Strip Features for Fast Object Detection

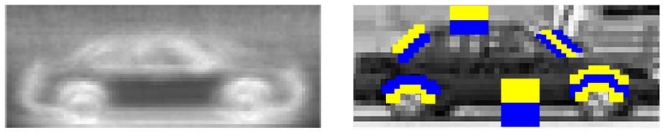
Wei Zheng, Hong Chang, Luhong Liang, *Member, IEEE*, Haoyu Ren,
Shiguang Shan, *Member, IEEE*, and Xilin Chen, *Senior Member, IEEE*

***Abstract*—This paper presents a set of effective and efficient features, namely strip features, for detecting objects in real-scene images. Although shapes of a specific class usually have large intraclass variance, some basic local shape elements are relatively stable. Based on this observation, we propose a set of strip features to describe the appearances of those shape elements. Strip features capture object shapes with edgelike and ridgelike strip patterns, which significantly enrich the efficient features such as Haar-like and edgelet features. The proposed features can be efficiently calculated via two kinds of approaches. Moreover, the proposed features can be extended to a perturbed version (namely, perturbed strip features) to alleviate the misalignment caused by deformations. We utilize strip features for object detection under an improved boosting framework, which adopts a complexity-aware criterion to balance the discriminability and efficiency for feature selection. We evaluate the proposed approach for object detection on the public data sets, and the experimental results show the effectiveness and efficiency of the proposed approach.**

***Index Terms*—Complexity-aware criterion, object detection, strip features.**

I. Introduction

OBJECT detection is a fundamental problem in computer vision and pattern recognition, and it is an indispensable technology in emerging applications such as video surveillance, driver assistance, and content-based image retrieval. Lots of researchers have paid much attention to object detection and proposed many powerful features and discriminative algorithms. Although great advances have been made for object detection, it is still a challenging problem to design a reliable object detector in images. One of the most important reasons is that the appearances of different objects change dramatically due to viewpoints, illuminations, deformations and occlusions. To overcome the large variance in appearance, all kinds of cues, such as shape, color and texture, are extracted from images for object detection. Among these cues, shape is considered as one of the most discriminative and reliable one. Therefore, lots of features are proposed to describe shape characteristics for object detection. For example, edgelet [28], contour fragments [24], and active basis [31] explicitly describe shapes based on shape templates, whereas histogram of oriented gradients (HOGs) [4], adaptive contours [14], and cooccurrence HOG (CoHOG) [21] implicitly describe shapes based on the statistical information of oriented gradients in local regions. These shape-guided features have achieved impressive performance on object detection for some specific classes, such as faces [26], [29], cars [7], [8], [35] and persons [4], [32], [34].



(a) (b)

Fig. 1. Car structures versus strip features. (a) Shapes of cars have large intraclass variance. (b) Components of cars consist of stable shape elements, such as line and arc strips.

Designing efficient features based on shape cue faces two challenging problems. First, the shapes of one object class have large intraclass variance due to many factors, such as translation variance, scale variance, and deformation. We intuitively show the consequence caused by these factors in Fig. 1(a). We extract the edge maps of the side-view cars and average the edge magnitudes. Apparently, the edge maps cannot be well aligned since there are different structures and different translations and scales for different car images. As a result, it is difficult to find a shape-guided feature that is well aligned with the edges of all the cars. Second, there seems to be a contradiction between discriminability and computation cost of features. Powerful shape features or descriptors (e.g., HOG [4] and CoHOG [21]) usually have high computation costs, which will slow down the detection process. On the contrary, most of the fast features may be not robust or discriminative for describing objects with complex shapes. For example, edgelet features [28] are susceptible to shape variance as they are based on pointwise matching, and Haar-like features [26] are short of discriminability as they describe shapes with simple contrast patterns in rectangular regions. Therefore, it is a formidable problem to design such a feature that has powerful discriminability and low computation cost.

Although the shapes of a specific class have large intraclass variance, some basic shape elements are relatively stable. As shown in Fig. 1(b), we take the side-view cars as an example. We can find many consistent shape elements on the wheels,

Manuscript received January 31, 2012; revised September 22, 2012; accepted December 4, 2012. Date of publication January 18, 2013; date of current version November 18, 2013. This work was supported in part by the National Basic Research Program of China (973 Program) under Contract 2009CB320902; by the Natural Science Foundation of China under Contract 61222211, Contract 61272319, and Contract 61272321; and by Beijing Natural Science Foundation (New Technologies and Methods in Intelligent Video Surveillance for Public Security) under Contract 4111003. This paper was recommended by Associate Editor H. Zhang.

W. Zheng, H. Chang, S. Shan, and X. Chen are with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wei.zheng@vipl.ict.ac.cn; hong.chang@vipl.ict.ac.cn; shiguang.shan@vipl.ict.ac.cn; xilin.chen@vipl.ict.ac.cn).

L. Liang was with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. He is now with the Hong Kong Applied Science and Technology Research Institute, Hong Kong (e-mail: luhongster@gmail.com).

H. Ren was with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. He is now with the Beijing Samsung Research Center, Beijing 100016, China (e-mail: 4bitadder@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2012.2235066

pillars, bumpers, roofs, and chassis of different cars. We highlight some of these car structures using blue–yellow strips. There are strong intensity contrasts between the yellow strips and the associated blue strips. The widths of the strips reflect the scales of the car structure, and the shapes of the strips reflect the patterns of the associated structures. Obviously, the shapes of the strips consist of some basic geometric elements, such as lines and arcs. There are such strip patterns on other object classes, such as persons, cows, and bicycles. These strips are informative to describe the shape characteristics in different scales and thus provide us with crucial cues for distinguishing the target objects from background.

Based on this observation, we propose a novel set of shape features, namely strip features, to explicitly describe the shape characteristics for object detection in still images. The proposed feature set consists of lines and arcs with edgelike and ridgelike strip patterns of different widths. Strip features have three merits for object detection. First, the proposed features describe shape characteristics based on local regions rather than single-pixel-width shape templates; thus, they may be robust to translation and scale variance. Second, the proposed features can be efficiently calculated via two kinds of approaches, one of which is based on integral images. Third, the proposed features can be easily extended to a perturbed version (namely, perturbed strip features) to alleviate the misalignment caused by shape variances. We utilize strip features for object detection under an improved boosting framework. The boosting algorithm assembles many features into a discriminative classifier according to a *complexity-aware* criterion, which balances the discriminability and efficiency of the features. Experimental results on the University of Illinois at Urbana-Champaign (UIUC) car data set [1] and the Visual Object Classes 2006 (VOC 2006) data set [6] show that our approach achieves impressive performance and a fast speed.

The main contributions of this paper are in three folds: 1) a novel set of strip features and its perturbed version for object detection; 2) two efficient approaches for calculating strip features; and 3) a new complexity-aware criterion in boosting framework to balance the discriminability and efficiency.

The rest of this paper is organized as follows. Section II reviews the related works. Section III describes strip features. Section IV elaborates the complexity-aware criterion in a boosting framework. Section V shows our experimental results. The last section draws our conclusions.

II. Related Works

There has been extensive literature on object detection. Most of these approaches address the detection problem from two aspects, i.e., designing features or descriptors and developing generative models or discriminative algorithms. Therefore, we review the related works from the above two aspects.

Efficiency and discriminability are two major considerations for designing features or image descriptors. On one side, the efficiency of features is critical for fast detectors. Lots of detection approaches adopt simple features with low computation costs. Two kinds of fast features are widely used for object detection, i.e., edge template features and binary pattern features. For edge template features, they describe shapes with single-pixel-edge template, such as edgelet [28] and contour fragments [24]. The feature responses can be efficiently calculated via a lookup table [28] or a fast shape-matching algorithm [24]. For binary pattern features, they describe object patterns with intensity contrast patterns, such as Haar-like [26], local binary pattern (LBP) [2], and local assembled binary [33]. These features can be rapidly calculated via integral images [26]. On the other side, the discriminability of features decides the final accuracy of the object detectors. Thus, a variety of complex features or descriptors is designed to capture useful information from images. We review these complex features from two aspects, i.e., statistical descriptors and shape descriptors. Statistical descriptors describe object patterns based on various statistics of images rather than explicit describe patterns, such as covariance descriptors [20], scale-invariant feature transform descriptors [16], HOG features [4], and adaptive contours [14]. These features may be unsuitable for fast object detection due to the high computation costs. Unlike statistical descriptors, shape descriptors [19], [25] encode the contours, which consist of boundaries and meaningful inner edges [24], to explicitly represent object shapes. This kind of approaches is based on a powerful contour representation and an effective matching algorithm. Since the contour maps are generated by edge detection algorithms [3], [17], the detection results of such approaches severely rely on the edge detection algorithms. Similar to statistical descriptors, most of shape descriptors are unsuitable for fast object detection due to high computation cost in the matching process. Different from the above features, strip features explicitly describe shape cues without relying on the edge detection algorithms. All the above features are manually designed, whereas some other features are autogenerated by generative algorithms, such as principal component analysis [38], independent component analysis [40], and nonnegative matrix factorization [39]. These features are widely used for object recognition but seldom used for object detection. One of the reasons is that the autogenerated features are less efficient in both training and testing processes than the manually designed features (e.g., Haar-like [26], edgelet [28], HOG [4] and LBP [2]). Moreover, the autogenerated features cannot outperform the manually designed ones in object detection [37], [41]. Therefore, we prefer designing strip features with the fast calculation algorithm rather than autogenerating the features.

