An Integrated System for Moving Object Classification in Surveillance Videos

Longbin Chen
U.C. at Santa Barbara
Santa Barbara, CA 93117
lbchen@cs.ucsb.edu

Rogerio Feris, Yun Zhai, Lisa Brown, Arun Hampapur
IBM Watson Center
Hawthorne, NY 10532
{rsferis,yunzhai,lisabr,arunh}@us.ibm.com

Abstract

Moving object classification in far-field video is a key component of smart surveillance systems. In this paper, we propose a reliable system for person-vehicle classification which works well in challenging real-word conditions, including the presence of shadows, low resolution imagery, perspective distortions, arbitrary camera viewpoints, and groups of people. Our system runs in real-time (30Hz) on conventional machines and has low memory consumption. We achieved accurate results by relying on powerful discriminative features, including a novel measure of object deformation based on differences of histograms of oriented gradients. We also provide an interactive user interface, enabling users to specify regions of interest for each class and correct for perspective distortions by specifying different sizes in different positions of the camera view. Finally, we use an automatic adaptation process to continuously update the parameters of the system so that its performance increases for a particular environment. Experimental results demonstrate the effectiveness of our system in standard dataset and a variety of video clips captured with our surveillance cameras.

1. Introduction

Traditional video surveillance usually involves the use of analog cameras and video-tape storage, suffering from high cost of employing security personnel to monitor screens and watch vast amount of videos. With the advent of digital video, IP cameras, and networked video recorders, a new generation of *smart surveillance systems* is arising. These systems rely on analytic modules that use computer vision techniques to automatically extract useful information from surveillance videos, enabling the user to easily search over the data and receive real-time alerts.

Visual object classification is a key component of smart surveillance systems. The ability to automatically recognize objects in images is essential for a variety of surveillance applications, such as the recognition of products in retails stores for loss prevention, automatic identification of vehicle license plates, and many others.

In this paper, we address a simplified two-class object recognition problem: given a moving object in the scene, our goal is to classify the object into either a person (including groups of people) or a vehicle. This is a very important problem in city surveillance, as many existing cameras are pointing to areas where the majority of moving objects are either humans or vehicles. In our system, this classification module generates metadata for higher-level tasks, such as event detection (e.g., cars speeding, people loitering) and search (e.g., finding red cars in the video).

We assume static cameras, and thus benefit from background modelling algorithms to detect moving objects. In spite of these simplifications, the classification problem still remains very challenging, as we desire to satisfy the following requirements:

- Real-time processing and low memory consumption;
- The system should work for arbitrary camera views;
- Correct discrimination under different illumination conditions and strong shadow effects;
- Able to distinguish similar objects (such as vehicles and groups of people).

Our approach to address these issues consists of three elements: (1) *discriminative features*, (2) an *adaptation process*, and (3) an *interactive interface*.

In addition to conventional features like object size and velocity, we propose to use *differences of histograms of oriented gradients* (DHoG) to measure the amount of intra-object deformation from frame to frame. This is a very useful feature to differentiate vehicles from people in different camera views, scenes with shadows, and to better separate groups of people from vehicles. An adaptation process is also proposed in our framework, which can update the classification parameters of the system as new samples came. This enables the classification system to become more specialized for a target camera view and continuously improving its classification performance. An interactive interface is provided so that the user can specify regions of interest for each class and correct for perspective distortions by

specifying different sizes in different locations of the camera field of view. The use of this information in our system allows it to work for arbitrary camera viewpoints, while significantly improving the accuracy of classification.

Finally, all the three elements are integrated by means of a probabilistic framework. Our system is straightforward to implement, runs in real-time (30Hz), and provides very accurate classification results in challenging scenarios, as proven by our experimental results.

The remainder of this paper is organized as follows: related work is discussed in Section 2; Section 3 presents the proposed approach to classify moving objects into categories of people and vehicles; Section 4 explains the initialization of prior information using an interactive calibration tool; finally, experimental results are presented in Section 5, and conclusions are drawn in Section 6.

2. Related Work

Over past several decades, many different approaches have been proposed to automatically classify objects in images and videos. Bag of Words and SIFT features [11, 8] have been popular methods for large-scale classification problems, involving multiple object classes. However, these techniques are designed to handle still images with high resolutions and are not appropriate to classify moving objects in low resolution surveillance videos. We are interested in processing video captured by far-field cameras, where objects such as pedestrians may appear 30-pixel tall. For these scenarios, there are two main streams of work which differ according to whether segmentation of moving objects is applied prior to classification.

