Fast Car Detection Using Image Strip Features

Wei Zheng[1, 2, 3], Luhong Liang[1,2]

[1]Key Lab of Intelligent Information Processing, Chinese Academy of Sciences (CAS), Beijing, 100190, China
[2]Institute of Computing Technology, CAS, Beijing, 100190, China
[3]Graduate School of the Chinese Academy of Sciences, Beijing, 100039, China
{wzheng, lhliang}@jdl.ac.cn

Abstract

This paper presents a fast method for detecting multi-view cars in real-world scenes. Cars are artificial objects with various appearance changes, but they have relatively consistent characteristics in structure that consist of some basic local elements. Inspired by this, we propose a novel set of image strip features to describe the appearances of those elements. The new features represent various types of lines and arcs with edge-like and ridge-like strip patterns, which significantly enrich the simple features such as haar-like features and edgelet features. They can also be calculated efficiently using the integral image. Moreover, we develop a new complexity-aware criterion for RealBoost algorithm to balance the discriminative capability and efficiency of the selected features. The experimental results on widely used single view and multi-view car datasets show that our approach is fast and has good performance.

1. Introduction

Car detection is an indispensable technology in emerging applications such as intelligent traffic surveillance, driver assistant systems and driverless vehicles. Recently, as one of the fundamental problems in computer vision and pattern recognition, car detection, like other related topics such as face detection and pedestrian detection, attracts more and more attentions of researchers worldwide. Different from faces and human bodies, cars are artificial rigid objects with obvious and consistent characteristics in structure such as wheels, bumpers and pillars, which provide crucial cues for car detection. In this paper, we address the car detection problem in static images and focus on local features that describe these structural characteristics in particular.

There has been extensive literature on object detection. Recently, more and more research works on object detection cover the car detection problem. Various models and methods have been proposed, including the Implicit Shape Model (ISM) [8, 14], the deformable part model [6], the Biologically Inspired Model (BIM) [11, 20, 25], the synthetic 3D models based on 3D feature maps [15], the wavelet based method [21, 24], the random fields based method [12, 31], the Cluster Boosted Tree (CBT) [34], etc. These approaches achieve good performance in single view, multi-view and/or partially occluded car detection experiments. Some approaches directly focus on the car detection problem, such as the sparse part-based representation [1], the Gabor filter and SVM based method [28] and the two-level hierarchical SVM [35], and have made considerable progress. As shown in the literature, local features [16, 22, 30, 33] and descriptors [2, 3, 17, 29] usually play important roles in detecting objects including cars. However, to our best knowledge, most of these local features and descriptors are originally designed for the detection of faces, pedestrians or general objects. There are few approaches that specially consider the structural characteristics of cars.

Unlike faces and human bodies, cars have relatively consistent characteristics in structure such as four wheels, a certain number of pillars, two bumpers, etc. Although the appearances of these parts have various changes due to different car models, view points and lighting conditions, they consist of some basic geometric elements such as lines and arcs with edge-like and ridge-like strip patterns. Inspired by the observations above, we design a new set of *image strip features* that explicitly describe the appearance of the structural characteristics of cars. These features can be extracted based on the integral images. We also propose a *complexity-aware* criterion for the boosting framework in order to balance the discriminative capability and efficiency of the selected features. The experimental results on UIUC [1] and PASCAL 2006 challenge [5] car datasets show that our approach outperforms the methods based on edgelet and haar-like features.

1.1. Related works

There have been lots of local features and descriptors proposed for various object detection tasks. In some earlier works, wavelet based features [21, 24] are proposed for object detection. Haar-like features [30] and the extended harr-like features such as [16] are successfully applied in face detection. These haar-like features describe the information of edges and ridges in multi-scale, and can be

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

fast calculated via the integral image. However, the feature itself can only capture simple and regular-shaped patterns such as a segment of edge and ridge in horizontal, vertical or specific directions. Local region descriptors, e.g., Histogram of Oriented Gradients (HOG) [3] and covariance descriptor [29], capture more rich local statistical information, and are proved powerful in object detection. Another kind of features [26, 27, 33] focuses on the contours and edges of objects rather than the local regions. For example, the edgelet feature [33] extracts the gradient information using a single-pixel wide template. These contour and edge based features are successfully used in pedestrian detection. However, it seems that they are sensitive to scaling, shifting and rotation variations. Besides the simple features above, researchers also propose combinations of simple features for object detection, such as joint features [18, 22] and heterogeneous features [4, 32]. To our best knowledge, most of these local features are originally designed for the detection of faces, pedestrians or general objects. Although some of these features achieve good performance in car detection, they do not explicitly describe the structural characteristics of cars, which are crucial cues in car detection.