Various generative models and discriminative algorithms have been proposed for object detection. Generative approaches represent objects in a probabilistic framework, such as a part-based generative model [11] and an implicit shape model [8], [12]. Unlike generative models, discriminative algorithms (e.g., SVM [4] and boosting [26]) treat object detection as a binary classification problem and output a binary label (positive or negative) for each sample. Among these algorithms, boosting is widely used for object detection [20], [26], [28] as it is capable of assembling a large amount of simple features into a discriminative classifier. Most of the boosting algorithms greedily select the features according to the discriminability criterion. Since different features have different computation costs, the complex features may slow down the detection process if they are selected according to the discriminability.

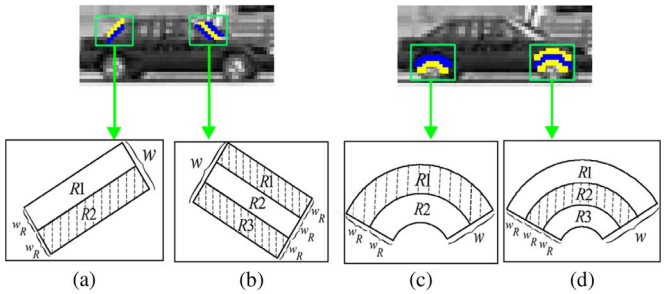


Fig. 2. Relationship between strip features and typical structures on cars. (a) Edgelike line. (b) Ridgelike line. (c) Edgelike arc. (d) Ridgelike arc.

Recently, several researchers propose new feature selection criteria in a boosting framework to balance the discriminability and efficiency of features. They select the heterogeneous features according to the quotient of the discriminability and computation cost [22], [27]. Previous works deduce the complexity-aware criterion for only one classifier, whereas we deduce the proposed complexity-aware criterion by minimizing the detection time of the cascaded classifier that consists of a group of classifiers. Thus, the proposed criterion may be more suitable for the cascaded object detector.

III. Strip Features

As mentioned in Section I, there are many challenging problems for designing effective and efficient shape features, such as variances in translation or scale and deformation. In this section, we propose strip features and the perturbed version to address these problems to some extent. Moreover, we elaborate two kinds of fast calculation approaches for strip features.

A. Definition of Strip Features

Objects of a specific class generally have relatively stable characteristics in shape. We take side-view cars as an example in Fig. 2. Apparently, all of the cars have common structural components such as wheels, pillars, and bumpers. Although the appearances of these components may look different due to variations of car models and illumination, they consist of some basic shape elements, such as lines and arcs with edgelike and ridgelike contrast patterns. As shown in Fig. 2, the appearance of a tire can be represented by several arcs with the length of 6–9 pixels and a ridgelike pattern. These characteristics provide crucial cues for discriminating cars and background. Likewise, we can also find such common strip patterns on the other object classes, such as bicycles, cows, and persons. Guided by the given observations, we propose a set of strip features to describe these shape elements on objects. A strip feature can be formally represented by triplet $S = \langle c_L, t_w, p \rangle$, where c_L represents the curve pattern, with L being the number of pixels; t_w represents the contrast pattern, with w being the width of the strip; and p represents the position of the feature in the detection window. All the features with valid c_L, t_w, and p form feature set $\{S\}$. In this paper, we suppose that the curve patterns of strip features contain lines and arcs, and the contrast patterns of strip features contain edgelike and ridgelike patterns.

As shown in Fig. 2(a) and (c), an edgelike feature can be described by two back-to-back *substrip regions* with the same curve pattern and width w_R, whereas a ridgelike feature can be described by three substrip regions. The responses of the features can be calculated via the averaged intensities of the substrip regions according to

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

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

where $I(x,y)$ is the intensity at (x,y), $|\bullet|$ represents the total number of pixels in a particular region, and $R1$, $R2$, and $R3$ are the substrip regions, as shown in Fig. 2.

We calculate the feature responses according to (1) and (2). Such feature responses with absolute value may be more robust to describe the contrast pattern than that without an absolute value. Taking cars as an example, the white cars may be brighter than the background, whereas the black cars may be darker than the background. Using the absolute values as the feature responses, strip features can describe the contrast information that may be robust to the color variances of the objects.

Based on the observation of the object structure, we specifically restrict the curve patterns c_L to lines, 1/8 circles, 1/4 circles, and 1/2 circles, which are similar to edgelet [28]. The substrip regions in one strip feature have the same width, i.e., $w = 2w_R$ (edgelike) or $w = 3w_R$ (ridgelike). Moreover, we restrict the curve length L to 4–24 pixels and substrip region width w_R to 2–6 pixels in a detection window. The feature set contains different strip features with different region widths for each curve pattern. The same curve pattern of different region widths can be considered as reflecting the same curve pattern in different scales; thus, strip features are capable of describing various curve patterns of different scales. We can adopt the boosting algorithm to select the strip features with the best curve pattern and scale (i.e., region width).

It deserves to be mentioned that the proposed feature set is not only a meaningful extension of edgelet feature set [28] but also a superior set of Haar-like feature set [26]. Compared with edgelet features, strip features are based on the statistics of regions rather than the single-pixel-width edges. Thus, strip features may be more robust to slight misalignment caused by translation variance, scale variance, and deformation. In some sense, strip features can be considered as joint Haar-like features constrained by some curve patterns. Therefore, strip features can represent typical shapes with basic geometric elements, such as lines and arcs. Of course, Haar-like features belong to a subset of strip features, which only contains simple curve patterns of horizontal and vertical lines. In Fig. 3, we show three exemplars for edgelet, Haar-like, and strip features around the wheel of the car. We translate the features around their initial positions and then calculate the feature responses at different positions. The feature responses are normalized

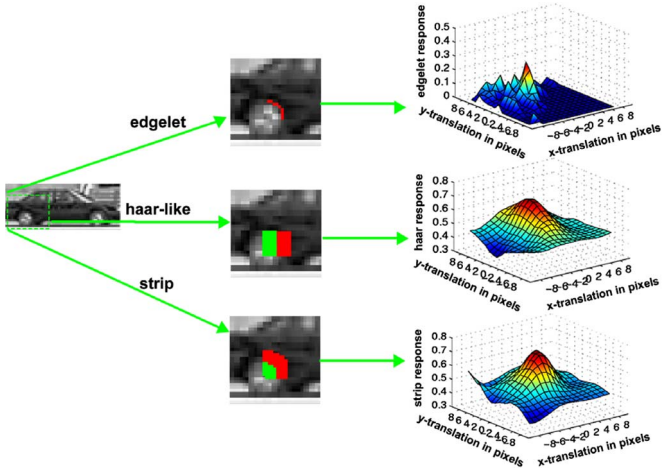


Fig. 3. Feature responses versus translation variances.

into [0, 1] and visualized in Fig. 3. The responses of the edgelet feature have some sharp peaks. It means that the edgelet feature is very sensitive to translation variance. On the contrary, the Haar-like and strip features are robust to the slight misalignment caused by translation variance. For the Haar-like feature, the response peak deviates from (0,0) as it cannot precisely describe the curve pattern of arcs. Compared with the Haar-like feature, the strip feature captures the wheel shape more precisely with arc and edgelike patterns. As a result, the response peak of the strip feature locates at (0,0). Moreover, the response peak of the strip feature is sharper than that of the Haar-like feature, which means it is more precise and discriminative than the Haar-like feature.

B. Fast Feature Extraction

In order to generate a strip feature, we need to specify the curve pattern, contrast pattern, and position, i.e., c_L, t_w, and p. We utilize edgelet [28] to represent c_L and p, and then dilate the edgelet features along the normal directions to form the edgelike and ridgelike strip patterns, as shown in Fig. 2. Obviously, an edgelet can generate multiple strip features by varying t_w. A straightforward way of extracting the strip features is calculating the response according to (1) and (2) by directly summing all the points, namely the DStrip approach (or feature). However, the computation cost will be expensive when the strip regions contain many pixels. Thus, we propose an alternative method based on integral image to calculate the features, namely IStrip approach (or feature).