Methods that do not rely on moving object segmentation generally scan the entire video frame, applying specialized detectors (such as pedestrian or car detectors) at each image location. Examples of methods that fall under this category include the work of Viola *et. al.* [14], Dalal and Triggs [3], and Schneiderman and Kanade [12]. Learning appearance for classification may be very useful for moving cameras or crowded scenarios, where segmentation/background subtraction is difficult. However, these approaches often require excessive amount of training data to learn robust classifiers and suffer from object pose variability.

Our work falls into the category of methods that assume static cameras and use background modelling to segment moving objects prior to classification. Relevant work in this area include *shape-based* techniques [2, 1] which exploit features like size, compactness, aspect ratio, and simple shape descriptors obtained from the segmented object. Shape features are in general sensitive to the presence of shadows cast over the object (as its alters the object shape) and may not offer sufficient discrimination power in specific classification tasks, like distinguishing groups of people from vehicles.

Motion-based methods have been proposed to overcome part of these issues, mostly when the classification task involves only people and vehicles. Velocity features are important for discrimination, but are view-dependent and may be harmful when vehicles are moving slowly. A more interesting feature is based on the fact that humans often undergo non-rigid deformations compared to the rigid motion of cars. The *Recurrent Motion Image* method, proposed by Javed and Shah [7], consists in measuring the amount of deformation of the segmented foreground blob using the exclusive-or operator. Failures may occur in far-field pedestrian images where motion of the legs and arms may not be clearly captured. Also, processing the foreground blob to check non-rigid motion (rather than the image) may not be reliable when morphological operations are applied to the blobs for improving segmentation. Lipton [10] computes a dense per-pixel optical flow estimation to identify non-rigid motion, and it is a relatively expensive approach. Recently, Li *et. al.*, [9] integrated appearance-based features with approaches based on background subtraction to classify objects in far-field videos. They used Adaboost learning with *Multi-block Local Binary Patterns* and *Error Correcting Output Codes* to classify moving objects into six classes: car, van, truck, person, bike, and group of people. A large training set was required to learn a robust classifier.

Adaptation to new environments is an important property of object classifier systems in video surveillance. Bose and Grimson [1] proposed a method to first identify *scene-invariant* features and use them to design a baseline classifier. This classifier is then adapted to any given scene by learning scene-specific features with the help of unlabelled data. Appearance-based *online learning* [5] has also been used to improve classification in new environments.

3. People and Vehicle Classification

In this section, we present our approach to the problem of moving object classification in surveillance domain and describe a novel integrated system to provide an effective solution to this problem for multiple views.

3.1. Problem Settings and Challenges

In most video surveillance systems, camera views are static, i.e., there is no change in either extrinsic or intrinsic parameters of the camera. In these situations, users are often interested in the problem of differentiation between people and vehicles. The proposed classifier is able to distinguish people from vehicles, and treat individual persons and groups of people as the same object class, People.

The input of our system includes (1) input video frames, (2) background subtraction outputs, and (3) object tracking results. Since background subtraction and object tracking are not the focus of this paper, the object classifier considers them as perfect results. In actual experiments, we applied multiple track filters to eliminate noisy tracks. There are many existing approaches for background sub-

traction (BGS)[4, 6] and moving object tracking [13, 15]. In our experiments, we have used the Mixture of Gaussians for the background subtraction modelling and appearance-based blob tracking for generating object tracks.

One major challenge of the object classification task in surveillance videos is to handle different camera views and scenes, such as different view angles, zooms and lighting conditions. This causes variations in object appearance, shape, speed, and sometimes also results in long shadows along the objects. Many existing approaches only focus on a subset of these variations in a specific view, and thus, fail to address the challenge of multi-view invariance.

Another interesting challenge is to distinguish a group of people from a vehicle, which have similar shapes and sizes in the same camera view. Our system improves the classification accuracy by correctly distinguishing groups of people from vehicles via novel view-invariant feature, DHoG. This is explained in the subsequent section.

3.2. Feature Extraction

Given the limited computational resources and real-time requirement in practical video surveillance applications, the features used for object classification must be low cost and efficient for computation. In our framework, we have used four object track features. Two of these features, *object size* and *velocity direction*, are view-dependent and extracted from the BGS and tracking results.

The purpose of using object size is that size information is the most distinctive feature to distinguish single persons from vehicles since persons possess much smaller shapes than vehicles at the same location in the field-of-view. Another advantage of using object size is that the sizes of persons are more coherent, and there is less variance within the class of people compared to vehicles. The size of an object at frame i is computed as the area of the corresponding foreground motion blob and is denoted as s_i. The size vector of the entire track is defined as $\mathbf{s} = (s_0, s_1, \cdots, s_n)$, where s_0 is the object size at the starting point of the track.