In object detection, a lot of approaches are based on classifiers and sliding window strategy. SVMs [9, 19, 21, 28] and boosting algorithms [10, 30, 32, 34] are widely used in training the classifiers. In order to deal with multi-view object detection, some tree structured classifiers such as vector boosted tree [10] and CBT algorithm [34] are proposed. In some recent works such as [4, 32, 36], the computational complexity of weak classifiers is considered in the boosting algorithms. There are also some approaches beyond the sliding window framework, e.g., the efficiently subwindow search scheme [13], the generalized Hough transform [14] and the BIM based methods [11, 20, 25]. Most of these approaches focus on general object detection problems, and some of the approaches use the car detection as an additional experiment.

1.2. Overview of our approach

In this paper, we propose a novel set of image strip features for car detection. An image strip feature can be considered as a template of a curve segment with a certain strip pattern, which is described by several back-to-back regions as shown in Fig. 1. The feature response reflects the contrast of these image regions. We also develop an approximate algorithm based on the integral image method. Given a window, a full set of the image strip features with different curve segments, strip patterns and positions can be built.

We employ the RealBoost algorithm to train the cascade classifiers and the CBT classifiers based on the image strip features. In order to speedup the detection process, a new complexity-aware criterion is proposed to balance the discriminative capability and efficiency of the selected features. In detection, we follow the sliding window strategy and employ the mean-shift clustering algorithm to merge the positive responses of the classifier and obtain the bounding boxes of the objects.

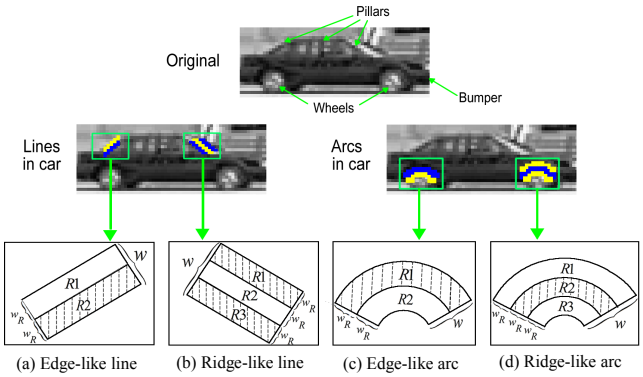


Fig. 1. Image strip features vs. car structure.

The major contributions of this paper are: (1) a novel set of image strip features specially designed for car detection; (2) an integral image based algorithm to approximately calculate the image strip features; (3) a new complexity-aware criterion in boosting framework to balance the discriminative capability and efficiency of the selected features.

The rest of this paper is organized as follows. Section 2 presents the image strip features. Section 3 introduces the complexity-aware criterion in boosting framework. Section 4 shows our experimental results. Conclusion and discussion are given in the last section.

2. Image Strip Features

2.1. Overview of image strip features

Most of cars have relatively consistent characteristics in structure. Intuitively, if we look into the car structure (See Fig. 1), we can find that all of the cars have common structural components such as wheels, pillars, bumpers, etc. Although the appearances of these components may look much different due to variations of car models, view points and lighting conditions, they consist of some basic geometric elements such as lines and arcs with edge-like and ridge-like strip patterns. For example, as shown in Fig. 1, the appearance of a tire can be represented by several arcs with the length of 6~9 pixels and ridge-like pattern. These features are important cues for discriminating cars and non-car objects. Inspired by the observations above, we propose a new set of image strip features to describe these multi-scale geometric elements in car structures.

An image strip feature can be formally represented by a triple $S=<c_L, t_w, p>$, where c_L represents the curve segment of length L, t_w represents the strip pattern of width w, and p represents the position of the feature within a given window. All the features with valid c_L, t_w, and p form an exhaustive set $\{S\}$. For car detection, we only consider the line segments and arcs, and limit the strip patterns to edge-like and ridge-like.

As shown in Fig. 1a and 1c, an edge-like feature can be described by two back-to-back *single strip regions* with the same curve pattern and width w_R, while a ridge-like feature consists of three single strip regions. The responses of the features can be calculated via the mean intensities of the single strip regions as (1) and (2)

$$f_{edge} = \left| \frac{\sum_{(x,y)\in R1} I(x,y)}{\|R1\|} - \frac{\sum_{(x,y)\in R2} I(x,y)}{\|R2\|} \right|, \quad (1)$$

$$f_{ridge} = \frac{1}{2} \left| \frac{\sum_{(x,y)\in R1} I(x,y)}{\|R1\|} + \frac{\sum_{(x,y)\in R3} I(x,y)}{\|R3\|} - \frac{2\sum_{(x,y)\in R2} I(x,y)}{\|R2\|} \right|, \quad (2)$$

where $I(x,y)$ is the pixel intensity, $\|\bullet\|$ represents the total number of pixels in a particular region, and $R1$, $R2$ and $R3$ are the single strip regions as shown in Fig. 1.

Particularly, we use the absolute value in (1) and (2), because the contrast pattern of cars is more complex than that of objects like faces. For example, eyes are always darker than skin of cheeks, but there is no such simple regularity between cars and background. For example, the white cars may be brighter than the background, while the black cars may be darker. Using the absolute values as the feature responses, the image strip features just describe the contrast information, which are mostly invariant to the colour of cars.

