Vehicle Detection Method Based on Edge Information and Local Transform Histogram

Xue Fan, Jingchun Piao, Teng Yu, Hyunchul Shin

Department of Electronics and Communication Engineering, Hanyang University, Ansan, 426791, Korea. (fanxue, kcpark, yuteng)@dslab.hanyang.ac.kr

Abstract

Robust and reliable vehicle detection from images is an important problem with applications to Advanced Driver Assistance Systems (ADAS). In this paper, we show that edge information can be used as vehicle absence cue to reject early a large portion of background windows. An improved coarse-to-fine vehicle detection method is proposed in order to achieve efficient detection with high accuracy. Furthermore, according to our study, we find that contour is the major property of a vehicle. We propose an effective method to capture the contour information, which is edge based Local Transform Histogram (LTH). We first extract the Sobel edge of the input image, and then use LTH to encode the contour information. Though linear SVM remains a popular choice for its speed and performance, we use Histogram Intersection Kernel (HIK) SVM for its better classification accuracy. Our method is evaluated on a famous benchmark dataset: UIUC car dataset. The experimental results show that our method has significantly improved the accuracy and stability of vehicle detection.

Keywords: Object Recognition, Vehicle Detection, Coarse-to-Fine Method

1. Introduction

In recent years, detecting the predefined objects in images has been an attractive problem in computer vision, e.g., face [1-3], vehicle [4-6], and pedestrian [7-10]. Vehicle detection is one of the popular areas in this field. It is an essential technology in emerging applications such as intelligent traffic surveillance, driver assistant systems, and driverless vehicles. Detecting vehicle is a challenging task owing to different viewpoints, weather conditions, color changes, and cluttered backgrounds. Significant research has been devoted to detecting, locating, and tracking vehicles in images and videos. Considerable progress has been made in vehicle detection over the past decade, which has advanced the frontiers of this problem in many aspects, e.g., features, classifiers, testing speed, and occlusion handling [11][12]. Nevertheless, there are still rooms for improvement in order to realize reliable vehicle detection in complex real world.

Among all the aspects, the performance of vehicle detection system is mainly determined by two key factors: the learning algorithm and the feature representation. In this paper we stick with the classifier which has performed well in recent evaluations [13][14]: Support Vector Machines (SVM) with linear kernel and Histogram Intersection Kernel (HIK) [8][15]. The linear SVM remains a popular choice for object detection because of its good performance and speed. Not like other kernelized SVMs, HIKSVM can be computed exactly in logarithmic time while consistently outperforming others.

Commonly used features are Histogram of Oriented Gradients (HOG) features [7], Haar-like features [3], edge and symmetry features [16][17], shadow information [18], motion information [19], Gabor features[20] and vertical/horizontal edges [21]. These features show pleasurable performance compared to others.

In this paper, we concentrate on detecting vehicles in static images and make three major contributions. First, a coarse-to-fine detection method is proposed to improve the detection efficiency and accuracy. Second, we find that contour is the major property of vehicle. So we develop an edge based Local Transform Histogram (LTH) descriptor to extract the contour information. It encodes the signs of local comparisons and has the ability to capture global contour information. Third, we propose to use the HIKSVM classifier instead of the linear SVM, which can further improve the classification accuracy about 2%.
The paper is organized as follows. In Sec. 2, a brief review of the related work is presented. The proposed vehicle detection methods are explained in Sec. 3. Experimental results are shown in Sec. 4. Finally, the conclusions and future works are summarized in Sec. 5.

2. Related work

Object detection has been widely studied and significant progress has been made in recent years. Lots of local features and descriptors are proposed for various object detection tasks. Schneiderman et al. [22] successively proposed a wavelet coefficient histogram based statistic method for multi view face and car detection. But its major drawback was the high computing expenses. Papageorgiou et al. [23] proposed a general object detection scheme using Harr wavelet and SVMs and applied it to face, pedestrian, and vehicle detection. Viola et al. [3] proposed a more popularly referred method for face detection, in which the Adaboost classifier was trained on a series of Harr wavelet coefficients and real time operation was achieved by using integral images and cascade techniques. Sun et al. [20] provided an on-road vehicle (rear views) detection using Gabor filter and support vector machines. But it relied on some other vehicle location method and only can be taken as a verification step.

