Unsupervised Model Selection for View-Invariant Object Detection in Surveillance Environments

Behjat Siddiquie
SRI International
behjat.siddiquie@sri.com

Rogerio S. Feris Ankur Datta
IBM Research
{*rsferis, ankurd*}*@us.ibm.com*

Larry S. Davis
University of Maryland
lsd@cs.umd.edu

Abstract

We propose a novel approach for view-invariant vehicle detection in traffic surveillance videos. Instead of building a monolithic object detector that can model all possible viewpoints, we learn a large array of efficient view-specific models corresponding to different camera views (source domains). When presented with an unseen viewpoint (target domain), closely related models in the source domain are selected for detection based on a novel discriminatively trained distance metric function, which takes into account scene geometry, vehicle motion patterns, and the generalizing ability of the models. Extensive experimental evaluation on a challenging test set, consisting of images collected from fifty different surveillance cameras, demonstrates that our unsupervised approach can outperform complex methods that utilize labeled training data from the target domain, both in terms of speed as well as accuracy.

1 Introduction

This paper adresses the problem of vehicle detection in urban surveillance environments. Traffic surveillance cameras are becoming increasingly widespread. Government agencies seek to use such cameras not just for monitoring traffic but also to search for suspicious vehicles, which requires accurate detection and localization of each vehicle. However, detection and localization of vehicles in surveillance video, which is typically low resolution, is extremely difficult as it requires dealing with view-invariance, varying illumination conditions (e.g. sunlight, shadows, reflections, rain, snow) and high density traffic situations, where vehicles tend to partially occlude each other.

There exist many methods for view-invariant object detection [3, 8, 4, 5, 7, 6]. However, some of these approaches restrict themselves to learning appearance models for a small number of fixed viewpoints [3, 8] and suffer a performance drop when presented with a new viewpoint. Although this can be overcome by learning models for a large number of viewpoints, doing so considerably reduces the detection speed as models for each viewpoint need to be evaluated. Likewise, methods capable of detecting objects from previously unseen viewpoints [4, 5, 7, 6] are quite slow and therefore unsuitable for real-time applications.

In order to perform fast view-invariant object detection, we propose a novel approach which exploits scene layout and geometry to select apropriate models for detection in an unsupervised manner. Instead of building a view-invariant detector that can model all possible viewpoint deformations, which is extremely hard, we train simple object detectors for a large number of different viewpoints (source domains) that densely span the viewpoint space that we want to model. Given a new viewpoint (target domain), we exploit scene geometry and vehicular motion patterns to find closely related viewpoints from the source domain where vehicles are expected to occur in poses similar to the target viewpoint. Our dense representation in the viewpoint space ensures that we are guaranteed to find closely related viewpoints in the source domain. We then transfer the knowledge learnt a priori on the selected viewpoints for detecting vehicles in the new viewpoint. To match a new viewpoint to relevant viewpoints in the source domain, we learn a distance metric which, in addition to vehicle pose, also takes into account the generalizing ability of the detectors trained on the viewpoints in the source domain. The details of our proposed approach are described in the next section, followed by a comprehensive experimental validation on a real-world dataset.

2 Proposed Approach

Training Dataset Collection: We have collected more than 400 hours of video from 50 different traffic surveillance cameras in Chicago. We adopted a simple semi-automatic procedure, which involved performing background subtraction on surveillance videos and extracting image regions corresponding to foreground blobs with pre-specified ranges of size and motion direction, with the false positives being manually removed. This simple procedure enables us to collect a large number of images of vehicles(about 220,000) in a variety of poses and illumination conditions, while requiring minimal supervision. We utilize the motion direction of each foreground blob for categorizing the images of vehicles of each camera viewpoint into a set

of clusters. The clustering leads to a categorization of the training images into a two level hierarchy, where the first level of categorization is according to the camera viewpoint and the second level is based on the motion-direction within each camera viewpoint. Since all the camera viewpoints are distinct, each leaf node of our hierarchy consists of training images of vehicles in a distinct pose. On an average, each camera viewpoint has about two clusters, resulting in a total of about 100 clusters(leaf nodes of the hierarchy) which is an extremely diverse collection of vehicles in different poses.

Object Pose Parametrization: We parametrize the pose of the vehicles within each cluster, in terms of their zenith (ϕ) and the azimuthal angles (θ) with respect to the camera. The zenith angle can be estimated based on the position of the horizon and the azimuthal angle can be approximated by the motion direction of vehicles with respect to the camera.