In many scenarios, velocity direction of moving objects can be a distinctive feature. For instance, at a street-intersection, pedestrians typically walk along the zebra-crossings which are perpendicular to vehicle movements. The velocity direction measurement is equally discretized into 20 bins. The velocity of an object at frame i is denoted as v_i, and similar to the size vector, the velocity vector of the entire object track is defined as $\mathbf{v} = (v_0, v_1, \cdots, v_n)$.

There is an additional straightforward yet very informative feature, the *location* x_i of the moving object. The location feature relates to the context of the environment, and its usage is applied through the settings of regions-of-interests (ROIs). The ROIs define where objects of each class would appear in the camera view by object's centroid. This is a strong cue for identifying people in views such as roads and building entrances where vehicles seldom appear. The location vector of the object is denoted as $\mathbf{x} = (x_0, x_1, \cdots, x_n)$.

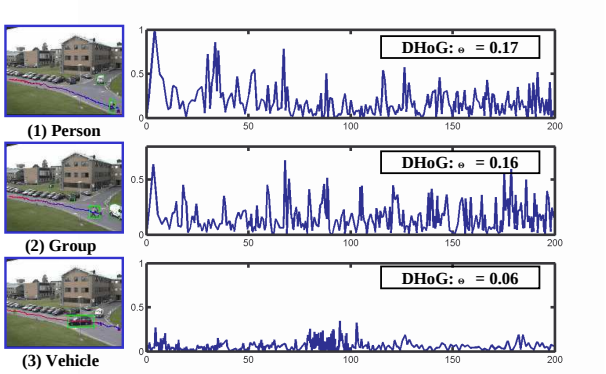

Figure 1. DHoG plots for three types of objects: *Person*, *Group of People* and *Vehicle*. Horizontal axis represents the *track length* of the samples, and vertical axis represents the DHoG values. Note that the overall DHoG values $\mathbf{\Theta}$ for *Person* and *Group of People* are much higher than the one for *Vehicle*, and thus, it is a very discriminative feature to separate people from vehicles.

Lastly, we have developed a novel view-independent feature, *Differences of Histograms of Oriented Gradients*, DHoG. Given an input video image with foreground blob mask generated by background subtraction module, the histogram of oriented gradients (HoG) is computed. It is well known that HoG is robust to lighting condition changes in the scene. HoG of the foreground object is computed at every frame in the track, and DHoG Θ is calculated in terms of the difference between HoGs obtained in consecutive frames in terms of histogram intersection,

$$\Theta_i = 1 - \sum_j \min(HoG_i^j, HoG_{i-1}^j), \qquad (1)$$

where j represents the bins in the oriented gradient histogram, and $j = 1, \cdots, 8$. The DHoG of the entire object track is formulated using both spatial and temporal information of the track data. Here we introduce the concept of *track distance*, $T = \sum_{m=1} |x_m - x_{m-1}|$. This is used to normalize the incremental distance rather than the Euclidean distance between observations. The overall DHoG $\mathbf{\Theta}$ is defined as the weighted mean of individual samples,

$$\mathbf{\Theta} = \frac{\sum_{i=1}^{n} \Theta_i \times a_i}{L}, \qquad (2)$$

where weight $a_i = |x_i - x_{i-1}|$ and the length of the entire object track $L = \sum_i a_i$. The advantage of using the weighted mean is that the observed samples with low velocities do not contribute much in the overall DHoG modelling. This is critical in scenarios where people stand still for a long time after/before walking. In this case, since the person is being still, the frame-to-frame DHoG will be small due to small deformation. The weighted mean in this case will ignore these samples and only considers the samples with significant motion.

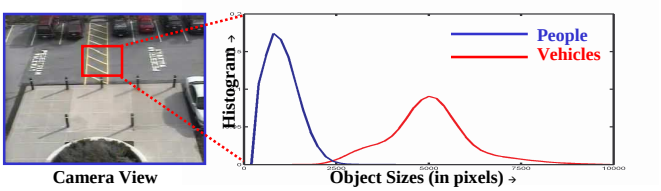

Figure 2. Object size distributions for *People* and *Vehicles* from a single camera view block (red box in the key-frame). Size models of people and vehicles are very distinctive within the same view block due to less perspective effect.