DStrip Approach: We can directly calculate the average intensity of each substrip region point-by-point. The coordinates of pixels within each substrip region can be easily obtained by a flood fill algorithm and can be then stored in a list, so that one edgelike feature has two lists and one ridgelike feature has three lists. For feature extraction, the feature response can be calculated according to (1) or (2) using the point lists. This pointwise method needs a lot of memory and high computation cost for calculating the average intensities of the substrip regions, particularly when the substrip regions are of large sizes.

IStrip Approach: In order to reduce the memory and computation cost, we propose an approximate algorithm based on an integral image. Apparently, when the curve patterns only contain horizontal and vertical lines, strip features degrade to Haar-like features and can be directly extracted using the integral image approach [26]. The integral image can be computed from an image using a few operations per pixel. Once computed, the average intensity of a rectangle region can be computed at any scale or location in a constant time. In this paper, we focus on the features with oblique line and arc patterns, as shown in Fig. 4(a) and (b). We employ two series of small upright rectangles to represent the upper and lower substrip regions, respectively. More specifically, we assign two *associated rectangles* shown as $R1_i$ and $R2_i$ in Fig. 4(a) for each point P_i along the edgelet. $R1_i$ is an upright rectangle that 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 substrip region. $R2_i$ and other associated rectangles can be determined in a similar way. Then, the response of the strip feature in Fig. 4(a) can be calculated according to

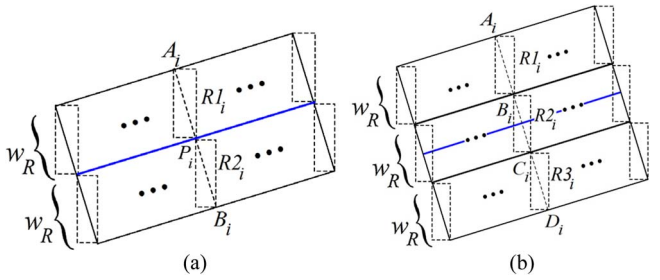


Fig. 4. Line features approximated by associate rectangles. (a) Edgelike. (b) Ridgelike.

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

where function $g(\bullet)$ sums up the intensities of all the points in a particular region via the integral image approach [26], and L is the length of the edgelet feature.

Fig. 4(b) illustrates a ridgelike feature with the oblique line pattern. Similar to the edgelike 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 a series of associated rectangles, the response of the ridgelike feature can be calculated according to

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

Likewise, strip features of arc patterns can be calculated through the associated rectangles according to (3) and (4).

C. Perturbed Strip Features

Strip features only contain the parametric curve patterns, such as lines and arcs. However, most of the shapes cannot be represented using lines or arcs precisely due to slight misalignment caused by translation variance, scale variance, and

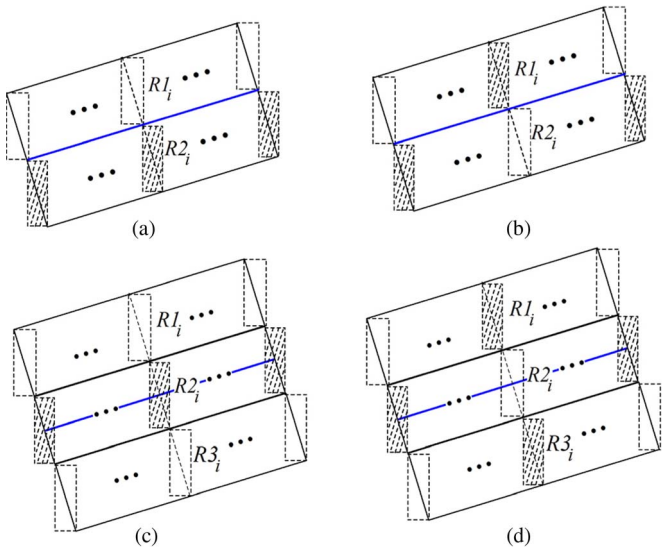


Fig. 5. Different calculations of strip features describe different contrast patterns.

deformation. To alleviate such misalignment, we first improve the flexibility of IStrip features by modifying calculation of feature responses and then elaborate perturbed strip features based on the improved IStrip features.

As discussed in Section III-B, we can calculate responses of IStrip features according to (3) and (4). In this section, we modify the calculation of feature responses by summing the absolute differences of the associated rectangle pairs as follows:

$$f_{\text{edge}} = \sum_{i=1}^{L} \left| \frac{g\left(R1_i^{(x_i,y_i)}\right)}{|R1_i|} - \frac{g\left(R2_i^{(x_i,y_i)}\right)}{|R2_i|} \right|, \tag{5}$$

$$f_{\text{ridge}} = \frac{1}{2} \sum_{i=1}^{L} \left| \frac{g\left(R1_i^{(x_i,y_i)}\right)}{|R1_i|} + \frac{g\left(R3_i^{(x_i,y_i)}\right)}{|R3_i|} - \frac{2g\left(R2_i^{(x_i,y_i)}\right)}{|R2_i|} \right|. \tag{6}$$

It is worth mentioning that (5) can describe more contrast patterns than (3). If we calculate the feature response according to (3), strip features can only reflect simple contrast patterns, as shown in Fig. 5(a). For such simple patterns, the associated rectangles should be consistently bright in one substrip region and dark in the other substrip region. If we calculate the feature response according to (5), the strip feature can describe the complex contrast patterns, as shown in Fig. 5(b). The contrast pattern of an associated rectangle pair can perturb to an inverse contrast pattern, and the rectangles in one substrip region may be not consistently bright or dark. Equation (5) can describe the contrast patterns in Fig. 5(a) and (b). Likewise, (6) can describe the contrast patterns in Fig. 5(c) and (d). If we calculate the response of strip features according to (5) and (6), the contrast patterns of the associated rectangle pairs can be more flexible to describe some inconsistent edges in real-scene images.

Based on the given improved IStrip features, we perturb the associated rectangles in a neighborhood to make the features

Algorithm 1: Calculating perturbed strip features.

1: $\mathbf{N}$ is predefined neighborhood, L is length of curve pattern
2: Supposing $\delta_i^{(x_i,y_i)}$ is contrast of i^{th} associated rectangles with offset (x_i, y_i),i.e.,
3: $\delta_i^{(x_i,y_i)} = \frac{g(R1_i^{(x_i,y_i)})}{|R1_i|} - \frac{g(R2_i^{(x_i,y_i)})}{|R2_i|}$,
4: $\delta_i^{(x_i,y_i)} = \frac{1}{2}(\frac{g(R1_i^{(x_i,y_i)})}{|R1_i|} + \frac{g(R3_i^{(x_i,y_i)})}{|R3_i|} - \frac{2g(R2_i^{(x_i,y_i)})}{|R2_i|})$.
5: Initialization: $f = 0$.
6: **for** $t = 1 : L$ **do**
7: find $(x_i^*, y_i^*) \in \mathbf{N}$ that maximize $|\delta_i^{(x_i,y_i)}|$.
8: $f = f + |\delta_i^{(x_i^*,y_i^*)}|$.
9: **end for**
10: **return** f

Fig. 6. Algorithm for calculating perturbed strip features.

well aligned with different images. We utilize such a simple hypothesis that a strip feature is well aligned with an image when its response reaches the highest value during perturbation. As shown in Fig. 3, the feature response should be the local maxima if it matches the actual boundaries of the wheel. Therefore, the perturbed strip features can be calculated according to

$$f_{\text{edge}} = \max_{(x_1,y_1)\in\mathbf{N}_1,\ldots,(x_L,y_L)\in\mathbf{N}_L} \sum_{i=1}^{L} \left| \frac{g\left(R1_i^{(x_i,y_i)}\right)}{|R1_i|} - \frac{g\left(R2_i^{(x_i,y_i)}\right)}{|R2_i|} \right|, \tag{7}$$

$$f_{\text{ridge}} = \frac{1}{2} \max_{(x_1,y_1)\in\mathbf{N}_1,\ldots,(x_L,y_L)\in\mathbf{N}_L} \sum_{i=1}^{L} \left| \frac{g\left(R1_i^{(x_i,y_i)}\right)}{|R1_i|} + \frac{g\left(R3_i^{(x_i,y_i)}\right)}{|R3_i|} - \frac{2g\left(R2_i^{(x_i,y_i)}\right)}{|R2_i|} \right|, \tag{8}$$

where $R1_i^{(x_i,y_i)}$ represents a rectangle generated by perturbing $R1_i$ with the offset (x_i, y_i), and $\mathbf{N}_i$ is the neighborhood. Of course, the curve pattern will be destroyed if the associated rectangles move arbitrarily in the neighborhood. To preserve the curve pattern to some extent, we constrain that the neighborhood contains 9 pixels surrounding the ith edgelet point. The computation complexity of (7) and (8) is $\prod_{i=1}^{L} |N_i|$, where $|\mathbf{N}_i|$ is the number of pixels in $\mathbf{N}_i$, and L is the number of the rectangle pairs. It is too expensive for fast object detectors. Hereby, we constrain that all the neighborhoods have the same size and simplify (7) and (8) as follows:

$$f_{\text{edge}} = \sum_{i=1}^{L} \max_{(x_i,y_i)\in\mathbf{N}} \left| \frac{g\left(R1_i^{(x_i,y_i)}\right)}{|R1_i|} - \frac{g\left(R2_i^{(x_i,y_i)}\right)}{|R2_i|} \right|, \tag{9}$$

$$f_{\text{ridge}} = \frac{1}{2} \sum_{i=1}^{L} \max_{(x_i,y_i)\in\mathbf{N}} \left| \frac{g\left(R1_i^{(x_i,y_i)}\right)}{|R1_i|} + \frac{g\left(R3_i^{(x_i,y_i)}\right)}{|R3_i|} - \frac{2g\left(R2_i^{(x_i,y_i)}\right)}{|R2_i|} \right|. \tag{10}$$