Horizon Estimation: We estimate the position of the horizon in each camera view. Since our task is that of detecting vehicles in a traffic surveillance setting, our images consist of urban environments, which enables us to utilize the inherent structure present in such scenes to infer their 3D geometry. We use the recently proposed, geometric image parsing approach [1] by Barinova et al. which has attained the best performance on the task of horizon estimation, on two different urban datasets.

Motion Pattern Estimation: For each camera viewpoint, we estimate the direction of motion of vehicles appearing in that scene. For this purpose, we collect a five minute($\sim$ 9000 frame) video clip of the scene. We found that a clip of this duration is sufficient for capturing the regular motion patterns that occur at an intersection. We follow an approach similar to that of Yang et al. [9], who employ a clustering based method for discovering motion patterns in video. The clusters so obtained represent the different directions of motion of vehicles appearing in the scene. We represent each cluster by the dominant direction of motion of the points within it and by its location in the image plane.

The pose of a vehicle can be defined in terms of its azimuthal angle θ and the zenith angle ϕ with respect to the camera. We assume there is no camera roll, as it can be easily rectified based on our estimation of the horizon. One can represent the variation in the pose of vehicles within a particular motion cluster of a camera viewpoint, in terms of the ranges of the zenith and azimuthal angles of the vehicles appearing in it. We define (u_c, v_c) as the optical center of the camera in the image plane and v_0 as the y-coordinate of the horizon. Let v_{min} and v_{max} respectively denote the upper and lower extent of a cluster in the $y-$direction (Figure 2), then the range of zenith angles ϕ (Figure 1) of vehicles appearing in that cluster can be defined as:

$$\phi_{\text{max}} = \tan^{-1}\left(\frac{v_{\text{max}}-v_c}{f}\right) + \tan^{-1}\left(\frac{v_c-v_0}{f}\right) \quad (1)$$

$$\phi_{\text{min}} = \tan^{-1}\left(\frac{v_{\text{min}}-v_c}{f}\right) + \tan^{-1}\left(\frac{v_c-v_0}{f}\right) \quad (2)$$

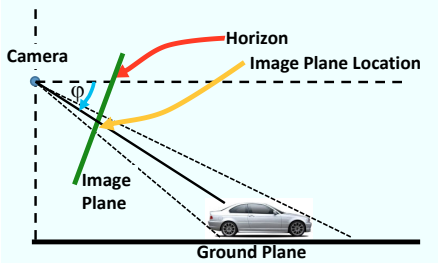


Figure 1: The zenith angle of a vehicle w.r.t. the camera.

where f is the focal length of the camera. Here the assumption is that the optical center of the camera (v_c) lies below the location of the horizon in the image plane (v_0). The equations are similar in case the reverse is true. Note that these equations are valid even when the image plane is not perpendicular to the horizon. We also compute the maximum (θ_{max}) and minimum (θ_{min}) directions of motion of vehicles with respect to the camera, based on the optical flow, and use them to approximate the azimuthal angles of vehicles within the motion cluster (Figure 2). Hence the pose of the vehicles appearing in a cluster c_i can be represented in terms of the range of their zenith angles with respect to the camera ($\mathbf{A_i} = [\phi_{max}\ \phi_{min}]$) and the range of the direction of motion with respect to the camera ($\mathbf{Z_i} = [\theta_{max}\ \theta_{min}]$).

Transferring Object Detection Models: During the training phase we build models for recognizing vehicles in a variety of poses that are present in different camera viewpoints (source domains). As described previously, our training dataset has been categorized into a two level hierarchy, with each leaf node representing vehicles traveling in a specific direction as seen from a particular camera viewpoint. We train a Deformable Parts Model (DPM) [3] based object detector DPM_s corresponding to each leaf-node cluster c_s. While we chose DPM based detectors because they have consistently achieved the best performance on several object recognition benchmarks [2], our approach allows for using any off-the-shelf object recognition system.

Given a video captured from a previously unseen camera viewpoint (target domain), we first estimate the position of the horizon and compute the motion patterns of vehicles appearing in the scene. Corresponding to each cluster c_i, we then compute the range of azimuthal angles $\mathbf{A_i}$ and the range of zenith angles $\mathbf{Z_i}$. Since our source data contains a large number of camera viewpoints each of which contains vehicles moving in multiple directions, we have DPM based object detectors trained for a large number of possible poses. Hence for each motion cluster c_i in the target view, we simply select the object recognition model from the source view that is likely to contain vehicles in the same pose and directly use it to detect vehicles in the target view. As discussed earlier, the vehicle pose is a function of the direction of motion of vehicle with respect to the camera $\mathbf{A_i}$ and the zenith view direction $\mathbf{Z_i}$. While choosing a motion cluster c_j in the source domain, apart from the