According to the observation of the car structure, we specifically set the curve pattern c_L to lines, 1/8 circles, 1/4 circles and 1/2 circles, which are similar to edgelet [33]. The single strip regions in one image strip feature have the same width, i.e. $w = 2w_R$ (edge-like) or $w = 3w_R$ (ridge-like), and limits the curve length l to 4~12 pixels and the single strip region width w_R to 2~6 pixels in a sliding window of 64×32 pixels.

The proposed image strip features remind us of haar-like features [30]. However, the image strip features are not just simple extensions of haar-like features. In some sense, the image strip features can be considered as joint haar-like features constrained by some curve patterns. The curve patterns can be designed to represent higher level semantic information of the essential characteristics of objects with relative consistent structures. Compared with edgelet features, the image strip features are based on the statistics of regions rather than the gradients along a line or arc. This makes the image strip features more robust to slight variation of scaling, shifting and rotation.

2.2. Fast feature extraction

In order to generate an image strip feature, we need to specify the curve pattern, the strip pattern and the relative position, i.e. c_L, t_w, and p. We employ edgelet [33] to enumerate the c_L and p, and then dilate the edgelet features along the normal directions to form the edge-like and ridge-like strip patterns as shown in Fig. 2. Obviously, settings different t_w, one edgelet can generate multiple image strip features. A direct way of extracting the image strip features is calculating the response via (1) and (2) by points, namely the *D-Strip* approach. Besides the direct method, we also propose an integral image based method to extract the image strip features, namely the *I-Strip* approach.

D-Strip approach:

We can directly calculate the mean intensities of each single strip regions point-by-point. The coordinates of pixels within each single strip region can be easily obtained by flood fill or similar algorithms, and then stored in a list. So that one edge-like feature has 2 lists and one ridge-like feature has 3 lists. For feature extraction, the response of the feature can be calculated via (1) or (2) using the lists as lookup tables. However, this point-by-point method needs a lot of memory for storing the lists and a lot of time for calculating the mean intensities of the single strip regions, especially when the single strip regions are large.

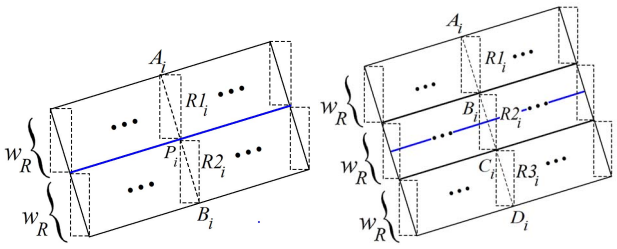
I-Strip approach:

In order to reduce the memory and computation cost, we propose an approximate algorithm based on the integral image. Apparently, when the curve patterns are horizontal and vertical lines, the image strip features degrade to haar-like features and can be directly extracted using the approach in [30]. Therefore, we only focus on the complex features with oblique line and arc patterns.

Fig. 2a illustrates our approach for an edge-like feature with the line pattern. We employ two series of small upright rectangles to represent the upper and lower single strip regions respectively. More specifically, we assign two *associated rectangles* shown as $R1_i$ and $R2_i$ in Fig. 2a for each point P_i along the edgelet. $R1_i$ is an upright rectangle which is determined by the point P_i as one vertex and the point A_i as the diagonal vertex, where A_i is the intersection of the normal at P_i and the upper boundary of the single strip region. $R2_i$ and other associated rectangles can be determined in a similar way. Then, the response of the image strip feature in Fig. 2a can be calculated as (3)

$$f_{edge} = \left| \frac{\sum_{i=1}^{L} g(R1_i)}{\sum_{i=1}^{L} \|R1_i\|} - \frac{\sum_{i=1}^{L} g(R2_i)}{\sum_{i=1}^{L} \|R2_i\|} \right|, \quad (3)$$

where $g(\bullet)$ is the function to sum up the intensities of all the points in a particular region via the integral image [30], and L is the length of the edgelet feature.



(a) Edge-like feature (b) Ridge-like feature

Fig. 2. Line features approximated by associate rectangles.

Fig. 2b illustrates a ridge-like feature with the line pattern. Like the edge-like features, each point on edgelet specifies four points A_i, B_i, C_i and D_i, which decide the three associated rectangles $R1_i$, $R2_i$ and $R3_i$. With the series of associated rectangles, the response of this ridge-like feature can be calculated as (4)

$$f_{ridge} = \frac{1}{2}\left| \frac{\sum_{i=1}^{L} g(R1_i)}{\sum_{i=1}^{L}\|R1_i\|} + \frac{\sum_{i=1}^{L} g(R3_i)}{\sum_{i=1}^{L}\|R3_i\|} - \frac{2\sum_{i=1}^{L} g(R2_i)}{\sum_{i=1}^{L}\|R2_i\|} \right|. \quad (4)$$

Furthermore, the image strip features with arc patterns can be approximately calculated through the associated rectangles in the same way.