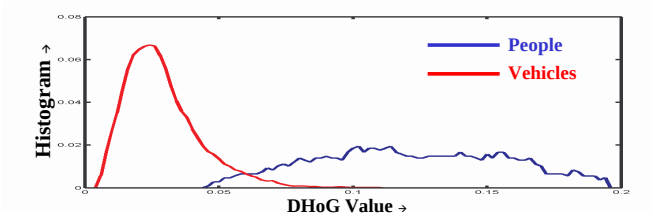

Figure 3. DHoG distributions for classes of *People* and *Vehicles*. Note that DHoG values for people are much higher and more spread than DHoG values for vehicles.

DHoG models the intra-object deformation in the temporal domain. Thus, it is invariant to different camera views. In general, vehicles produce smaller DHoG than people since vehicles are more rigid when in motion. This feature is useful to distinguish large groups of people from vehicles, in which case they have similar shapes and sizes. Examples are shown in Figure 1 to demonstrate the effectiveness of using DHoG to distinguish vehicles from people (both single persons and groups of people).

3.3. Object Classification

We pose the moving object classification task as a Maximum A Posterior (MAP) problem. The classification is performed by analyzing the features of the entire object track, i.e., classification decision is made after the tracking is finished. Let C be the class label of a target object, where $C \in \{P, V\}$ (P stands for *people* and V stands for *vehicles*). The classification is carried out by estimating the following probability,

$$\begin{aligned} p(C|\mathbf{s},\mathbf{v},\mathbf{x},\mathbf{\Theta}) &= \frac{p(\mathbf{s},\mathbf{v},\mathbf{\Theta}|\mathbf{x},C)p(\mathbf{x},C)}{p(\mathbf{s},\mathbf{v},\mathbf{x},\mathbf{\Theta})} \\ &\propto p(\mathbf{s},\mathbf{v},\mathbf{\Theta}|\mathbf{x},C)p(\mathbf{x},C). \end{aligned} \quad (3)$$

Let $\mathbb{M} = (\mathbf{s},\mathbf{v},\mathbf{x},\mathbf{\Theta})$. Given the assumed object class C and its location vector $\mathbf{x}$, the other three features, $\mathbf{s}$, $\mathbf{v}$ and $\mathbf{\Theta}$, are considered independent to each other. Therefore, Eqn.3 becomes the following,

$$\begin{aligned} p(C|\mathbb{M}) &\propto p(\mathbf{s}|\mathbf{x},C)p(\mathbf{v}|\mathbf{x},C)p(\mathbf{\Theta}|\mathbf{x},C)p(\mathbf{x},C) \\ \log p(C|\mathbb{M}) &\propto \log p(\mathbf{s}|\mathbf{x},C) + \log p(\mathbf{v}|\mathbf{x},C) \\ &\quad + \log p(\mathbf{\Theta}|\mathbf{x},C) + \log p(\mathbf{x},C)) \end{aligned} \quad (4)$$

Since DHoG $\mathbf{\Theta}$ models the self-variance of the object appearance and is assumed to be independent to each object location, Eqn.4 is further derived to be,

$$\begin{aligned} &\log p(C|\mathbb{M}) \\ &\propto \sum_{t=1}^{n} \log p(s_i|x_i,C) + \sum_{t=1}^{n} \log p(v_i|x_i,C) \\ &\quad + \sum_{t=1}^{n} \log\left[p(C|x_i)p(x_i)\right] + \log p(\mathbf{\Theta}|C), \end{aligned} \quad (5)$$

Eqn.3 is now decomposed into different likelihood and prior terms. To accommodate the perspective effect of the camera, the entire image field-of-view is equally partitioned into a set of N blocks, each of which is represented by its center location b_m. For the likelihoods/priors that use the view-dependent features (size, velocity, location), the distributions are estimated and maintained for each image block, and this is followed through out the rest of this paper. The priors and likelihoods are initialized at the beginning of the object classification process and continuously updated using the adaptation as more samples acquired.

The size likelihood $p(s_i|x_i,C)$ is specified by a Mixture of Gaussians model for each image block. There are two Gaussian models in this distribution corresponding to the person and vehicle classes respectively. The initial values of the model parameters (mean and variance) are specified by the user through the interactive calibration tool presented in Section 4, and one example distribution of object sizes in a selected view block is shown in Figure 2. The velocity likelihood $p(v_i|x_i,C)$ is initialized as uniformly distributed since there is little evidence for users to draw inference on appropriate parametric distributions to model the object velocities. The DHoG likelihood $p(\mathbf{\Theta}|C)$ is specified by using the training data acquired from multiple views. Similar to the size likelihood, the DHoG likelihood is also modelled by a Mixture of Gaussian distribution with two models corresponding to people and vehicle classes. One example DHoG distribution for *people* and *vehicles* are shown in Figure 3. The location likelihood $p(C|x_i)$ is related to the object class region-of-interests (ROIs). Given ROIs for people $\{R_P\}$ and ROIs for vehicles $\{R_V\}$, the likelihoods should satisfy the following conditions,

