.

By vehicle segmentation and tracking, all distinct vehicle segments are extracted and form a sample pool. By "distinct", we mean each vehicle segment in the sample pool corresponds to a real distinct vehicle in reality. For repetitive appearances of a vehicle object across multiple frames, only one instance (segment) of that vehicle is chosen for training or testing. The preprocessing step is time consuming and is performed offline. ICA Analysis step requires some mannul work i.e. selecting the training samples and therefore is also executed offline. The classification step can work in real time.

In our experiment, three sets of training samples are formed for three categories of vehicles. They are "passenger cars (PC)", "pickup trucks (PK)" and "vans and SUVs (VAN)". In each training set, there are 50 vehicles. It is worth mentioning that, the system can be easily extended to detect more categories of vehicles. The only modification for this is to gather samples for each category of vehicles.

Table 1. The Test Result with ICA

ICA-SVM		**Test 1**	**Test 2**	**Test 3**
PC	***Recall***	74%	66%	64%
	Precision	84.60%	82%	78%
PK	***Recall***	68%	56%	70%
	Precision	72.3%	72%	83.3%
VAN	***Recall***	64%	58%	74%
	Precision	74.70%	68%	79%

We have three sets of test samples with each containing 150 vehicles randomly chosen from the sample pool. Table 1 shows the precision and recall values of the proposed ICA-based algorithm and the test result of using the PCA-based algorithm is presented in Table 2.

Table 2. The Test Result with PCA

PCA-SVM		**Test 1**	**Test 2**	**Test 3**
PC	***Recall***	40%	66%	54%
	Precision	57.3%	71.3%	62.7%
PK	***Recall***	54%	56%	52%
	Precision	62%	62.7%	57.3%
VAN	***Recall***	64%	54%	66%
	Precision	73.3%	64%	69.3%

From the above two tables we can see that ICA performs better than PCA. It is worth mentioning that the precision of ICA-based algorithm is much higher than that of PCA. This is because ICA can better identify negative samples in the testing data set. The system proposed in this paper incorporates video segmentation, vehicle tracking, and vehicle classification into one single integrated process. Especially, the classification is designed to find the invariant features of vehicles so as to categorize them at a fine granularity.

5 Conclusion

In this paper, a vehicle classification framework is proposed which incorporates several stages of work. First, traffic video sequence is processed to extract vehicle segments, which provides a means for vehicle tracking and classification. Secondly, vehicle segments are normalized so that all vehicles are along the same direction and uniformly scaled. A texture analysis technique is then used to filter the vehicle images. The final stage is classification, in which an ICA-based algorithm is applied. We choose ICA because of its ability to find inner characteristics of a group of data. The ICA based algorithm is compared with a well-known subspace analysis technique – PCA. Experimental results show that given a sufficient amount of sample data our system can effectively categorize vehicles at a fine granularity.

Acknowledgement

The work of Chengcui Zhang was supported in part by SBE-0245090 and the UAB ADVANCE program of the Office for the Advancement of Women in Science and Engineering.

References

1. Marco, F., Yu, H.H.: Vehicle Classification in Distributed Sensor Networks. Journal of Parallel and Distributed Computing. V64:7, (2004).
2. Sahambi, H. S., Khorasani, K. A: Neural-Network Appearance-Based 3-D Object Recognition Using Independent Component Analysis. IEEE Trans. on Neural Networks. vol. 14, no. 1, (2003), pp. 138-149.
3. Gupte, S., Masoud, O., Martin, R. F. K., Papanikolopoulos, N. P.: Detection and Classification of Vehicles. IEEE Trans. on Intelligent Transportation Systems. v3:1, (2002), pp. 37-47.
4. Fortuna, J., Schuurman, D., Capson, D.: A Comparison of PCA and ICA for Object Recognition under Varying Illumination. Proc. of 16th International Conference on Pattern Recognition. vol. 3, (2002), pp 11-15.
5. Sezer, O. G., Ercil, A., Keskinoz, M.: Subspace Based Object Recognition Using Support Vector Machines. Proc. of European Signal Processing Conference (EUSIPCO), (2005).
6. Lai, A. H. S., Yang, N. H. C.: Vehicle-Type Identification through Automated Virtual Loop Assignment and Block-Based Direction-Biased Motion Estimation. IEEE Trans. on Intelligent Transportation Systems, v1:2, (2000), pp. 86-97.
7. Wu, W., Zhang, Q., Wang, M.: A Method of Vehicle Classification Using Models and Neural Networks. Proc. of IEEE 53rd Vehicular Technology Conference. Vol. 4, (2001), Rhodes, Greece, pp. 3022-3026.

8. Harlow, C., Peng, S.: Automatic Vehicle Classification System with Range Sensors. Transportation Research Part C: Emerging Technologies. Vol. 9, No. 4, (2001), pp. 231-247.
9. Schölkopf, B., Platt, J. C. et al.: Estimating the Support of a High-dimensional Distribution. Microsoft Research Corporation Technical Report MSR-TR-99-87, (1999).
10. Chen, S.-C., Shyu, M.-L, Sista, S., Zhang, C.: Learning-Based Spatio-Temporal Vehicle Tracking and Indexing for Transportation Multimedia Database Systems. IEEE Trans. on Intelligent Transportation Systems, v4:3, (2003), pp. 154-167.
11. Zhang, C., Chen, X., Chen, W.-B.: A PCA-based Vehicle Classification Framework. Proc. of IEEE International Workshop on Multimedia Databases and Data Management, in conjunction with IEEE International Conference on Data Engineering (ICDE 2006), Atlanta, Georgia, USA.
12. Hyvärinen, A., Oja, E.: A Fast Fixed-point Algorithm for Independent Component Analysis. Neural Computation, v9:7, (1997), pp. 1483-1492.
13. Abdelbaki, H.M., Hussain, K., Gelenbe, E.: A Laser Intensity Image Based Automatic Vehicle Classification. Proc. of IEEE Intelligent Transportation Systems, (2001), Oakland, CA, U.S.A, pp. 460-465.
14. Nooralahiyan, A.Y., Kirby, H.R., Mckeown, D.: Vehicle Classification by Acoustic Signature. Mathematical and Computer Modeling, vol. 27, No. 9, (1998), pp. 205-214.
15. Lipton, A.J., Fujiyoshi, H., Patil, R.S.: "Moving Target Classification and Tracking from Real-time Video", Proc. of Fourth IEEE Workshop on Applications of Computer Vision, Princeton, NJ, USA, pp.8-14, Oct 19-21, 1998.
16. Khammari, A., Nashashibi, F., Abramson, Y., Laurgeau, C.: "Vehicle Detection Combining Gradietn Analysis and AdaBoost Classification", Proc. of 8th International Conference on Intelligent Transportation Systems, pp. 66-71, Vienna, Austria, Sept. 13-15, 2005.