Here, we give a brief analysis on the computation costs of the D-Strip and the I-Strip algorithms. Given L and w_R, and suppose $g(\bullet)$ consumes 3 additions, according to (1)~(4), we can list the computation costs in Table 1, where C is the sum of the addition and multiplication operations[1]. As shown in Table 1, the computation costs of haar-like features are constant. The computation costs of the I-Strip features are irrelevant to w_R. It can be seen that it is "cheaper" to calculate via I-Strip when w_R is large, and it is "cheaper" to calculate via D-Strip when w_R is small.

3. Complexity-Aware RealBoost

As illustrated in Table 1, the computation costs of different I-Strip features are greatly different. For example, the haar-like feature of edge-like pattern consumes 9 operations, while the ridge-like feature with l=12 consumes 148 operations. If we use the "cheaper" features in the earlier levels of the cascade classifiers, we can speed up the detection process. In order to obtain an effective and efficient classifier, we propose a *complexity-aware* criterion to balance the discriminative capability and the computation complexity. Some related works have been done in [4, 32, 36]. In this paper, we introduce a complexity- aware criterion for boosting framework

$$Z' = Z + aT, \quad (5)$$

where Z is the *discriminative criterion*, i.e. the measurement of the discriminative capability of the weak classifiers, T is the *complexity criterion* for minimizing the expectation of the total execution time in detection, a is the *complexity-aware factor* to balance the discriminative capability and the computation complexity. In the RealBoost framework, the discriminative capability is measured by Bhattacharyya coefficient between the distributions of the object and non-object classes [23]

$$Z = 2\sum_j \sqrt{W_+^j W_-^j}, \quad (6)$$

where W_+^j / W_-^j is the probability distribution of the feature value for positive/negative samples.

Suppose there are M weak classifiers with computation cost C_i. Denote the total execution time consumed by the positive and negative windows T_{pos} and T_{neg} respectively, and the number of the true positive and the false positive windows through the i^{th} level by N_i^+ and N_i^- respectively. The expectation of the total execution time of detection is

$$E(t) = T_{pos} + T_{neg} = \sum_{i=1}^{M} N_i^+ C_i + \sum_{i=1}^{M} N_i^- C_i. \quad (7)$$

Since the total number of false positive windows is much larger than the true positive windows, $E(t)$ can be approximated by T_{neg}

$$E(t) \approx T_{neg} = \sum_{i=1}^{M} N_i^- C_i = \sum_{i=1}^{M} (N\ fp_{i-1}) C_i = N \sum_{i=1}^{M} fp_{i-1} C_i, \quad (8)$$

where N is the total number of the sliding windows , fp_i is the false positive rate of the i^{th} level. In order to minimize $E(t)$, we select the i^{th} feature with the minimum T value

$$T = fp_{i-1} C, \quad (9)$$

where C is the computation cost of the features as shown in Table 1. Substitute (6) and (9) into (5), we get the final complexity-aware criterion of the i^{th} level for the RealBoost framework

$$Z_i' = 2\sum_j \sqrt{W_+^j W_-^j} + a\ fp_{i-1}\ C. \quad (10)$$

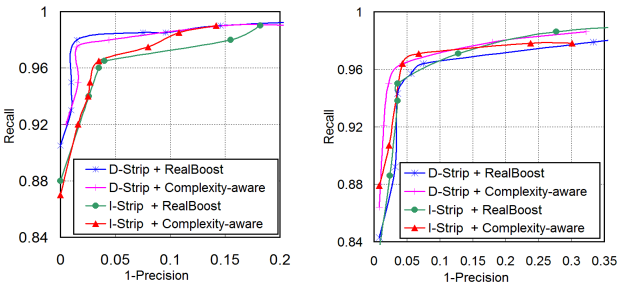
Intuitively, we can explain (10) as below. When the false positive rate is large, i.e. most of the windows are processed by the earlier levels of the cascade classifier, we tend to select the features with cheaper computation cost. When the false positive rate becomes smaller, i.e. only a few windows are processed by the later levels of the cascade classifier, the algorithm adaptively makes the computation cost less important.

TABLE 1: The computation costs of D-Strip and I-Strip.

	I-Strip				D-Strip[2]	
	Haar-like		Complex features		Edge	Ridge
	Edge	Ridge	Edge	Ridge		
Add.	7	11	8L-1	12L-1	$2Lw_R$ +1	$3Lw_R$ +2
Mul.	2	5	2	5	2	5
C	9	16	8L+1	12L+4	$2Lw_R$ +3	$3Lw_R$ +7

[1] It is reasonable to consider the addition and multiplication as same operations in analyzing the complexity, since the throughput and latency of these operations in P4 CPU are similar.

[2] Since it is hard to accurately calculate the point number of an irregular region, we list the reasonable estimations in the table.



(a) UIUC single scale (b) UIUC multi-scale

Fig. 3. Evaluation on UIUC dataset.

4. Experiments

We evaluate the performance of our approach on the widely used datasets of the single view and multi-view car images in real-world scenes. Besides the end-to-end comparisons with the state-of-the-art approaches, evaluations of the individual modules show more insights into the proposed method.