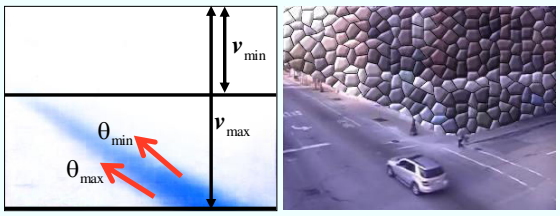


Figure 2: **Camera Viewpoint Parametrization:** The range of the azimuthal angles of a vehicle w.r.t. the camera $(\theta_{max}, \theta_{min})$. v_{max} and v_{min} denote the max. and min. y-coordinates of the motion cluster, determining the range of the zenith angles of the vehicles (Eq. 1,2).

vehicle pose, another important consideration is the size of the training set used for learning DPM_j. In general, training on a larger amount of data, leads to a better generalization. Based on all these factors, given a cluster c_i in the target domain, we can choose a cluster c_j in the source domain $\mathcal{S}$ and transfer its object recognition model DPM_j for detecting vehicles in the source domain according to the following criterion:

$$\begin{aligned} \text{DPM}_j \quad &= \arg\min_{j \in \mathcal{S}} w_a ||\mathbf{A_i} - \mathbf{A_j}||_2 \\ &+ w_z ||\mathbf{Z_i} - \mathbf{Z_j}||_2 + w_s \left(1 - \tfrac{|S_j|}{|S_{max}|}\right) \end{aligned} \quad (3)$$

where w_a, w_z and w_s are the relative weights assigned to the difference in the azimuthal direction $\mathbf{A}$, the difference in the motion direction $\mathbf{Z}$ and the relative size of the training dataset $|S|$ corresponding to cluster c_j. These weights are chosen by cross-validation. $|S_j|$ is the cardinality of the training set of cluster c_j and $(|S_{max}| = 20000)$ is the cardinality of the largest cluster. The third term is a penalty term which attempts to avoid selecting DPM models trained on small amounts of data by penalizing them. While our approach is exceedingly simple, our experiments demonstrate that given a large and diverse set of source domains $\mathcal{S}$, our approach can outperform a DPM based object detector that utilizes labeled data from the target domain.

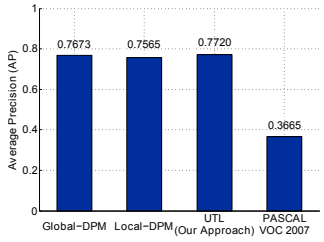
3 Experiments and Results

In order to evaluate our approach, we collected a test dataset consisting of about 3000 images obtained from the same set of 50 cameras that were used for collecting the training data. In order to evaluate our unsupervised model selection approach, we adopt a leave-one-out scheme, where each stage involves treating a particular camera viewpoint as the target domain and the remaining cameras as the source domains. Given a target camera viewpoint, the most appropriate object detection models are chosen from the source domain according to the distance criterion (Equation (3)); we refer to this approach as Unsupervised Model Selection (*UMS*). We follow the same experimental protocol that was used in the PASCAL VOC 2006 challenge. Different models are compared based on the Average Precision (AP) of their precision-recall curve on the test set.

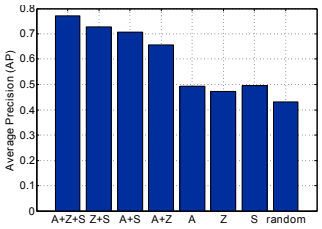
Comparison to Target-Domain Models: We compare the performance of our approach (UMS) against three different methods - *Local-DPM* and *Global-DPM* - that utilize training data from the target domain, and a DPM based model trained on the PASCAL VOC 2007 training set. In *Local-DPM*, for each camera viewpoint, we build two component DPM models for each motion cluster in the viewpoint. These models are then evaluated on test images captured from the same viewpoint. The *Local-DPM* method represents the performance of the DPM object recognition model which has access to training data from the target domain. In the case of *Global-DPM*, we utilize all the training images from each camera viewpoint to learn a DPM based object recognition model. The number of components in *Global-DPM* was set to eight as it resulted in the best performance. The *Global-DPM* approach, in addition to training data from the target domain, also utilizes training data from all the other source domains. The results are shown in Figure (3a). We can see that *UMS* which is unsupervised w.r.t. the target domain, outperforms *Local-DPM* and *Global-DPM*, which utilize labeled training data from the target domain. While it may seem surprising that our approach can outperform *Local-DPM*, which has access to training data from the target domain, the size of the local training dataset plays an important role. In some cases a model trained on a slightly different viewpoint but with a larger amount of training data can outperform a model trained on the same viewpoint. At the other extreme, simply learning a model from the entire training data might also be suboptimal as indicated by the performance of *Global-DPM*, which is slightly less than *UMS*. We conjecture that *Global-DPM* is disadvantaged by its grouping of the components based on the aspect ratio of training images instead of a more semantic criterion(e.g. the viewpoint/motion-cluster hierarchy used by us).