TABLE I
COMPUTATION COSTS OF ISTRIP AND DSTRIP

Feature Type / Operation Type	IStrip				DStrip		Perturbed strip			
	Haar-like		Non-Haar-like		Edge	Ridge	Haar-like		Non-Haar-like	
	Edge	Ridge	Edge	Ridge			Edge	Ridge	Edge	Ridge
Addition	7	11	8L - 1	12 L - 1	2 L w_R + 1	3 L w_R + 2	7|**N**|	11|**N**|	7|**N**|L+L-1	11|**N**|L +L-1
Multiplication	2	5	2	5	2	5	2|**N**|	4|**N**|+1	2|**N**|L	4|**N**|L+1
Modulus	1	1	1	1	1	1	|**N**|	|**N**|	|**N**|L	|**N**|L
Total	10	17	8 L + 2	12 L + 5	2 L w_R + 4	3 L w_R + 8	10|**N**|	16|**N**|+1	10|**N**|L+L-1	16|**N**|L + L

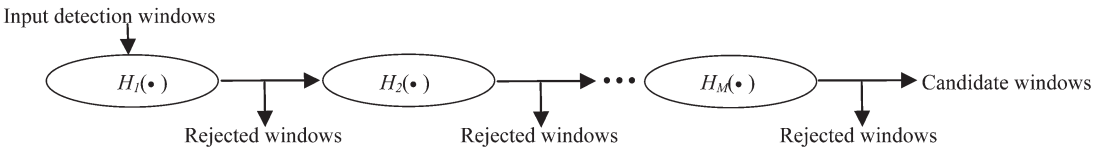


Fig. 7. Cascaded classifier for object detection.

We can see that the maximum value of (9) and (10) can be solved sequentially, thus the computation complexity becomes $|\mathbf{N}|L$ that is much smaller than $|\mathbf{N}|^L$. The algorithm of calculating perturbed strip features is shown in Fig. 6. It is worth mentioning that the proposed perturbation algorithm is a straightforward method to deform strip features as it may destroy the curve patterns of strip features. However, the perturbation algorithm is effective in our experiment, and detailed analysis is given in Section V-D.

To this end, we give an analysis on the computation costs of DStrip and IStrip approaches. We suppose that $g(\bullet)$ consumes three additions via the integral image approach, and L and w_R are given. According to (1)–(4), and (9)-(10), we list the computation costs in Table I. For simplification, we consider that the addition, multiplication, and modulus operations have the same computation cost, and we list the total computation cost in the last row of Table I. As shown in Table I, the computation costs of Haar-like features are constant. The computation costs of the IStrip approach are irrelevant to w_R. For the non-Haar-like features, the IStrip approach is faster than the Dstrip approach when w_R is larger than a width of 4 pixels, whereas the DStrip approach is faster than the IStrip approach when w_R is smaller than a width of 4 pixels.

IV. COMPLEXITY-AWARE REALBOOST

In Table I, we can see that the computation costs of different strip features are greatly different. The simplest IStrip feature consumes only ten operations, whereas the ridgelike feature with $L = 12$ consumes 149 operations. If the earlier stages of the cascaded classifier use the "cheaper" features, they can reject a lot of negative detection windows with low computation cost. In order to obtain an effective and efficient classifier, we propose a *complexity-aware* criterion to balance the discriminability and computation cost for the features under the boosting framework, i.e.,

$$Z' = Z + aT, \tag{11}$$

where Z is the discriminative criterion, i.e., the measurement of the discriminability of the weak classifiers; T is the *complexity criterion* that can be obtained by minimizing the expectation of the runtime for the cascaded classifier; and a is the *complexity-aware factor* to balance the discriminability and computation cost. In a RealBoost framework [23], the discriminability is measured by the Bhattacharyya coefficient between the distributions of the object and nonobject classes, i.e.,

$$Z = 2\sum_k \sqrt{W_+^k W_-^k}, \tag{12}$$

where $W_+^j(W_-^j)$ is the distribution of the feature response for positive (negative) samples. As shown in Fig. 7, we adopt a cascaded classifier [26] for object detection. The cascaded classifier is sequentially assembled by M strong classifiers. Each strong classifier consists of several weak classifiers, and each weak classifier is learned by the boosting algorithm based on a strip feature. We denote the total runtime of the cascaded classifier as T_{pos} and T_{neg} for the positive and negative detection windows, respectively, and the number of the true positive and false positive windows processed via the ith strong classifier by N_i^+ and N_i^-, respectively. Then, the total runtime is

$$E(T) = T_{\text{pos}} + T_{\text{neg}} = \sum_{i=1}^{M} N_i^+ S_i + \sum_{i=1}^{M} N_i^- S_i, \tag{13}$$

where S_i is the computation cost of the ith strong classifier, and M is the total number of the strong classifiers.

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

$$\begin{aligned} E(T) \approx T_{\text{neg}} &= \sum_{i=1}^{M} N_i^- S_i \\ &= \sum_{i=1}^{M} (N\ \text{fp}_{i-1}) S_i = N \sum_{i=1}^{M} \text{fp}_{i-1} S_i, \end{aligned} \tag{14}$$

where N is the total number of the detection windows, and fp_i is the false positive rate of the ith strong classifier. We use the computation costs of strip features in Table I to represent the

TABLE II
SIZES OF DETECTION WINDOWS AND FEATURE SETS

	UIUC cars	Cars	Bicycles	Buses	Motorbikes	Persons	Cows	Sheep	Horses	Dogs	Cats
Window	100×40	80×40	90×60	80×80	96×60	48×108	90×60	80×48	80×48	80×80	80×80
Haar-like	190,983	186,720	194,850	187,860	160,524	193,023	194,850	121,200	121,200	187,860	187,860
Edgelet	80,331	81,200	72,784	71,897	62,626	78,678	72,784	70,320	70,320	71,897	71,897
Strip	297,271	298,030	294,175	248,117	241,851	297,215	294,175	206,730	206,730	248,117	248,117

computation costs of the strong classifiers, and the expectation of the total runtime can be represented as

$$E(T) \approx N \sum_{i=1}^{M} \mathrm{fp}_{i-1} S_i = N \sum_{i=1}^{M} \mathrm{fp}_{i-1} \sum_{j=1}^{m_i} C_{i,j} = N \sum_{i=1}^{M} \sum_{j=1}^{m_i} \mathrm{fp}_{i-1} C_{i,j}, \quad (15)$$

where $C_{i,j}$ is the computation cost of the jth features of the ith strong classifier, as shown in Table I, and m_i is the number of features in the ith strong classifier. To minimize $E(T)$, we greedily select the feature with the minimum value of $\mathrm{fp}_{i-1} C_{i,j}$ in each boosting round. Substituting $\mathrm{fp}_{i-1} C_{i,j}$ into (11), we derive the final complexity-aware criterion for selecting the jth feature of the ith strong classifier, i.e.,

$$Z'_{i,j} = 2 \sum_{k} \sqrt{W_+^k W_-^k} + a\, \mathrm{fp}_{i-1} C_{i,j}. \quad (16)$$

Intuitively, we can explain (16) as follows. When the false positive rate is large, the proposed criterion tends to select the features with cheaper computation cost. The computation cost plays a more important role than discriminability for the first several stages of strong classifiers since they should efficiently evaluate a large amount of detection windows and reject most of the detection windows by simple features. When the false positive rate becomes smaller, the proposed criterion adaptively makes the discriminability more important than computation cost. It will not dramatically slow down the detection process as there are only a few detection windows to be processed when the false positive rate is small.