4.1. Experiments on single view car detection

In this section, we focus on the single view car detection task. The UIUC side view car dataset [1] is used in the experiment. The dataset contains a single scale test set (170 images with 200 cars), a multi-scale test set (108 images with 139 cars), and a training set of 550 side view car images. The car patches from the training images are resized to 64×32 pixels and horizontally flipped. So that there are totally 1,100 car patches in the positive training set. We also collect 10,000 images without any car on the Internet as the negative training set. A reduced set of 32,862 edgelet features (about 1/30 of a full set) are used for generating the image strip features, so that the size of the image strip feature set is limited to be acceptable to our experimental environment. The total number of image strip features (I-Strip or D-Strip) is 198,174 as listed in Table 3. In the training, we set the complexity-aware factor a 0.05.

In testing, we follow the criterion described in [5] to distinguish the correct detections from the false positives. Fig. 3 shows the precision-recall curves. We can see that the D-Strip and the I-Strip have similar performance. Furthermore, the results of the complexity-aware criterion based algorithm are very close to that of the RealBoost algorithm, i.e. the complexity-aware criterion does not reduce the accuracy of the system apparently.

We compare our approach with previous approaches following the Equal Precision and Recall rate (EPR) method. The results are listed in Table 2. It can be seen that the image strip features have high performance competitive to other state-of-the-art methods on both single scale and multi-scale test sets. Fig. 8a shows some results of the I-Strip and complexity-aware criterion based system.

TABLE 2: EPR rates of different methods on UIUC dataset.

Method	Single scale	Multi-scale
Leibe *et al*. [14]	97.5%	95%
Fergus et al. [7]	~86.5%	-
Mutch & Lowe [20]	99.94%	90.6%
Wu et al. [34]	97.5%	93.5%
Fritz et al. [8]	88.6%	87.8%
Zhu et al. [35]	~81.0%	-
Lampert et al. [13]	98.5%	98.6%
D-Strip + RealBoost	98.0%	95.0%
D-Strip + Complexity-aware	98.0%	96.0%
I-Strip + RealBoost	96.3%	95.7%
I-Strip + Complexity-aware	96.5%	96.0%

4.2. Experiments on multi-view car detection

In this section, we evaluate the image strip features on the multi-view car detection task using two experiments.

For the first experiment, we manually label 4,000 car samples in MIT street scene dataset [38] and PASCAL 2006 challenge training and validation set [5]. The car patches are cropped and normalized to 64×32 pixels. The negative image set is the same as that used in section 4.1. Since the inner class variation of multi-view objects detections is large, most of previous method adopt "divide and conquer strategy" to design multi-channel or tree structured classifiers [10, 34]. We also follow this strategy and train a CBT classifier in the experiment.

For comparison, we use an image collection similar to [32] in the PASCAL 2006 challenge test sets. There are 390 images with 491 multi-view cars larger than 64×32 pixels. According to [32], the test set is further divided into the close shot images (with cars of 250 to 500 pixels high) and the long-distant shot images (with cars of 32 to 250 pixels high). We also implement the haar-like features and edgelet features. Since the pattern of haar-like features is much simpler than that of edgelet and the image strip features, a much more dense sampling strategy is used in feature generation. As for the edgelet features, we use the set of 32,862 features, which is the same subset used for generating the image strip features (See Section 4.1). Table 3 lists the size of the feature sets used in the training.

Fig. 4 shows the precision-recall curves. The result of heterogeneous features reported in [32] is also drawn in the chart. The proposed approach obviously outperforms the haar-like feature and edgelet, and performs very close to the heterogeneous features [32]. The image strip features reflect more semantic information on the structural

TABLE 3: Feature sets used in CBT training.

Feature Type	Number of features
Haar-like feature	70,008
Edgelet	32,862
Image strip feature	198,174

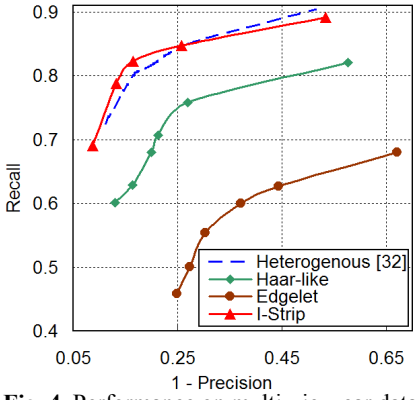


Fig. 4. Performance on multi-view car dataset.

elements of cars than the haar-like features. Compared with edgelet features, our method obviously outperforms edgelet features on the multi-view car test set. The reason is that the image strip features are based on local statistics rather than gradients of pixels along a curve, so that they are more robust to noise and variations. This is perhaps important to the detection of the multi-view objects. Our method just achieves performance comparable to the heterogeneous features [32], but as a kind of simple and fast features the performance can be improved by incorporating with more powerful features.