$$\begin{aligned} &p(C_P|x_i) > p(C_V|x_i), \text{ if } x_i \in \{R_P\} \wedge x_i \notin \{R_V\}, \\ &p(C_P|x_i) \cong p(C_V|x_i), \text{ if } x_i \in \{R_P\} \wedge x_i \in \{R_V\}, \\ &p(C_P|x_i) < p(C_V|x_i), \text{ if } x_i \notin \{R_P\} \wedge x_i \in \{R_V\}. \end{aligned} \quad (6)$$

The calibration tool presented in Section 4 also provides the users an intuitive and convenient way to specify the ROIs for classes of people and vehicles, and the location priors are initialized using the user specification.

3.4. Adaptations

In practice, specified prior information does not perform perfectly in a new camera view, e.g., due to different viewing angles or unexperienced lighting conditions. In this case, effective feature adaptation becomes critical to the object classification task. There are two aspects of the adaptation: (1) adaptation to the physical environment of the scene to accommodate different perspective effects, and (2) adaptation to particular time periods to handle different object appearances across the time. In this section, we describe how to update the likelihood information using newly acquired object samples. In our adaptation process, only objects with high classification confidence values are used to update the likelihoods, i.e., $|p(C_P|\mathbb{M}) - p(C_V|\mathbb{M})| > Th$. The following explanation of the adaptation process uses an example of classifying people. An analogous adaptation process is carried out for vehicles in the similar fashion.

The adaptation is applied to the likelihood distributions of every image block. Once an object is confidently classified to be a person, its observation $\mathbb{M}$ at each frame is used to update the four likelihood terms described in the previous section. The size likelihood is modelled using a Mixture of Gaussians with the mean $\bar{s}_{_P}$ and variance $\sigma^2_{_P}$,

$$\begin{aligned} \bar{s}_{_P} &= E(s_{_P}) = \frac{1}{Z}\sum_k s^k_{_P}, \\ \sigma^2_{_P} &= E(s^2_{_P}) - \bar{s}^2_{_P} = \frac{1}{Z}\sum_k (s^k_{_P})^2 - \bar{s}^2_{_P}. \end{aligned} \quad (7)$$

where Z is the total number of observed size samples for people class. If the terms $\sum_k s^k_{_P}$ and $\sum_k (s^k_{_P})^2$ are calculated in an incremental fashion, there is no need to store all the samples. The same updating mechanism is applied in the adaptation for the DHoG likelihood $p(\mathbf{\Theta}|C)$.

For the velocity likelihood $p(v|x, C_P)$, an orientation histogram is constructed and $p(v = \nu|x, C_P) = f(\nu|x, C_P)$, where $f(\nu|x, C_P)$ is the frequency of velocity direction ν at image block x for the people class. Thus, adaptation of the velocity likelihood computation is to update the frequency values of the orientation histogram.

Different from the above likelihoods which are updated using an exact computation, adaptation of location likelihood is carried out approximately. If an object is classified as a person, the location likelihood update is defined as,

$$\begin{aligned} p^{new}(C_P|x) &= min[1, (1-\alpha)p^{old}(C_P|x) + \alpha], \\ p^{new}(C_V|x) &= (1-\alpha)p^{old}(C_V|x). \end{aligned} \quad (8)$$

where α is a fixed update factor. Similarly, the same update schema applies to the situation when vehicles are detected. The reason for using a fixed update factor α is to make the object classification system adapt to the more recently acquired samples. By doing this, the classifier can also adapt to the temporal context of the camera view. This is particularly useful in places where people and vehicles occupy the same area in different time periods. For instance, in a shopping area, only pedestrians are allowed to walk along the street during the daytime. However, after store closing time, as amount of people is significantly reduced, loading/shipping vehicles start coming into and going out of the scene. In this example, using fixed updating factor gives a favorable bias to the vehicle samples during the night time surveillance even if they appear in areas which are initially declared to be people ROIs.

4. Setting Likelihoods Using Calibration Tool

For prior information that is computed using view-dependent features (size, location, velocity, etc), it is very inconvenient for the user to specify the proper initializations for every camera view. In this section, we present a graphical user interface (UI) tool which provides users an efficient way to specify appropriate calibration settings of the target camera view in an interactive way. In particular, the calibration tool helps the user to initialize two likelihoods which were discussed in the previous section: (1) $p(C|x_i)$, the probability of an object class at a given location in the image, and (2) $p(s_i|x_i, C)$, the probability of tracked blob size given the object class and location. As aforementioned, camera view is partitioned into a grid of N blocks. The center of image block i is taken as location x_i.