The existing criteria [22], [27] assign a fixed computation cost for the features, whereas the proposed criterion reevaluates the computation costs of features and deduces an adaptive computation cost [i.e., $\mathrm{fp}_{i-1} C_{i,j}$ in (15)] according to the cascaded classifier that consists of many strong classifiers, as shown in Fig. 7. According to the existing criteria, two different strong classifiers of the cascaded classifier balance the discriminability and efficiency of features in the same way. For example, the criteria of the ith and jth strong classifiers are the same. However, the proposed method can adaptively tune the criteria for different strong classifiers. Thus, the criteria of the ith and jth strong classifiers are different since the false positive rates of these two strong classifiers are different. According to the proposed criterion, the earlier strong classifiers of the cascaded classifier give larger weights [i.e., larger false positive rate in (16)] to the computation costs, and the latter strong classifiers give smaller weights to the computation costs.

V. EXPERIMENTS

We evaluate the performance of the proposed approach on the widely used public data sets, including both man-made objects and natural objects, namely the UIUC car data set [1] and the VOC 2006 data set [6]. For different object classes, the sizes of detection windows and feature sets are also different. We also implement two other related features, namely Haar-like and edgelet, for comparison. We uniformly sample the parameters to generate the feature sets. The sizes of detection windows and the feature sets in our experiments are listed in Table II. Since the patterns of Haar-like features are simpler than that of edgelets and strip features, a much more dense sampling strategy is used in the feature generation process. To accelerate the training process, we only evaluate 10 000 randomly selected features rather than the overall feature set. For example, the feature set of side-view cars consists of 297 271 features, and we only sample 10 000 features to evaluate in each boosting round. It can be guaranteed that at least one of the top 148 features can be selected with a probability of $99.3\%(= 1 - (1 - 0.0005)^{10000})$. In the training process, we set the complexity-aware factor a as 0.05.

The object classifiers may generate multiple positive detection windows around an object, and we adopt the mean-shift algorithm to merge the multiple detection results of the same object. For a bounding box, it is considered as a correct detection when the area of overlap between the predicted bounding box B_p and ground-truth bounding box B_{gt} exceeds 50% of their union area, which is indicated by α_0 in the following formula:

$$\alpha_0 = \frac{\mathrm{area}(B_p \cap B_{\mathrm{gt}})}{\mathrm{area}(B_p \cup B_{\mathrm{gt}})}. \quad (17)$$

We evaluate the proposed approach according to Equal Precision and Recall (EPR) rate [8] on the UIUC side-view car data set and the precision–recall curve [6] on the VOC 2006 data set. As discussed in [1] and [37], the common receiver operating characteristic curves are not suitable for evaluating the detectors since they are designed for evaluating the classification results. Thus, we adopt the precision–recall curves to evaluate the object detectors. In addition to the end-to-end comparisons with other popular approaches, evaluations of the individual modules show more insights into the proposed approach. All the approaches are implemented using C++ in Visual Studio 2008 environment without code optimization and tested on a desktop with an Intel 2.93-GHz CPU.

A. UIUC Car Data Set

First, we evaluate strip features on the UIUC car data set [1]. The data set contains a single-scale test set (170 images with

TABLE III
EPRs OF DIFFERENT METHODS ON UIUC CAR DATA SET

Method	Single scale	Multi-scale
Leibe et al. [8]	97.5%	95%
Fergus et al. [11]	~86.5%	-
Mutch & Lowe [18]	99.94%	90.6%
Fritz et al. [9]	88.6%	87.8%
Zhu et al. [36]	~81.0%	-
Casalino et. al. [37]	~61.0%	-
Felzenszwalb et. al. [10]	~92.0%	~87.8%
Dalal et. al. [4]	~91.0%	~86.2%
DStrip + RealBoost	98.0%	95.0%
DStrip + complexity aware	98.0%	96.0%
IStrip + RealBoost	96.3%	95.7%
IStrip + complexity aware	96.5%	96.0%
perturbed strip + RealBoost	98.5%	96.5%
perturbed strip + complexity aware	97.5%	96.0%

200 side-view cars), a multiscale test set (108 images with 139 side-view cars), and a training set of 550 side-view car images. The car images in the training set are horizontally flipped, so that there are 1100 cars in total in the positive training set. Since the bootstrapping requires a large negative training set, we collect the negative images from the VOC 2006 data set. The cars in these images are removed, and the remaining parts are used as the negative training images. A reduced set of 80 331 edgelet features are used for generating strip features, so that the size of the strip feature set is limited to be acceptable for our experimental environment. The total number of strip features is 297 271, as listed in Table II.

We compare the EPRs of the proposed and other approaches. The results are listed in Table III. It can be seen that strip features achieves competitive performance comparing with the other state-of-the-art methods on both single-scale and multi-scale test sets. In Table III, we also compare the performance of different kinds of strip features and different feature selection criteria, i.e., complexity-aware boosting and RealBoost. We can see that the DStrip and IStrip approaches have similar performance. The perturbed strip features can improve the performance. Furthermore, the performance of the complexity-aware-criterion-based algorithm is very close to that of the RealBoost algorithm. In other words, the complexity-aware criterion does not apparently reduce the accuracy. The first row in Fig. 15 shows some detection results of the UIUC side-view car data set.

B. VOC 2006 Data Set

In this section, we evaluate the proposed approach on a more challenging data set, namely the VOC 2006 data set [6]. This data set consists of 2618 training images and 2686 testing images. This data set contains ten object classes including man-made object classes (i.e., bicycles, buses, cars, and motorbikes) and natural object classes (i.e., cows, sheep, horses, cats, dogs, and persons). We evaluate the proposed approach on these object classes. As shown in Table II, we design different object windows and feature sets for different object classes. In the training process, we use nonoccluded samples in the given training set as positive training samples. We remove the positive samples and use the remaining parts as negative training images. We have empirically verified that the complexity-aware boosting performs similarly with RealBoost in Section V-A; thus, we use the complexity-aware boosting in this section for efficiency consideration. For a fair comparison, we implement Haar-like and edgelet features, and then train the classifiers using the proposed complexity-aware boosting framework.

We draw the precision–recall curves in Fig. 8 and list the Average Precision (AP) rates in Table IV. We also give the results of the detection competition [6] and HOG-based support-vector-machine (SVM) linear classifier (HOG + SVM) [5]. It can be seen that strip features achieve promising results on the VOC 2006 data sets. To gauge the statistical significance, we compare the detection performance over the ten object classes of the VOC 2006 data set. In Table V, we list the AP difference of two different approaches for each object class and give the statistical significance in the last column. We can draw several conclusions from the comparison of different approaches: 1) The features based on strip regions (DStrip) perform better than the features based on single-pixel-width templates (edgelet) since they are more robust to slight misalignment than edgelet features; 2) IStrip features perform better than Haar-like features since they can describe more complex shapes; 3) IStrip and DStrip features achieve similar performance since they describe the same contrast pattern for the same region; and 4) the perturbation algorithm improves the performance of IStrip by 0.015 ± 0.011 since it makes strip features more robust to shape misalignment. Some detection results are shown in Fig. 15.

As discussed in Section III, strip features are initially designed according to the typical shape elements of cars, but they also achieve promising results on the natural object classes. The possible reason is that the natural objects, such as persons and cows, also contain many typical shape elements that can be described by strip features. Of course, the proposed approach is not perfect, and it may fail in some situations. In the following, we present some of the situations that may cause our approach to fail. First, each object class contains many objects in different views, and they look very different, such as bicycles and buses. The proposed approach will fail due to the classifier since the classifier adopts a bounding box of a fixed aspect ratio to represent the objects, and it is difficult to detect objects of multiviews. Second, many of the objects are occluded by the other objects or truncated by the image boundaries, such as persons and sheep in crowds, as shown in Fig. 15(d) and (i). The proposed approach may fail since the boosting classifier cannot predict the missing parts caused by occlusion or truncation. Third, some animals have extremely large articulation variance and nonrigid deformations, such as cats and dogs. Both the strip features and the classifier may not work well in this situation. On one hand, strip features may misalign with the animals' structure due to large deformation. On the other hand, the window classifier is unsuitable to represent these animals due to pose and view variances. For example, the shapes of cats and dogs change greatly, and only the shapes of their heads are stable. Thus, we train the classifiers using their heads. Although the classifiers can detect the heads of cats and dogs, as shown in