Image edges are popular cues for vehicle detection, with additional extra features such as symmetry or shadow. As one of the main signatures of man-made objects, symmetry is often used for object detection, especially for vehicle detection. Images of vehicles observed from rear or frontal views are in general symmetrical. Broggi et al. [16] used edge and symmetry features to detect vehicles in a multi-resolution framework. Using shadow information as a sign pattern for vehicle detection was initially discussed in [18]. By investigating image intensity, it is found that the shadow area underneath a vehicle is distinctly darker than any other areas on asphalt road. But the intensity of shadow is easily affected by the illumination.

Local region descriptors, e.g., Histogram of Oriented Gradients (HOG) [7], capture more rich local statistical information, and are proved powerful in object detection. Another type of features, such as the CENTRIST [24], focuses on the contours of objects rather than the local regions, which can be successfully used in pedestrian detection. In the following, we show that contour is also the major property of vehicles. We propose an edge based Local Transform Histogram (LTH) descriptor to extract contour information, which achieves good detection accuracy.

As for the learning framework, many researchers have attempted to improve the performance of classic classifiers like SVM and Adaboost. It has been shown that AdaBoost does not perform well on challenging datasets with multiple viewpoints. Linear SVM remains a popular choice for object detection because of their good performance and speed. Straightforward classification using kernelized SVM requires evaluating the kernel for a test vector and each of the support vectors. They typically bring some improvement, but the time required to classify an example is intractable in practice. Wu et al. [15] shows that one can build Histogram Intersection Kernel SVM (HIKSVM) with logarithmic runtime complexity in the number of support vectors as opposed to linear for traditional kernelized approach.

3. Proposed vehicle detection methods

The flowchart of proposed vehicle detection method is shown in Figure 1. In this paper, we use coarse-to-fine vehicle detection with edge based Local Transform Histogram (LTH) and HIKSVM classifier. For the coarse detection, our purpose is to reduce the workload for fine detection. We generate an edge density factor by using the sum of binary values of edge map, which attempts to distinguish vehicles from rather uniform background, to filter out some edgeless sub-windows, while the windows containing vehicles will be reserved for fine detection. Then we extract LTH features of the windows which passed through the coarse detection. Finally, HIKSVM classifier is used to verify the windows.

The HIKSVM is a simple generalization of the linear SVM. Linear SVM is fast in evaluation and training speed, while more informative features need to be created for higher classification accuracy. Traditional kernelized SVM typically bring some improvement in accuracy, but commonly the time required to classify an example is linear in the number of support vectors. By contrast, the classification time of HIKSVM is exactly in logarithmic time.
3.1. Coarse vehicle detection with edge map

In sliding window detection approach, all the sub-windows are scanned at several scales. Since most of the sub-windows are backgrounds, evaluating these windows with complex descriptors is time consuming. Therefore we seek to use coarse detection to filter out some edgeless background windows in order to reject most of the negative windows early while preserving the positive ones. We find, by experiments, that image areas without vehicles usually are the sky, buildings or just the road, etc., where the intensity values change slowly, hence the edge density of these areas are often much smaller than that of area with objects. Therefore, we can use the edge density information of the detection windows to improve the efficiency during the coarse detection. By rejecting a large portion of simple background early, the workload can be significantly reduced and the time for fine detection can be shortened.

In order to get the edge density map, we first compute the gradients. In paper [7], Dalal et al. had reported that the centred \([1, 0, -1]\) works best when used in HOG. Thus we also use the centred 1-D point derivative \([1, 0, -1]\). The formula for computing gradient magnitude can be written as follows:

\[
\text{grad} \ (I(x, y)) = \sqrt{G_x^2 + G_y^2}
\]

where

\[
G_x = I(x + 1, y) - I(x - 1, y), \quad G_y = I(x, y + 1) - I(x, y - 1)
\]

I \((x, y)\) is the input image, \(G_x\) and \(G_y\) indicate the gradients in \(x\) and \(y\) directions respectively. Then a reasonable gradient magnitude value will be chosen as the threshold to extract binary edge map. The selection of the threshold is important to obtain useful edge maps. To decide reasonable threshold \(T_1\), we experimented with 500 positive windows and negative windows from UIUC training dataset. After computing training data's mean gradients, we obtained the frequency distribution, as shown in Figure 2.