We also evaluate the I-Strip features on the overall PASCAL 2006 challenge dataset [5]. We use the training and validation set in [5] for training, including 490 "acceptable" car patches as original positive samples and 2,618 images as negative images in which cars are cropped away. Our approach consists of two detectors (a lateral-view detector and a rear-front-view detector), which are trained by RealBoost algorithm respectively. The lateral-view detector is trained using car samples of 64×32 pixels, and the rear-front-view detector is trained using car samples of 32×32 pixels. Our approach achieves 44.3% EPR rate. The performance is promising to be further improved by using more original car samples for training. The highest EPR rates reported in the PASCAL 2006 and 2007 challenges are about 45% and 55% respectively [37]. Considering that there are 233 truncated cars among the 854 cars in the test set and our approach is not specifically designed for detecting partially occluded cars, the proposed approach is comparable to the state-of-the-art methods. Some detection results are shown in Fig. 8b.

4.3. Image strip features vs. car structure

In this section, we further investigate the relations between the image strip features and the car structure.

We train a cascade classifier using a set of strictly aligned side-view car patches and investigate the selected image strip features. 150 side view car images from UIUC training set are used. All the cars in the images are towards left and aligned using the tangent points of the tires and the ground. We use RealBoost algorithm to select the I-Strip features from the feature set described in Section 4.1.

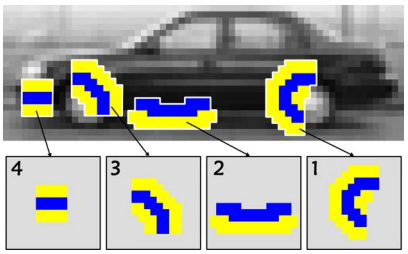
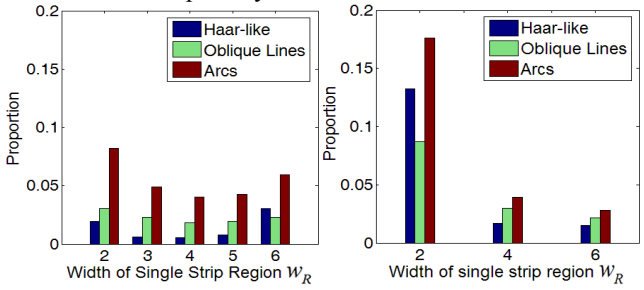


Fig. 5. Selected image strip features vs. car structures.

Fig. 5 visualizes the first selected 4 image strip features. It is interesting that all the 4 features reflect some car structural elements. Feature 1 and 3 are perfectly on the tires; feature 2 describes the contrast of the chassis and the ground; feature 4 reflects the ridge-like pattern of the front bumper. This shows that the image strip features can reflect the structural characteristics of cars. Of course, the experiment above demonstrates the ideal situation, in which the pattern of the aligned cars is very compact. In practical detection tasks, the relations between the features and the car structure may not be so intuitive, because the car samples always have larger inner class variations.

We also investigate the image strip features selected in the experiment on the multi-view car detection in Section 4.2. We collect the selected features by the CBT algorithm and count the numbers of different types. As shown in Fig. 6a, a large portion of the selected features are the oblique lines and arcs. It shows that the complex image strip features beyond haar-like ones play very important roles in discriminating cars and non-car objects. Another interesting phenomenon is that the ridges with $w_R = 2$ tend to be selected as shown in Fig. 6b. If we look into the car samples in Fig. 5 or resize the cars in Fig. 8 to 64×32 pixels, we can find that there are a lot of structural elements of cars, such as tires and pillars, appear as line or arc ridges with $w_R = 2$. This can intuitively explain the phenomenon shown in Fig. 6b. It can be seen that our image strip features can explicitly describe the structural characteristics of cars, which is perhaps important for the features with strong discriminative capability.



(a) Selected edge-like features. (b) Selected ridge-like features.

Fig. 6. Statistics of the selected image strip features.

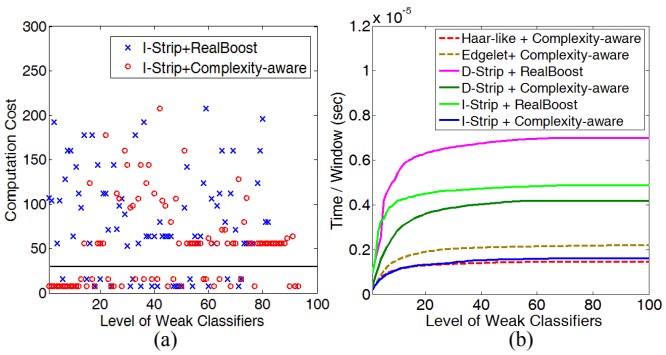


Fig. 7. (a) Computation costs of the I-Strip based classifiers. (b) Speed of the different classifiers (y-coordinate is the average execution time per window of the first *n* weak classifiers).

