HIGHWAY TRAFFIC SURVEILLANCE BY UNSUPERVISED LEARNING

1ARUNKUMAR, 2S. MISHRA 3K. T. TALELE

1Scholar Student of Sardar Patel Institute of Technology, Andheri West, Mumbai University
2Associate professor of Sardar Patel Institute of Technology, Andheri West, Mumbai University
Email: arun3946@gmail.com, taleeesir@gmail.com

Abstract: This paper presents an approach to a fine method to detect vehicles in video sequences. The detection process is divided into two stages: ROI (region of interest) generation stage and ROI classification stage. In the first stage Haar-like features are exploited to rapidly search the whole image and find interesting regions which may contain the vehicle. This method incorporates well-studied computer vision and machine learning techniques to form an unsupervised system, where particular vehicles (cars, SUVS and heavy vehicles) are automatically "learned" from video sequences. An adaptive background mixture model is used to identify the positive vehicles; then a classifier with trained examples is used for detection wherein both background subtraction and the classifier are used to achieve very accurate results while not compromising efficiency. The proposed method exploits robust vehicle detection to allow for particular vehicle tracking in high density traffic. This method is tested for Mumbai’s Roads (including National Highways NH-8 and NH-17, regular roads as SV road and Link road) in various situations as for day time, night time, sunset and under high traffic. The proposed surveillance is also tested in low, medium and high resolution with high level of accuracy and precision. The proposed model is giving an acceptable and high range of detection accuracies from 87% to 94%.

Keywords: Vehicle Detection, Highway Surveillance, Tracking.

I. INTRODUCTION

Automated traffic surveillance involving vehicle detection has been an important component of highways and law & policy making systems for decades. Disadvantages of loop detectors have spurred further research in vehicle detection. Also computer vision approach is quickly becoming popular alternatives. Videos surveillances are not only having lower maintenance costs, but are also capable of providing richer traffic information than their inductance loop counterparts. Speed, queue lengths, and individual vehicle delay can be extracted from video images with proper video detection algorithms. In detection process, different schemes like shape, length, width, texture, etc., are extracted for vehicle classification. However, vehicle occlusion will result in the failure of vehicle detection and further degrade the accuracy of vehicle classification and counting. Therefore, before vehicle recognition, a process of camera calibration for normalizing features should be applied in advance. Advancement of vision-based vehicle detection has triggered recent development of automated traffic surveillance in order to automatically detect and identify the moving vehicles in complex traffic scene. Vehicle detection can be developed to track and follow a car. Numerous vision-based techniques have been developed over the past decade to detect vehicles in various road scenes, as described in [1]-[11]. Tzomakas and Seelen [2] detected vehicles based on the shadows underneath them; and therefore, an adaptive threshold was applied to extract vehicle edges. Chang et al. [3] presented vehicle detection and tracking system by finding the footprint of a vehicle on road area and tracking the vehicle using the continuity measurement of two consecutive frames. Kate et al. [4] combined the knowledge-based methods such as shadow detection, entropy analysis and horizontal symmetry measurement for mid-range and distant vehicle detection without prior knowledge about the road geometry. Besides that, Khammari et al. [5] applied a horizontal Sobel filter on the 3rd level of the Gaussian pyramid to obtain local gradient maxima where a vehicle candidate is located. A temporal filter was used to further remove unwanted pixels and then a bounding box extraction was employed to retrieve a possible vehicle region for symmetry verification. Broggi et al. [6] presented a multi-resolution vehicle detection method to localize vehicles with variable sizes. They computed the symmetry property of vehicles in different sized bounding boxes on all the columns of the regions. Liu et al. [8] detected vehicle region based on the shadow underneath a vehicle and symmetry edges. Additionally, they combined knowledge-based and learning-based methods for vehicle verification. In vehicle tracking, templates were dynamically created on-line and tracking window was adaptively adjusted with motion estimation. On the other hand, Hoffman [9] performed a multi-sensor fusion approach incorporating 2-D visual features such as shadow and symmetry, with 3-D ground plane information such as camera height for vehicle detection. However, the presence of non-vehicle structures such as overbridge, fly-over roadway, tunnel, buildings, sign boards etc, in traffic scene may decrease the performance of knowledge-based vehicle detection since these non-vehicle structures posses the horizontal/vertical characteristics identical to...
vehicle’s edges [2]-[8]. Moreover, a complex road environment may complicate the process of vehicle detection as there are many possibilities of human activities along the road side [4]. Furthermore, the requirement of 3-D transformation and the knowledge of hardware parameters for stereo-based vehicle detection method have highly increased the computational cost and reduced the processing speed [9][10]. This paper presents a vehicle detection algorithm which addresses most of the aforementioned issues related to low quality, crowded surveillance videos. In Section 2 the proposed approach is explained in detail. In Section 3 experimental results are presented. Discussion and future work is included in Section 4.