4.1. Region of Interests (ROI)

In many situations, objects of one class are more likely to appear in certain regions in the camera view. For instance, in the city street environment, people usually walk along the sidewalk, while on the other hand vehicles mainly run in the middle of the road. This is a strong cue in the object classification process. In our system, we classify tracked objects into two classes, people (C_P) and vehicles (C_V), and users can specify the ROIs of each object class in the camera view through the calibration tool. In this calibration tool, one or multiple ROIs for the target class can be created, modified and deleted as needed. Screen-shots of the interface are shown in Figure 4.

Let the label of an object class be C_k, $k \in \{P, V\}$ and its complement class be defined as $C_{k'}$. The region-of-interests of object class C_k is defined as $\{R_k\}$. Similarly, ROIs for the complement object class is defined as $\{R_{k'}\}$. Thus, the likelihood $p(C_k|x_i)$ of a given image location x_i is computed as follows,

$$\begin{aligned} p(C_k|x_i) &= 1.0, \text{ if } x_i \in \{R_k\} \wedge x_i \notin \{R_{k'}\}, \\ p(C_k|x_i) &= 0.5, \text{ if } x_i \in \{R_k\} \wedge x_i \in \{R_{k'}\}, \\ p(C_k|x_i) &= 0.0, \text{ if } x_i \notin \{R_k\}. \end{aligned} \quad (9)$$

These likelihoods are later updated by the adaptation process. For instance, if vehicles are detected in the exclusive ROIs for people, likelihood $p(C_P|x)$ will be adjusted to a lower value while $p(C_V|x)$ is increased (Sec.5.3).

(a) ROI for People (b) ROI for Vehicles (c) Size Models

Figure 4. Calibration tool interface. (a) ROI for people - entire camera view. (b) ROI for vehicles - drive ways. (c) Person size models specified by the user. In (c), user can create, move and resize a target size sample (the largest box in the figure).

4.2. Object Size Normalization

Due to the more consistent appearance of people in the image, we focus on the person size models and use them for object size normalization of all object classes. Our calibration tool provides an easy and intuitive way to assist the user to specify possible person size information in the image field-of-view. To obtain a person size sample, the user can use the mouse to add a person model at the target location in the image. The person model is further moved and resized to the desired location and size, such that the camera perspective effect at that image location is properly approximated. One example of the person models is shown in Figure 4.c, where six person size samples are specified across the image. The specified person size sample is denoted as $\hat{s}_m, m \in [1, M]$, where M is the total number of person size samples. Its location is $\hat{x}_m$. For the purpose of effectiveness, users are usually required to specify at least five samples across the image.

Once sufficient person size models are specified by the user through the calibration tool, the predicted size distribution of each image block is initialized by interpolating the specified size samples. Given a target image block x_i, its predicted mean person size is denoted as $\bar{s}_i$ computed as,

$$\bar{s}_i = \sum_{m=1}^{M} w_m \times \hat{s}_m, \tag{10}$$

where w_m is the interpolation weight for size sample $\hat{s}_m$. It should satisfy $w_m \propto \frac{1}{\|x_i - \hat{x}_m\|}$ and $\sum w_m = 1$. Based on this interpolation, the object size likelihood at a given location for the object classes is defined as,

$$p(s_i|x_i, C_P) = N(s_i|\bar{s}_i, \sigma^2), \tag{11}$$

$$p(s_i|x_i, C_V) = N(s_i|\alpha\bar{s}_i, \sigma^2), \tag{12}$$

where $N(a|b, c)$ represents the probability of value a in a Gaussian distribution with mean of b and variance of c, and α is a scaling factor to separate models for people and vehicles. For the initialization, scaling factor α and distribution variance σ^2 are set to be constant values for both classes. (Note: values of α and σ^2 can also be learned from previous empirical statistics.) As described in the previous section, the mean and variance of the object classes are later updated by the adaptation process.

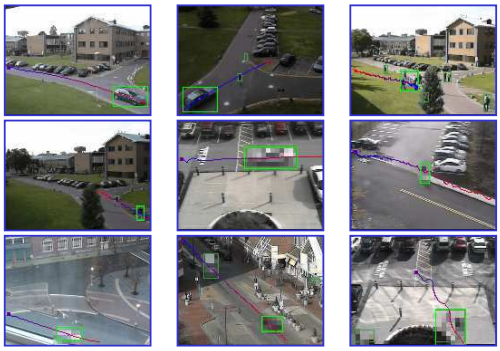

Figure 5. Example keyframes of testing videos. Dataset covers various scenes and different views of the same scene.