UMS also offers a significant speedup over view-invariant methods which attempt to learn appearance models of all viewpoints simultaneously, such as *Global-DPM* or the discriminative mixture-of-templates [4]. *UMS* selects a two component local DPM model corresponding to each motion cluster in a viewpoint. Each camera viewpoint contains two motion clusters on average, hence *UMS* requires evaluation of four DPM components resulting in a **speedup by a factor of two** over *Global-DPM*, which consists of an eight component DPM model.

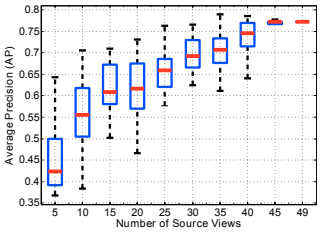
We also compare our approach to a DPM model trained on the *car* class of the PASCAL VOC 2007 training set [2]. However, its performance was substantially poor compared to models learnt using our training data (Figure 3a). While this is not a completely fair comparison, it demonstrates that training on high quality in-domain data can have a significant impact on performance, and that our approach offers an effective mechanism for transferring information from a closely related source domain to the target domain.



(a) Unsupervised Model Selection vs. Supervised approaches



(b) Performance of different combinations of the distance measure



(c) Performance variation with amount of source-domain data

Figure 3: Performance of our Unsupervised Model Selection (UMS) approach.

Figure 4: Examples of vehicle detection. Note: Background regions have been obfuscated for legal/privacy reasons.

Distance Measure: For each motion cluster in a new camera viewpoint, our approach(*UMS*) utilizes a distance measure (Equation (3)) to identify the most appropriate DPM models from the source domain. The distance measure consists of three components: the difference between the Azimuthal direction (A) of the vehicles in the two clusters, the difference between the Zenith angle (Z) of the vehicles and the relative size (S) of the training set of the motion cluster in the source domain. In order to demonstrate the importance of each of the three components that comprise our distance measure, we compare our approach *UMS* using all three components of the distance measure(A+Z+S) for selecting the most appropriate models, against all possible combinations. The results (Fig. 3b) indicate that using all three components together significantly outperforms all other combinations using just one or two components, confirming the significance of each components comprising the distance measure. Randomly choosing the DPM models corresponding to each motion cluster from the source domain results in the lowest accuracy.

Amount of Source-Domain Data: To study how the amount of source-domain data affects our approach, we evaluate the performance of our method by transferring recognition models for a given target domain from a subset of k randomly chosen source domains (camera viewpoints). Figure 3c plots the performance of our approach as a function of the number of source views(k). The largest value of k is 49, which corresponds to the case when all the camera viewpoints other than the target camera are treated as source domains. When $k < 49$, the performance is the mean over fifty runs, with a set of source domains of size k being randomly chosen during each run. The performance of our approach increases with an increase in the number of training viewpoints. This is expected, as a larger number of camera viewpoints implies a higher probability of there existing source viewpoints with vehicles in poses that closely match poses of vehicles in the target domain. Some detection results on images captured from different camera viewpoints are shown in Figure (4).

4 Conclusion

We have presented an approach for view-invariant vehicle detection in traffic surveillance videos, which learns a large number of view-specific detectors during the training phase and given an unseen viewpoint exploits scene geometry and vehicle motion patterns to select a particular view-specific detector for object detection. The key advantage of our approach is that it enables utilization of fast and simple view-specific object detectors for accurate view-invariant object detection. **Acknowledgement:** This research was funded in part by the ONR MURI grant N000141010934.

References

[1] O. Barinova, V. Lempitsky, E. Tretyak, and P. Kohli. Geometric image parsing in man-made environments. *ECCV*, 2010.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 2010.

[4] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. *ECCV*, 2010.

[5] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. *ICCV*, 2007.

[6] H. Su and et al. Learning a dense multi-view representation for detection, viewpoint classication and synthesis of object categories. *ICCV*, 2009.

[7] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. *CVPR*, 2006.

[8] A. Torralba, K. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. *CVPR*, 2004.

[9] Y. Yang, J. Liu, and M. Shah. Video scene understanding with multi-scale analysis. *ICCV*, 2009.