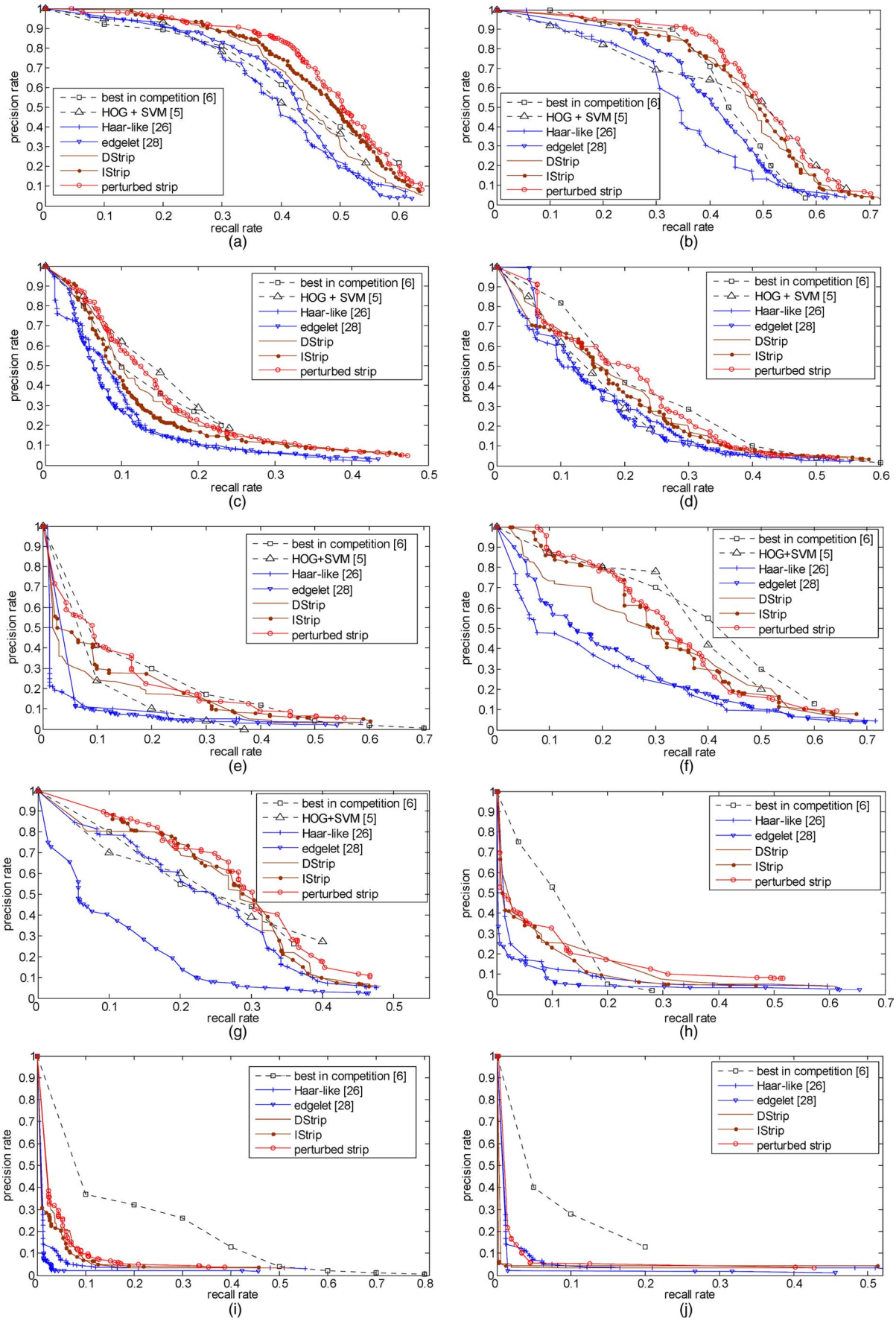


Fig. 8. Precision-versus-recall curves on VOC 2006 dataset. (a) Car. (b) Bicycle. (c) Person. (d) Cow. (e) Bus (f) Motorbike. (g) Sheep. (h) Horse. (i) Cat. (j) Dog.

TABLE IV
COMPARING AP RATES OF DIFFERENT APPROACHES ON THE VOC 2006 DATA SET

	Car	Bicycle	Bus	Motorbike	Person	Cow	Sheep	Horse	Dog	Cat
Best in competition [6]	0.444	0.440	0.169	**0.390**	0.164	0.252	0.251	0.140	**0.118**	**0.160**
HOG + SVM [5]	0.411	0.441	0.132	0.366	**0.183**	**0.254**	0.294	-	-	-
Haar-like	0.397	0.348	0.118	0.219	0.140	0.172	0.253	0.130	0.109	0.107
Edgelet	0.413	0.403	0.115	0.245	0.134	0.183	0.151	0.114	0.091	0.099
DStrip	0.447	0.461	0.154	0.309	0.170	0.217	0.282	0.158	0.103	0.109
IStrip	0.470	0.466	0.171	0.340	0.161	0.212	0.301	0.131	0.110	0.111
Perturbed strip	**0.493**	**0.492**	**0.182**	0.352	0.181	0.231	**0.309**	**0.163**	0.107	0.112

TABLE V
DIFFERENCES OF AP RATES BETWEEN DIFFERENT APPROACHES ON THE VOC 2006 DATA SET

	Car	Bicycle	Bus	Motorbike	Person	Cow	Sheep	Horse	Dog	Cat	Avg. $\pm$ Std.
DStrip - Edgelet	0.034	0.058	0.039	0.064	0.036	0.034	0.131	0.044	0.012	0.010	0.046 ± 0.034
IStrip - Haar	0.073	0.118	0.053	0.121	0.021	0.040	0.048	0.001	0.001	0.004	0.048 ± 0.045
IStrip - DStrip	0.023	0.005	0.017	0.031	-0.009	-0.005	0.019	-0.027	0.007	0.002	0.006 ± 0.017
Perturbed - IStrip	0.023	0.026	0.011	0.012	0.020	0.019	0.008	0.032	-0.003	0.001	0.015 ± 0.011

Fig. 15(j) and (k), the detections are counted as false detections according to the PASCAL criterion. From the experiments of previous work [5], [10], we can see that the detection-window-based approaches may be not suitable for detecting the animal classes, such as cats and dogs. Fourth, some of the object classes have similar shapes, such as bicycles versus motorbikes and sheep versus cows. In this situation, the proposed approach may fail due to strip features since it is difficult to distinguish the objects of one class from the objects of the other class based on shapes. Thus, the bicycle (sheep) classifier based on strip features may have false detection results on motorbikes (cows) and vice versa. These problems could be handled by adding texture and color features, which is beyond the topic of this paper.

It is worth noting that some recent works achieve better accuracy on this challenging data set. Such works utilize more complex models (e.g., deformable part models (DPMs) with different aspect ratio components [10]) or complex features (e.g., heterogeneous features combining HOG and LBP [30]). These approaches usually have high computation complexity and may not be suitable for real-time application. Compared with these approaches, the proposed approach is much faster and can achieve similar performance for detecting the objects of typical shapes (e.g., side-view cars). We will present the detailed comparison in Section V-E. We can improve the performance of the proposed approach by several possible ways, such as training multiview classifiers, training part models, integrating strip features with color or texture features, and developing more powerful deformation approaches (e.g., active contour models [15] and TPS-RPM [13]).

C. Relationship Between Strip Features and Object Structures

In this section, we conduct experiments to study the relationship between the selected strip features and object structures. First, we train a cascaded classifier for side-view cars and show the selected strip features. One hundred side-view car images from the UIUC training set are used. All the cars in the images are toward left and are aligned using the tangent points between the tires and ground. We use the RealBoost algorithm to select IStrip features from the feature set in Table II. We overlay the first four selected strip features on the average edge map in Fig. 9(a). It is interesting that all the four features reflect typical car shape elements. The first feature and the third feature are perfectly on the tires, the second feature describes the contrast of the chassis and the ground, and the fourth feature describes the ridgelike pattern of the front bumper. It shows that strip features can describe shape characteristics of side-view cars.

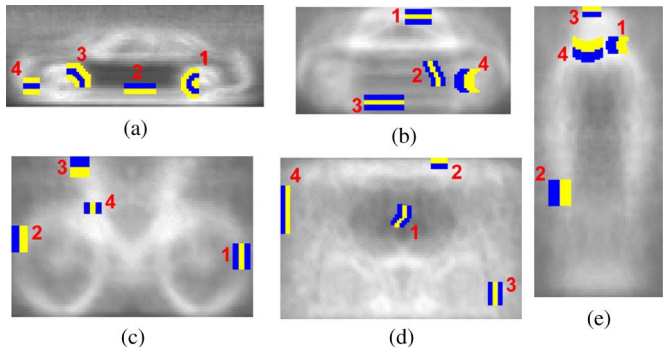


Fig. 9. Top-four selected image strip features versus object structures. (a) Side-view car. (b) Multiview car. (c) Bicycle. (d) Cow. (e) Person.