II. METHODOLOGY

Figure 1 System Overview

Figure 1 illustrates the overview of the proposed system. In the learning phase, some positive vehicle examples are identified directly from the background subtraction. Then the sets of positive and negative vehicles image patches are used to train a classifier. The result is a vehicle -specific binary classifier, which is used to further process the blobs (moving vehicle patches) that are suspected to have more than one vehicle. The details of each step are explained in the following sections.

2.1. Background Subtraction

For background subtraction, an adaptive background mixture model [12] is used. The adaptive background mixture model uses a mixture of Gaussians (MoG) to model the background of a scene. Analysis is done for both spatial and temporal information in the background model, and hence subtraction results in extremely good foreground masks. Additionally, the background model “adapts” itself to the changing lighting and weather conditions. Figure 2 illustrates a sample input and the foreground mask obtained from that input.

2.2 Classifier training

Training has two major steps: 1) forming positive and negative training sets and 2) feature extraction. In many applications the former step requires manual labeling which can be a costly operation. In this section we will explain how to form these samples automatically from the video sequences without using any manual supervision.

2.2.1 Forming Training Sets

Foreground masks have blobs of variable sizes. Also due to the fact that highway surveillance videos are usually captured from a distance, because of this, vehicles in the same video have similar sizes (exceptions are long buses and trucks). Hence, the area of a blob that contains a single vehicle is significantly different than the area of a blob that contains two or more vehicles. Figure 3(a) shows the blob areas seen in four different datasets of Mumbai’s highways and major roads. As the figure illustrates, it is relatively easy to determine where the blob areas show a steeper change. Therefore the area threshold \( \tau_{area} \) can be selected from the range where this change is not significant. Additionally, since this is a training phase, a “safe” threshold may be chosen, which would not capture all existing single vehicle blobs. With these observations, patches corresponding to “small” blobs are stored as positive examples. Once a positive example is located, corners and edge midpoints on the patch boundary are taken as negative example centers as illustrated in Figure 3(b). The generated negative examples are “hard” negatives which is better for the training of the classifier (illustrated in Figures 4(a) and 4(b)). With these hard negatives the classifier is forced to learn structured clutter, rather than just plain background, which is beneficial for accurate localization of vehicles in blobs that are suspected to have more than one vehicle.
2.2.2 Feature Extraction

As Figure 4 illustrates, color and shape of the patches are not discriminative enough to separate the positive examples from the negative ones. Furthermore, positive and negative examples have significantly different line distributions and orientations. For this reason a histogram of oriented gradients (HOG) based method, similar to [9] and [5], is used to extract features from the patches. First each patch is divided into four cells by a 2×2 grid. Then for each cell, an 8-bin HOG is computed. These histograms are normalized with respect to the number of pixels in each cell. And finally, these 8-bin histograms are concatenated and a classifier is trained. This method has two advantages: 1) since the training examples are automatically generated from the background subtraction, they have various sizes. This method does not require any resizing or alignment, which is very important since these operations would not be very accurate in this uncontrolled setting. 2) By using a 2×2 grid, some spatial information is also captured.

2.3 Vehicle Detection

Vehicle detection is done in two steps. First, the background subtraction explained in Section 2.1 is used to get the foreground masks. As mentioned in Section 2.2.1, blobs with relatively small areas are assumed to contain only a single vehicle. Hence, these vehicles can be detected directly by using the foreground masks. On the other hand, some blobs are suspected to have two or more vehicles, so they need to be further processed. This is done with the help of the binary classifier generated in the training step (see Section 2.2). Given a patch descriptor, this classifier decides if the corresponding patch contains a vehicle or not. In order to detect the vehicles in these suspicious blobs, a sliding window approach is used. The window size is determined by the median width and height of the positive training examples of Section 2.2. At each pixel location along the suspicious blob, a patch centered at that location is extracted. Then the sub window is used for that patch to compute and fed into the binary classifier. This process is illustrated in Figure 6. The leftmost image shows a blob which is suspected to have more than one vehicle. The middle image is the corresponding patch from the frame, and the rightmost image is the output of the pixel-wise sliding windowing binary classifier. Each pixel in this binary output indicates whether there is a vehicle centered at that pixel or not. As Figure 6 suggests, results of the binary classifier should further be processed in order to eliminate some false positives. This is handled by the constraint that the distance between two detected vehicle centers cannot be less than a particular threshold (τdist). The system is relatively more sensitive to this threshold since it directly affects the number of false positives and false negatives. However, a good threshold can still be determined using the window size and the average single vehicle blob area (see Section 2.2.1). So, for every suspicious blob, results of the binary classifier are sorted in decreasing order of areas. Then each result center is compared with all other result centers with larger areas. If the Euclidean distance is less than the threshold, this result is assumed to be a false positive and is eliminated.
2.4 Vehicle Tracking

In every frame vehicles are detected independently. Tracking in this setting is simply linking these independent detections in consecutive frames. For every vehicle, the position in the next frame is predicted using a constant acceleration dynamic model [6] as follows:

\[ \begin{align*}
    R_t &= R_{t-1} + (\Delta t)v_{t-1} \\
    v_t &= v_{t-1} + (\Delta t)a_{t-1} \\
    a_t &= a_{t-1}
\end{align*} \]