![Figure 2. Histogram of mean gradients in positive and negative windows](image-url)
In Figure 2 the x axis is the value of mean gradients and the y axis is the frequency of each mean gradient bin. Our calculation results show that nearly 50% of the negative windows’ mean gradient magnitudes are smaller than 60 while positive windows’ mean gradient magnitudes are mainly distributed between 60 and 80. Thus, we experiment with different gradient magnitudes ranging from 50 to 70 in order to choose an effective threshold $T_1$. The edge map will be constructed by using the selected threshold $T_1$ as shown in the following formula:

$$d_{i,j} = \begin{cases} 
1 & \text{grad} \geq T_1 \\
0 & \text{grad} < T_1
\end{cases}$$

(3)

where $T_1$ is the threshold used for obtaining a binary image. We compute the sum of the binary values for each window in training data. All these sum values are used to construct the distribution histogram. Figure 3 shows the frequency of sum in binary image when $T_1$ is 55. According to the distribution information from Figure 3, we select a suitable region $T_2$ to reject a large portion of negative windows while preserving nearly all the positive windows. If the sum of binary values of the detection window is outside the threshold region we selected, it will be rejected. Otherwise it will be preserved for fine detection.

![Figure 3. Histogram of binary value sums in positive and negative windows](image)

3.2. Local Transform Histogram (LTH) descriptor

Through carefully designed experiments, Wu et al. [24] proved that contour is the most important information for human detection and signs of comparisons among neighboring pixels are the key information to capture contours. They developed the CENTRIST feature for pedestrian detection, which achieved good detection accuracy. Compared to pedestrians, vehicles are artificial rigid objects with fixed structural characteristics. We think contour should also be an informative cue for vehicle detection. After directly implementing CENTRIST on UIUC car dataset, we find that the experimental results are satisfactory. It achieved 85.0% EER (Recall-Precision Equal Error Rate) for UIUC-single and 82.5% EER for UIUC-Multi, which proved that contour is the major property of vehicle.

The critical issue is how to capture the contour. In order to effectively encode the contour information, we propose an edge based Local Transform Histogram (LTH) descriptor. Instead of using intensity value of pixels to compute the descriptor, we directly use the edge information. Given an input image, we first perform sobel edge detection to generate binary image. Then we compute LTH feature for every detection window.

3.2.1. Compute Local Transform (LT) values

After we get sobel edge of the input image, we need to compute the local transform values for every pixel. Local Transform (LT) compares the binary value of a pixel with the values of its eight neighboring pixels. If the centered pixel’s binary value is bigger than one of its neighbors’, a bit 1 is set in the corresponding location. Otherwise a bit 0 is set. Figure 4 is an illustration of Local Transform structure. The LT value for the centered pixel is computed by collecting the eight bits from left to right and top to bottom and then converting them to base-10 values. If the binary value of a pixel is 1, we will compute its corresponding LT value. Otherwise, the LT value of the pixel will be set 0.
3.2.2. Build LTH visual descriptor

After getting the LT values for every pixel, we need to extract LTH descriptor for every detection window. Each detection window is divided into cells and any adjacent 2×2 cells are integrated into a block in a sliding fashion with 50% overlap, as shown in Figure 5. The LTH descriptor is extracted from each block. Then we collect descriptors over the detection window to form the final descriptor. In UIUC car dataset, the size of detection windows we used is 40×100. We divide the detection window into cells of 10×10 pixels. Each block is represented by a 256-D feature vector. There are 3×9=27 blocks, thus the feature vector for any detection windows has 256×27=6912 dimensions. These features are then used to train a HIKSVM classifier.

3.3. HIKSVM classifier

Discriminative classification using Support Vector Machines (SVM) and variants of boosted decision trees are two of the leading techniques used in vision tasks ranging from detection of objects in images like faces [25], pedestrians [26], cars, and multi-category object recognition [27][28]. Linear SVM remain a popular choice for human detection because of their good performance and speed, while more informative features need to be created for higher detection accuracy. Part of the appeal for SVM is that non-linear decision boundaries can be learnt using the so called kernel trick. Non-linear kernels, like RBF kernel typically bring some improvement, while the training speed is medium and the evaluation is slow. The time required to classify an example is linear in the number of support vectors, which is intractable in practice.