The above experiment demonstrates the ideal situation, in which the pattern of the aligned cars is rather compact. In practical detection tasks, the relationship between strip features and object structures may not be so intuitive. For the VOC 2006 object classes, we use the training images in Section V-B and the RealBoost algorithm to select IStrip features from the feature set in Table II. We also visualize the top four selected features for some object classes of the VOC 2006 data set in Fig. 9(b)–(e). Although the training images are not strictly aligned, most of the features can still capture the typical object structures. Obviously, the second feature of multiview cars and the first feature of cows totally miss the typical structures. The reason is that the training samples are not well aligned, and the positive pattern is not compact. Therefore, some features may capture the negative patterns instead of the positive pattern. For example, most of the cows do not have high-contrast ridgelike

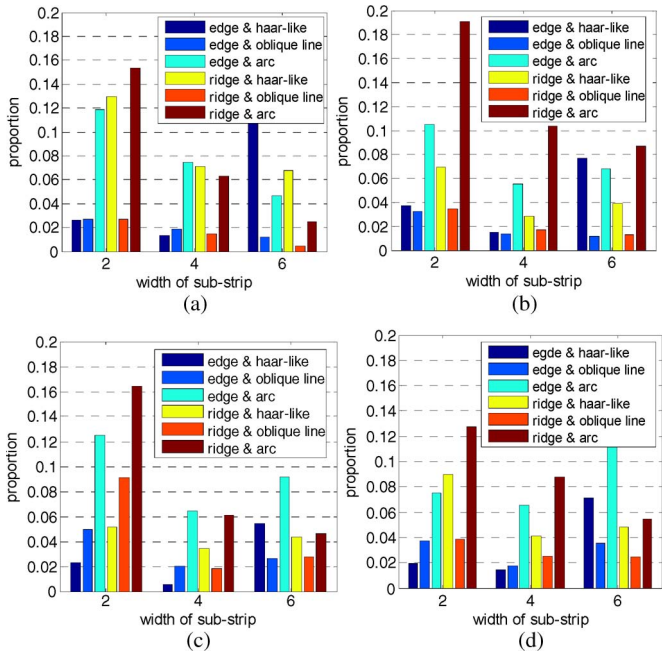


Fig. 10. Grouping selected strip features according to curve patterns and widths of substrips. (a) UIUC side-view car. (b) VOC 2006 car. (c) VOC 2006 bicycle. (d) VOC 2006 person.

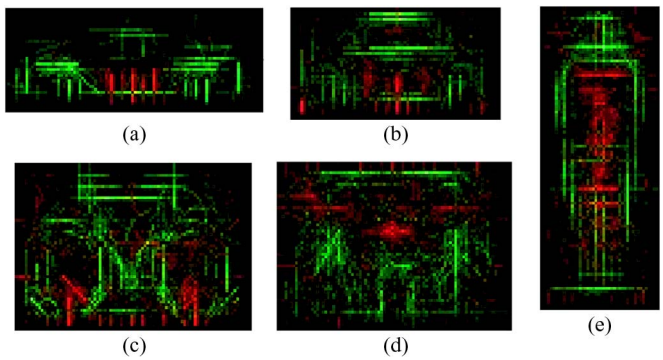


Fig. 11. Curve patterns of selected features delineate object structures. (a) Side-view car. (b) Multiview car. (c) Bicycle. (d) Cow. (e) Person.

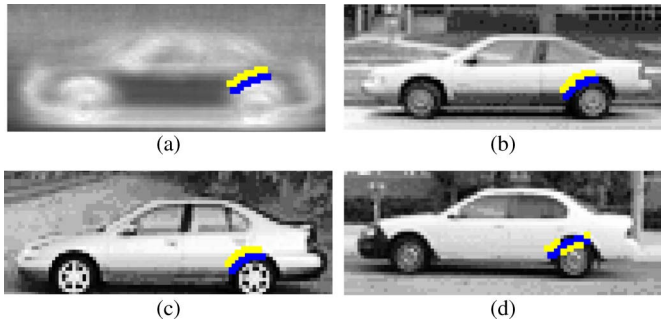


Fig. 12. Top-one selected perturbed strip feature and its perturbations on different samples.

structures on the belly; thus, the detection window will be classified as negative if the response of the first feature is large.

We also group the selected strip features according to the curve patterns and widths of the substrips. We collect the IStrip features by the complexity-aware RealBoost algorithm and count the numbers of different types, including edgelike, ridgelike, Haar-like, oblique lines, and arcs. We select four object classes and show the distribution of feature types in Fig. 10. It can be seen that a large portion of the selected features is the oblique lines and arcs. Apparently, non-Haar-like strip features play very important roles in classifying objects and nonobjects. Another interesting phenomenon is that the ridges of 2-pixel-width substrip region $(w_R = 2)$ tend to be selected. We take the side-view car in Fig. 9 as an example. We resize the car images to 100×40 pixels and find that a lot of shape elements, such as tires and bumpers, appear as line or arc ridges with $w_R = 2$. Likewise, the structures of other object classes also contain many patterns of 2-pixel-width ridgelike strips. We find that many 2-pixel-width edgelike arcs are selected for the man-made object classes, whereas many 6-pixel-width edgelike arcs are selected for the natural objects. Since the man-made object samples can be well aligned, the selected strip features can capture more detailed structures of fine resolution. However, the natural object classes cannot be well aligned; thus, most of the selected strip features can only describe the object structure of coarse resolution.

In Fig. 11, we overlay the curve patterns of all the selected IStrip features on a detection window. We weight the selected strip features using the log-likelihood ratio between positive and negative samples, i.e., the features with positive weights describe the positive patterns and vice versa. The features with positive weight are shown in green, whereas the negative ones are shown in red. We normalize the feature weight to [0, 255] for positive and negative weights, respectively. Obviously, the green features locate at the typical structures of the objects. There are also many red features that capture the negative patterns. For example, the second feature of the car in Fig. 9(b) and the first feature of the cow in Fig. 9(d) are such negative features. From the given analysis, we can see that strip features are capable of describing the shape characteristics of objects explicitly.

D. Perturbed Strip Features

In this section, we take the UIUC side-view car as an example to show the effectiveness of the perturbed strip features.

We adopt the RealBoost algorithm to select the perturbed strip features based on the 550 positive samples and 500 negative samples in the UIUC training set. We show that the first selected feature in Fig. 12(a) and its different perturbed version on some samples in Fig. 12(b)–(d). In Fig. 12, the blue regions correspond to the dark regions, and the yellow regions correspond to the bright regions. We can see that the contrast patterns and positions of the associated rectangle pairs may be different on different samples.

In Fig. 12, we can see that the wheels of different cars are not well aligned. Although the strip feature locates near the wheel, it cannot precisely capture the wheel structure. The associated rectangles can move in a neighborhood; thus, the perturbed strip features can fit to the precise location of the wheel according to (9) and (10). As shown in Fig. 12(b) and (c), the parameters of strip features are finally tuned to optimum that can capture the wheel structure more precisely. As shown in

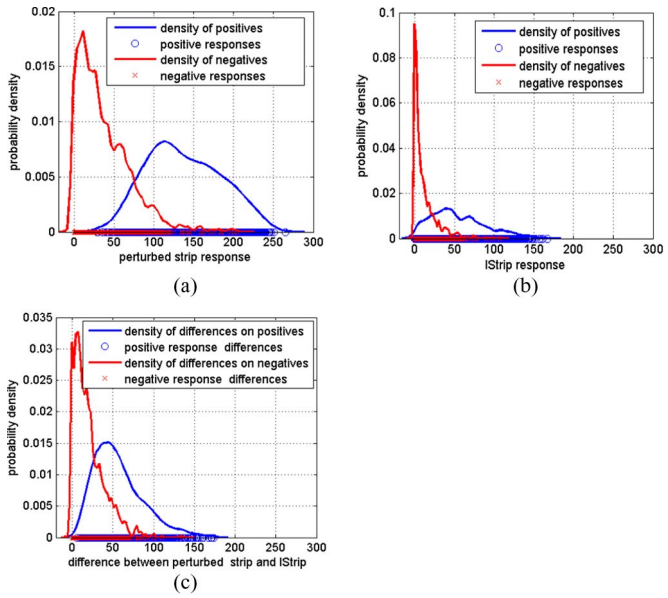


Fig. 13. Comparing distributions of feature responses on positive and negative samples. (a) Distributions of perturbed strip feature responses. (b) Distributions of IStrip feature responses. (c) Distributions of feature response differences.

Fig. 12(d), the contrast patterns of the associated rectangle pairs may flip during the perturbation so that the substrip regions may be inconsistently bright or dark. Therefore, the perturbed strip features may capture the discontinuous edges in the real-scene images. We also study the Z value [see (12)] for analyzing the discriminability of the perturbed strip features. The Z value is in the range of [0, 2], and a smaller Z value corresponds to powerful discriminability. We evaluate the Z value for the feature in Fig. 12. If we calculate the feature response according to the IStrip approach, the Z value is 1.304. If we calculate the feature response according to the proposed perturbation algorithm, the Z value is 0.886. The perturbed strip feature is more discriminative than the corresponding IStrip feature since the perturbation algorithm is robust to discontinuous edges and slight misalignment. Therefore, the perturbed version may be more robust for describing shapes than the IStrip approach.