4.4. Experiments on complexity-aware RealBoost

We evaluate the proposed complexity-aware criterion based algorithm using the UIUC single-scale car dataset described in Section 4.1. Haar-like features and edgelet features are used for comparison. All the approaches are implemented in Visual C++ 6.0 environment without any code optimization and tested on a desktop with Intel P4 2.8GHz CPU and 1GB memory. The detector performs the exhaustive search with the 1-pixel step and the scaling factor of 0.9 on images. In this case, the detector processes overall 6,908,035 sliding windows on the 170 images.

We firstly examine the feature selection in training. We visualize the selected features in Fig. 7a, in which all the haar-like features are under the black line[3]. It shows that the first 13 features selected by the complexity-aware criterion based algorithm are all haar-like features. There are also some haar-like features selected by the RealBoost algorithm, however, most of them appear in the later levels. The results show that our complex-aware RealBoost tends to select simple features in the first several levels that dominate the total execution time. Considering the results presented in Section 4.1, we can see that our approach is more efficient than the RealBoost algorithm, while preserving the accuracy.

Fig. 7b shows the average execution time of the first *n* levels of each cascade classifier. It can be seen that the D-Strip and traditional RealBoost based approach is slower than the other fast approaches, because some features with large w_R appear in the first several levels of the cascade classifier and affect the speed. The I-Strip and complexity-aware based approach is as efficient as the haar-like based one, and is about 30% faster than the edgelet based one. The total execution time of all the 170 test images is 18 seconds (about 0.1 second per image).

[3] Each feature is represented by its computation cost, i.e. the parameter C described in Section 3. Since haar-like features have $C \leqslant 16$ and the complex image strip features have $C > 16$, we can easily distinguish them by the black line.

5. Conclusion and Discussion

In this paper, we proposed a novel set of image strip features for car detection. An integral image based feature extraction and a complexity-aware criterion for RealBoost framework have been developed to make the image strip feature based approach more efficient while preserving the accuracy. Experimental results have shown that our method is fast and has good performance.

The image strip features represent the semantic information of higher level comparing to haar-like features. In some sense, the I-Strip features are similar to a subset of the joint haar-like features constrained by some curve patterns. Therefore, the proposed features have stronger discriminative capability than haar-like features. Furthermore, by designing different curve patterns, the image strip features can be tuned to describe various structural characteristics of different objects. Since the image strip features are based on the statistics of regions, they are robust to slight variation of scaling, shifting and rotation. Of course, compared with the complex local descriptors such as HOG [3] and covariance descriptor [29], the image strip features discard some statistical information, which weakens the discriminative capability. However, it seems the unavoidable cost of fast features.

To date the image strip features are just designed for car detection, but they can be easily extended to other object detection tasks. By incorporating with stronger features like in [4, 32], the performance can be further improved. The image strip features are also promising in designing part-based object detection system for occlusion problem. We will address the research on these topics in the future.

Acknowledgements

This paper is partially supported by NSFC under contracts Nos.60772071, 60833013, 60832004, 60872124; National Basic Research Program of China (973 Program) under contract 2009CB320902; and Hi-Tech Research and Development Program of China under contract No.2006AA01Z122. Especially, we would like to express deep gratitude to our colleague *Haoyu Ren*. This paper would be incomplete without his tremendous contribution for implementing part of the approach.

References

[1] S. Agarwal, A. Awan and D. Roth. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. In *PAMI*, Vol. 26, pp.1475-1490, 2004.

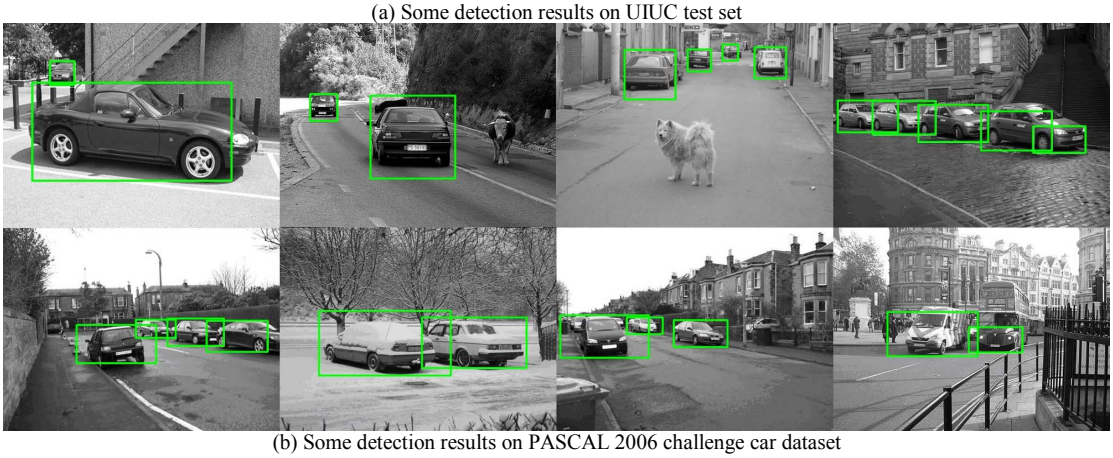
[2] H. Bay, T. Tuytelaars and L. V. Gool. SURF: Speeded Up Robust Features. In *ECCV*, 2006.