Recently, the Histogram Intersection Kernel (HIK) has attracted a lot of attention in the computer vision community. Histograms are used in almost every aspect of computer vision, from visual descriptors to image representations. HIK, as a measure for comparing the similarity (or dissimilarity) of two histograms, achieves better performances in various machine learning tasks than other commonly used measures. Wu et al. [15] shows that one can build histogram intersection kernel SVM (HIKSVM) with run time complexity of the classifier logarithmic in the number of support vectors as opposed to linear for the kernelized SVM. In our experiments we typically have SVM with a few thousand support vectors. The following equations show the decision function of histogram intersection kernel.

\[ h(x) = \sum_{i=1}^{n} \alpha_i K_{HI}(x, x^i) \]  (4)

\[ K_{HI}(x, x^i) = \sum_{i=1}^{m} \min(x_i, x_i^i) \]  (5)
where \( h(x) \) is the classification score, \( K_{\mathcal{H}} \) is the histogram intersection kernel, \( n \) denotes the number of support vectors, \( m \) denotes the dimension of support vectors, \( \alpha_j \) is the coefficient corresponding to the \( j \)-th support vector, \( \cdot \) represents the evaluated feature vector, and \( x_j \) represents the feature of the \( j \)-th support vector.

4. Experimental results

We evaluated the performance of our vehicle detection method on the UIUC car dataset, which contains images of side views cars. The test data are split into the set of 170 images of approximately same scale (UIUC-Single) and the set of 108 images at multiple scales (UIUC-Multi).

4.1. Detection speed

Our first contribution is the coarse-to-fine detection, which can reject a large portion of simple background early in coarse detection while preserving almost all positive ones, to significantly reduce the workload for fine detection and improve the detection speed. We first perform coarse detection on the training dataset in order to find suitable thresholds and then use the thresholds to do coarse detection on the corresponding test dataset. The threshold \( T_1 \) we used to get the binary edge map is 55, and the region \( T_2 \) used for rejecting negative windows is \([1000, 2700]\). The experimental results are shown in Table 1.

### Table 1. The performance of coarse-to-fine detection method

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Average time without using coarse detection</th>
<th>Average time using coarse detection</th>
<th>Average CPU time reduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIUC-Single</td>
<td>11.7 seconds</td>
<td>8.3 seconds</td>
<td>29.3%</td>
</tr>
<tr>
<td>UIUC-Multi</td>
<td>294.6 seconds</td>
<td>174.9 seconds</td>
<td>38.7%</td>
</tr>
</tbody>
</table>

4.2. Detection accuracy on UIUC car dataset

The second contribution is the edge based LTH descriptor. Table 2 shows the comparison of different methods on the UIUC single-scale cars and UIUC multi-scale cars at recall-precision equal error rate (EER). The run time complexity of HIKSVM is logarithmic in the number of support vectors as opposed to linear for the traditional kernelized SVM. Compared to linear SVM, HIKSVM is a little slower in training and evaluation but achieves higher detection accuracy. From the following experimental results you can see the performance of different methods.

### Table 2. Performance of different methods on UIUC car dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>UIUC-Single</th>
<th>UIUC-Multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse Representation [29]</td>
<td>79.0%</td>
<td>-</td>
</tr>
<tr>
<td>Boundary Shape Model [30]</td>
<td>85.0%</td>
<td>-</td>
</tr>
<tr>
<td>CENTRISI[24]</td>
<td>85.0%</td>
<td>82.5%</td>
</tr>
<tr>
<td>Our Method (Linear SVM)</td>
<td>89.5%</td>
<td>86.0%</td>
</tr>
<tr>
<td>Our Method (HIKSVM)</td>
<td>91.5%</td>
<td>87.8%</td>
</tr>
</tbody>
</table>

**Figure 5.** Sample detection results for our method on the UIUC-Single (top) and UIUC-Multi (bottom) dataset.
5. Conclusions and future works

In this paper, we described a novel coarse-to-fine vehicle detection method. The proposed coarse detection can reject a large portion of negative windows while preserving almost all the positive windows. By using coarse detection, we can reduce detection time by 29.3% for UIUC-Single and 38.7% for UIUC-Multi. Applying this coarse-to-fine method, LTH visual descriptor and HIKSVM for vehicle detection, we achieved an impressive 91.5% EER for UIUC-Single and 87.8% EER for UIUC-Multi. We believe that our work can make a contribution to the state of the art in vehicle detection.

In the future, we plan to explore ways of combining multiple features, such as self-similarity, color information and motion information, instead of using only single feature. On a more conceptual level, we will look into ways of detecting partial occluded vehicles, which is a weakness of our current detector.

6. Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MOE) (No. NRF-2013R1A1A2004421) and Ministry of Trade, Industry & Energy (MOTIE) through IDEC Platform center (IPC) at Hanyang University.

7. References

Vehicle Detection Method Based on Edge Information and Local Transform Histogram
Xue Fan, Jingchun Piao, Teng Yu, Hyunchul Shin