5. System Performance Analysis

5.1. Experimental Results

To demonstrate the effectiveness of our proposed object classification framework, we have tested our system on a variety of video data, including different scenes and different views of the same scene. There are two data sets used in our experiments. The first set is the PETS 2001 dataset. The second data set is a collection of videos obtained from different sources. Some keyframes of the testing videos are shown in Figure 5. The testing videos range from 15 minutes to 7 hours long, and they are resized to the same dimension of 320x240. For computation of location-related likelihoods, the camera views are equally partitioned into grids of blocks with size of 20x20, and distributions are established for each block. The experiments are performed using all presented features (object size, velocity, location/ROI and DHoG) and the feature adaptation process. Currently, all features have equal weights in the inference. Different weights can also be applied to achieve better performance.

Table 1 presents the object classification results on both testing data sets. This table shows the groundtruth (GT)

Table 1. System performance summary.

Videos	GroundTruth		People Detection		Vehicle Detection	
	People	*Vehicles*	*True Positives*	*False Positives*	*True Positives*	*False Positives*
PETS D1TeC1	6	2	6	0	2	0
PETS D1TeC2	5	1	5	0	1	0
PETS D2TeC1	6	3	6	0	3	0
PETS D2TeC2	7	2	6	0	2	1
PETS D3TeC1	16	0	10	0	0	6
PETS D3TeC2	13	0	6	0	0	7
Sequence HW1	16	82	14	1	81	2
Sequence HW2	67	22	66	0	22	1
Traffic Complex	125	696	121	0	696	4
Traffic VideoD	44	46	40	1	45	4
Shadow	3	1	3	0	1	0

(a) Long Shadow **(b) Group of People** **(c) Incorrect Tracking**

Figure 6. Correct classification results for long shadow (a) and group of people (b). The major reason that classification fails is incorrect tracking results (c).

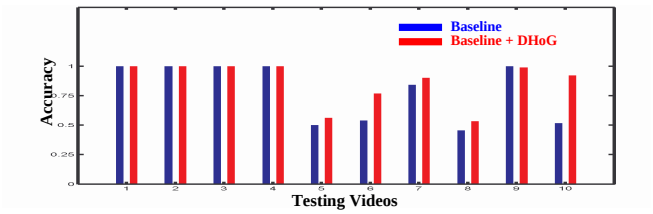

Figure 7. Performance comparison between using only baseline features and using both baseline features and DHoG.

of the data set and the classification results for *people* and *vehicles*, including true positives (TP) and false positives (FP). There are total 308 people and 855 vehicles in the groundtruth. PETS dataset has more people than vehicles, while dataset 2 contains more vehicles since they are mostly recorded from street scenes. The accuracy measurement is defined as follows,

$$\text{Accuracy} = \frac{\text{Total Number of True Positives}}{\text{Total Number of Groundtruth}}. \quad (13)$$

Based on the results shown in Table 1, the overall classification accuracy is 97.7% for over 1100 moving objects. Some of the successful cases are shown in Figure 6, where challenging scenarios, such as a person with long shadow (Figure 6.a) and a group of people (Figure 6.b), are correctly classified as *People*.

Currently, our framework assumes tracking results are perfect before performing object classification. Thus, if the background subtraction and/or tracking module fail to produce reliable results, the object classifier will also fail due to incorrect input data. This is the main reason for the lower performance on two sequences "PETS D3TeC1" and "PETS D3TeC2". One failing case is also shown in Figure 6.c, where tracking of a mass of crowd failed.

5.2. Feature Effectiveness Analysis

We have demonstrated the effectiveness of the overall system. In this section, we analyze the improvement caused by each individual features. Based on the empirical observations, we have drawn the conclusion that *size* and *velocity* play the major role in the classification process and combination of these two features provides over 60% classification accuracy in most videos. Thus, we consider these two features as our baseline features and focus our analysis on the effectiveness of *DHoG* and *location*.

Figure 7 shows the comparison between using base classifiers only and using both base classifiers and DHoG. Overall, using DHoG improved the classification accuracy by ∼10%. The major contribution of DHoG is to properly classify persons with shadows and distinguish groups of people from vehicles. There are a few incorrectly classified objects when using DHoG. The reason is that DHoG models the intra-object deformation based on object's global information (oriented-gradient histogram). Thus, if the target object is partially or completely occluded, DHoG's performance degrades since only partial information is available. One solution to this problem is to neglect the object samples when object is in occlusion and only uses fully observed samples for classification inference.