[3] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[4] P. Dollar, Z. Tu, H. Tao and S. Belongie. Feature Mining for Image Classification. In *CVPR*, 2007.

[5] M. Everingham, A. Zisserman, C. Williams and L. V. Gool. The Pascal Visual Object Classes Challenge 2006 (VOC2006) Results.

(a) Some detection results on UIUC test set



(b) Some detection results on PASCAL 2006 challenge car dataset

Fig. 8. Example detection results of the I-Strip and complexity-aware criterion based system.

[6] P. Felzenszwalb, D. McAllester and D. Ramanan. A Discriminatively Trained, Multiscale, Deformable Part Model. In *CVPR*, 2008.

[7] R. Fergus, P. Perona and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *CVPR*, 2003.

[8] M. Fritz, B. Leibe, B. Caputo and B. Schiele. Integrating Representative and Discriminative Models for Object Category Detection. In *ICCV*, 2005.

[9] D. M. Gavrila. Pedestrian Detection from a Moving Vehicle. In *ECCV*, 2000.

[10] C. Huang, H. Ai, Y. Li and S. Lao. Vector Boosting for Rotation Invariant Multi-View Face Detection. In *ICCV*, 2005.

[11] Y. Huang, K. Huang, L. Wang, D. Tao, T. Tan and X. Li. Enhanced Biologically Inspired Model. In *CVPR*, 2008.

[12] A. Kapoor and J. Winn. Located Hidden Random Fields: Learning Discriminative Parts for Object Detection. In *ECCV*, 2006.

[13] C. H. Lampert, M. B. Blaschko and T. Hofmann. Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In *CVPR*, 2008.

[14] B. Leibe, A. Leonardis and B. Schiele. Robust Object Detection with Interleaved Categorization and Segmentation. In *IJCV*, Vol. 77, pp. 259-289, 2008.

[15] J. Liebelt, C. Schmid and K. Schertler. Viewpoint -Independent Object Class Detection Using 3D Feature Maps. In *CVPR*, 2008.

[16] R. Lienhart and J. Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. In *ICIP*, 2002.

[17] D. G. Lowe. Distinctive Image Features from Scale-Invariant Key- Points. In *IJCV*, Vol. 60, pp. 91–110, 2004.

[18] T. Mita, T. Kaneko and O. Hori. Joint Haar-like Features for Face Detection. In *ICCV*, 2005.

[19] A. Mohan, C. Papageorgiou and T. Poggio. Example-Based Object Detection in Images by Components. In *PAMI*, Vol. 23, pp.349 – 361, 2001.

[20] J. Mutch and D. G. Lowe. Multiclass Object Recognition with Sparse, Localized Features. In *CVPR*, 2006.

[21] C. Papageorgiou and T. Poggio. A Trainable System for Object Detection. In *IJCV*, Vol.38, pp. 15-33, 2000.

[22] P. Sabzmeydani and G. Mori. Detecting Pedestrians by Learning Shapelet Features. In *CVPR*, 2007.

[23] R. E. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. In *Machine Learning*, Vol. 37, pp. 297-336, 1999.

[24] H. Schneiderman and T. Kanade. A Statistical Method for 3D Object Detection Applied to Faces and Cars. In *CVPR*, 2000.

[25] T. Serre, L. Wolf and T. Poggio. Object Recognition with Features Inspired by Visual Cortex. In *CVPR*, 2005.

[26] V. Sharma and J. W. Davis. Integrating Appearance and Motion Cues for Simultaneous Detection and Segmentation of Pedestrians. In *ICCV*, 2007.

[27] J. Shotton, A. Blake and R. Cipolla. Contour-Based Learning for Object Detection. In *ICCV*, 2005.

[28] Z. Sun, G. Bebis and R. Miller. On-Road Vehicle Detection Using Gabor Filters and Support Vector Machines. In *ICDSP*, 2002.

[29] O. Tuzel, F. Porikli and P. Meer. Human Detection via Classification on Riemannian Manifolds. In *CVPR*, 2007.

[30] P.Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *CVPR*, 2001.

[31] J. Winn and J. Shotton. The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects. In *CVPR*, 2006.

[32] B. Wu and R. Nevatia. Optimizing Discrimination-Efficiency Tradeoff in Integrating Heterogeneous Local Features for Object Detection. In *CVPR*, 2008.

[33] B. Wu and R. Nevatia. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In *ICCV*, 2005.

[34] B. Wu and R. Nevatia. Cluster Boosted Tree Classifier for Multi-View, Multi-Pose Object Detection. In *ICCV*, 2007.

[35] Z. Zhu, Y. Zhao and H. Lu. Sequential Architecture for Efficient Car Detection. In *CVPR*, 2007.

[36] J. Shotton, A. Blake and R. Cipolla. Efficiently Combining Contour and Texture Cues for Object Recognition. In *BMVC*, 2008.

[37] http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[38] http://cbcl.mit.edu/software-datasets/streetscenes/