Where; vector \( P \) gives the position, vector \( v \) the velocity and vector ‘\( a \)’ the acceleration. In the next frame, all the detections that overlap with this predicted position are considered to be matching candidates. The best match is determined by the correlation coefficient \( R_{corr} \) between the detection from the previous frame \( T \) and a candidate \( I \), as in Equation 4. While computing the correlation coefficient, occlusions are not a problem since between two consecutive frames; the amount of occlusion is not expected to alter.

\[ R_{corr} = \sum_{x,y} [T(x, y)I(x, y)] \]

Where

\[ T'(x, y) = T(x, y) - \frac{\sum_{x,y} T(x,y)I(x,y)}{\sum_{x,y} I(x,y)} \]

\[ I'(x, y) = I(x, y) - \frac{\sum_{x,y} T(x,y)I(x,y)}{\sum_{x,y} T(x,y)} \]

And \( w,h \) are the width and height of detection windows.

III. EXPERIMENTS AND RESULTS

Experiments are performed on different traffic surveillance videos taken in Mumbai-North, Mumbai-South. The names of roads SV road, Link road, NH-17 and NH-8 correspond to the locations that the videos were taken. SV road is a very low-quality 320 \( \times \) 240 video with resolution. NH-8 and NH-17 are middle- and high-quality videos with 384 \( \times \) 288 resolutions each. For SV and Link roads the “region of interests” are automatically determined. For NH-17 and NH-8, the region of interest did not contain any suspicious blobs (due to viewpoint, quality and amount of clutter).

Detection directly from background subtraction, red: detection by processing suspicious blobs (Best viewed in color). For background subtraction, the MATLAB code provided by [7] is used. Haar classifier related operations are handled by using the publicly available cv2 haar library. The rest of the system is implemented in C++, using the open-source computer vision library, OpenCV [1]. A linear sub-window tree is used for Haar training. In each experiment, the system may extract hundreds of positive and negative examples by processing less than a minute of the input video. For Highways, only 600 frames (~ 20 seconds) were enough to generate the required number of examples. Classification accuracies ranged from 89% to an impressive 93% for the NH-17 dataset. Detection results are presented in Table 1 and Figure 7. From each dataset, 100 frames (with 15 frame periods to avoid including very similar frames) are manually inspected. The total number of vehicles, detections, false positives and false negatives are counted. As Table 2 shows, the proposed system has very few false positives and slightly more false negatives. Hence, extremely good detection accuracies, ranging between 87% to 94%, could be achieved in these three significantly different and challenging high-way surveillance videos. As Figure 7 shows (top 3 rows), when the video is divided into two regions, very

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frames</th>
<th>Vehicles</th>
<th>Detected Vehicles</th>
<th>Accuracy</th>
<th>False Positives</th>
<th>False Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH-8</td>
<td>150</td>
<td>258</td>
<td>234</td>
<td>90.69%</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>NH-17</td>
<td>150</td>
<td>450</td>
<td>416</td>
<td>92.44%</td>
<td>23</td>
<td>27</td>
</tr>
<tr>
<td>S.V. Rd</td>
<td>150</td>
<td>139</td>
<td>116</td>
<td>83.45%</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>Link Rd</td>
<td>150</td>
<td>231</td>
<td>204</td>
<td>88.31%</td>
<td>19</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 1. Detection Results and Accuracy evaluation

Highway Traffic Surveillance By Unsupervised Learning
accurate detection results could be achieved even though the vehicles in two regions look different. This is because we had trained one classifier for each region independent from the other region. Figure 8 shows some shortcomings of our detection algorithm. The major shortcoming is the poor detection of large vehicles. This can be explained by the following: these vehicles look very different, training is done only with average-sized vehicles and the detection window size is too small for these vehicles. Other minor shortcomings are shadow-related localization problems and some false positives which could not be eliminated by the method explained in Section 2.2.

IV. DISCUSSION

Highways and other roads (like Link Roads, S.V. Roads) of Mumbai are used for the input to the program. Mumbai is served by National Highway 3, National Highway 4, National Highway 8, National Highway 17 and National Highway 222 of India’s National Highways system. We have presented a novel approach for vehicle detection in highway surveillance videos. The main contribution is the unsupervised learning framework where the vehicles in a video are directly learned from the video without any prior knowledge or supervision. With this framework, we achieved excellent detection results in three videos that are significantly different in terms of quality, viewpoint and clutter. We are planning to extend this work in two ways: The proposed method should be tested in a real world system and the accuracy should be evaluated in different weather and day/night conditions. We expect the system accuracy to be robust to these conditions to some extent since different illumination conditions do not affect the extracted features dramatically. However, the highway lighting conditions in the night may introduce a significant challenge to the system and may require re-training of the classifiers.

As mentioned in Section 3, separating a video into multiple regions and processing these regions independently may increase the overall accuracy. This approach may be applied to complex highway surveillance videos.

REFERENCES