Figure 8 shows the performance comparison between using baseline features and using both baseline features and location information. The improvement of using regions-of-interests (ROIs) is prominent in scenarios where people and vehicles have relatively separate areas. In this case, ROIs for people and vehicles have little or no overlap, and

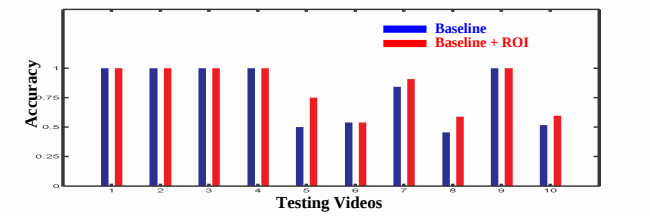

Figure 8. Performance comparison between using only baseline features and using both baseline features and ROI.

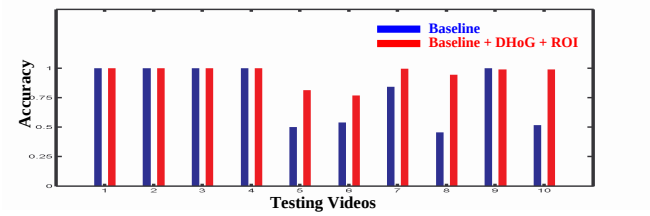

Figure 9. Performance comparison between using only baseline features and using all proposed features.

thus, provide strong prior for the classification process. On the other hand, when the people and vehicle ROIs are mixed together and have large percentage overlap, object location information becomes less distinctive (Eqn.6). The performance comparison between baseline features and all proposed features is presented in Figure 9.

5.3. Feature Adaptation

We have also performed accuracy analysis on using the proposed feature adaptation technique. Due to the space limitation of the paper, here we only present the results of applying adaptation on object size information. The size models for people and vehicles were intentionally improperly initialized, so that the classifier would provide lower accuracy. As more objects detected, the size model parameters (mean and variance) adapted to the correct values, and the classification accuracy improved dramatically. To demonstrate the effectiveness of our proposed adaptation technique, we have tested on an hour-long video segment with 495 moving objects. The accuracy measurement is calculated at the points of obtaining 8, 16, 32, 64, 128, 256 and 495 samples. The adaptation result is shown in Figure 10. The adaptation of size information improved the classification accuracy by over 20%.

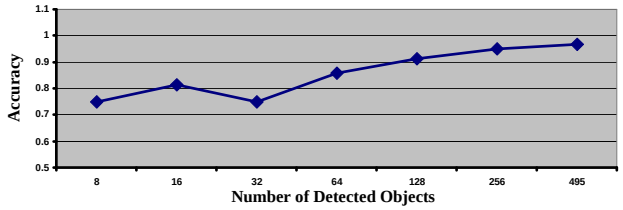

Figure 10. Performance plot of using adaptation on object size update.

6. Conclusions

In this paper, we have presented an integrated framework to classify moving objects into categories of *People* and *Vehicles*. The proposed framework incorporates four video features, object size, object velocity, location and difference of histogram of oriented gradients (DHoG). These features are easy to implement and computationally inexpensive. To make the features more suitable for newly deployed camera views, an adaptation process is applied. This enables the classification system to accommodate both new views and different time periods in the same view. The proposed classification is tested on a large dataset with over 1100 moving object and has achieved very high accuracy. In addition, we have designed an interactive user interface to assist users to initialize model priors. Users can specify region-of-interests for the object classes and person size samples to calibrate the camera view.

References

[1] B. Bose and E. Grimson. Improving object classification in far-field video. In *CVPR*, 2004. 2

[2] L. Brown. View independent vehicle/person classification. In *VSSN*, 2004. 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[4] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *ECCV*, 2000. 3

[5] H. Grabner and H. Bischof. Online boosting and vision. In *CVPR*, 2006. 2

[6] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *WMVC*, 2002. 3

[7] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *ECCV*, 2002. 2

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2

[9] L. Li, S. Yuan, and S. Xiang. Real-time object classification in video surveillance based on appearance learning. In *CVPR*, 2007. 2

[10] A. Lipton. Local application of optic flow to analyse rigid versus non-rigid motion. In *CMU Technical Report*, 1999. 2

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(1):91–110, 2004. 2

[12] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied for faces and cars. In *CVPR*, 2000. 2

[13] A. Senior. Tracking with probabilistic appearance models. In *PETS*, 2002. 3

[14] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, 2003. 2

[15] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based-on static body part detection. In *CVPR*, 2006. 3