We calculate the top selected feature (see Fig. 12) according to the IStrip approach and the proposed perturbation algorithm, respectively. Then, we use kernel density estimation to estimate the distribution of the feature responses and show the distributions in Fig. 13. In Fig. 13(b), we can see that the feature responses of many positive samples (cars) are smaller than 50 for the IStrip approach. The reason may be that the feature is misaligned with the actual contours. The perturbation algorithm can alleviate the misalignment to some extent; thus, most of the responses in Fig. 13(a) are larger than 50. We calculate the response difference between the IStrip approach and the perturbation algorithm for each sample. The distribution of the response difference is shown in Fig. 13(c). It can be seen that the differences of positive samples are larger than that of negative samples. It means that the perturbation algorithm plays a more important role for the positive samples compared with the negative samples. For the positive samples, the strip feature is close to the actual boundaries, and the perturbed version is likely to match the actual boundaries. For the negative samples, the strip feature is far from the actual boundaries, and the perturbed version may not match boundaries. From the end-to-end comparison in Fig. 8, we can see that the perturbed strip features can improve the final performance of the object detectors.

E. Complexity-Aware Criterion and Runtime

We evaluate the proposed complexity-aware-criterion-based algorithm using the UIUC single-scale data set described in Section V-A. Haar-like and edgelet features are used for comparison. The detectors perform the exhaustive search with a 1-pixel step on images. In this case, the detector processes overall 1 354 427 sliding windows on the 170 images.

We take the IStrip approach as an example and examine the feature selection criterion in the training process. We visualize the selected features in Fig. 14(a), in which all the Haar-like features are under the black line.[1] It shows that the first 18 features selected by the complexity-aware-criterion-based algorithm are all Haar-like features. On the contrary, the first 18 features selected by the RealBoost contain many non-Haar-like features. The result shows that the complex-aware RealBoost tends to select simple features in the earlier stages of the cascaded classifier, and such earlier stages dominate the total runtime. Considering the performance presented in Table III, we can see that our approach is more efficient than the RealBoost algorithm while preserving the accuracy.

We compare the proposed criterion with the criterion defined as the quotient of the discrimination ability and computation cost [22]. The selected features are shown in Fig. 14(a). It can be seen that the selected features by the quotient criterion are all Haar-like features, whereas the proposed criterion selects many complex features in addition to Haar-like features since it gives very small weights to the computation costs for the latter strong classifiers. We also compare these two classifiers in Fig. 14(b), and it can be seen that both of the two classifiers are very fast. Then, we test the classifiers on the UIUC single-scale data set. We find that the detection EPR (96.5%) of the proposed criterion is higher than the detection EPR (93%) of the quotient criterion. Compared with the quotient criterion, the proposed approach is more suitable for balancing the discriminability and efficiency for the cascaded object detector.

Fig. 14(b) shows the average runtime of the first n strong classifiers of each cascaded classifier. It can be seen that the DStrip-and-RealBoost-based approach is slower than the other fast approaches, because some features with large w_R appear in the first several stages of the cascaded classifier and slow down the speed. The IStrip-and-complexity-aware-based approach is as efficient as the Haar-like-based one, and is about 20% faster than the edgelet-based one. The total runtime of all the 170 test images is 1.26 s (about 7.4 ms per image). Although the perturbed strip features are theoretically nine times slower than the corresponding strip features, the detection systems based on perturbed strip features are about 2–3 times slower than that based on IStrip features. The reason is that the perturbed

[1]Each feature is represented by its computation cost, i.e., parameter C described in Table I. Since Haar-like features have $C \leq 17$ and the complex strip features have $C > 17$, we can easily distinguish them by the black line.

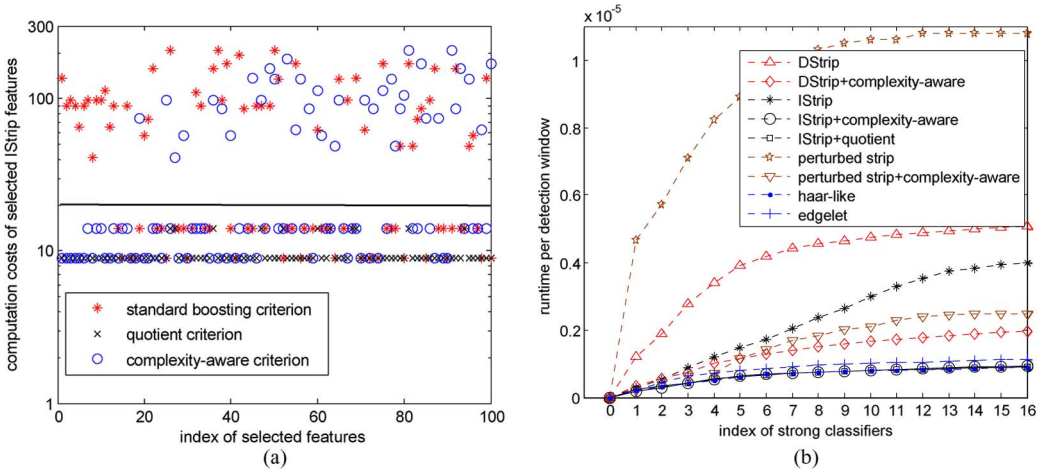


Fig. 14. Comparing computation cost and detection speeds for different features and different feature selection criteria. (a) Computation costs of the features that are selected by different feature selection criteria. (b) Comparing runtimes of detectors that are based on different features and different feature selection criteria.

strip features are more discriminative than strip features; thus, there are fewer perturbed features in the cascaded classifier. In practice, the detection system based on the perturbed strip features and complexity-aware criterion costs about 19.7 ms per image.

In addition, we compare the proposed approach with two other approaches, namely HOG + SVM [4] and DPM [10], on the UIUC single-scale car data set. For these two methods, we use the source codes published by the authors. The total runtimes of the 170 test images are 482 s for HOG + SVM, 153 s for DPM,and 1.26 s for the proposed approach. Although we do not use any code optimization, the proposed approach is still much faster than the other two methods. We also compare the EPR rates in Table III. It can be seen that the proposed approach achieves higher accuracy than the other two approaches. Therefore, the proposed approach may be more effective and efficient than the complex approaches when the target (e.g., side-view cars) have typical and stable shapes. Finally, we give some detection results of our approach on the public datasets in Fig. 15.

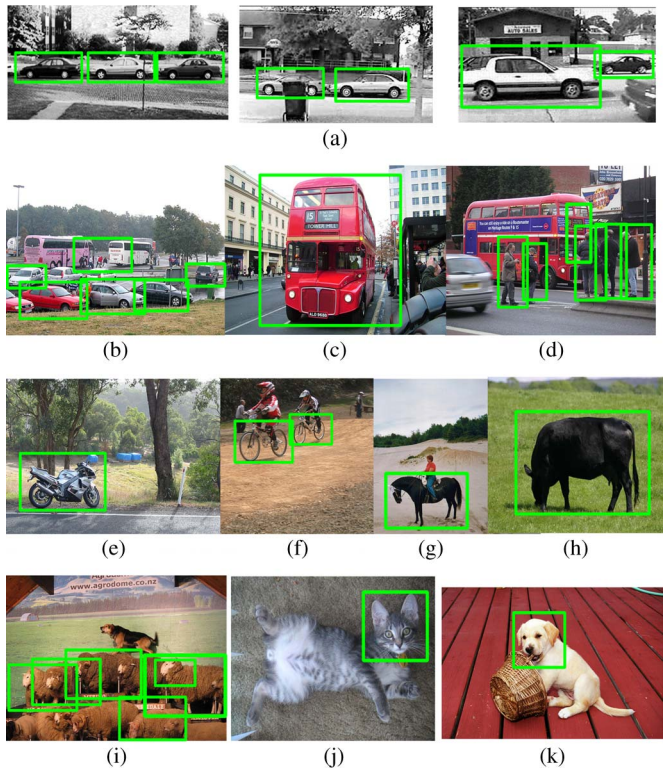


Fig. 15. Example detection results of public data sets. (a) Side-view car. (b) Car. (c) Bus. (d) Person. (e) Motorbike. (f) Bicycle. (g) Horse. (h) Cow. (i) Sheep. (j) Cat. (k) Dog.

VI. Conclusion and Discussion

In this paper, we have proposed a set of strip features for object detection. We propose two efficient approaches, namely IStrip and DStrip, to calculate the responses of strip features. Furthermore, we propose a deformable version based on the IStrip approach, namely perturbed strip feature, to improve the robustness and discriminability to shape variance. The complexity-aware criterion can reevaluate the computation cost of features according to the cascaded classifier and adaptively balances the discriminability and computation cost for feature selection. Experimental results have shown the effectiveness and efficiency of the proposed approach on the public data sets.

Strip features represent higher level semantic information than Haar-like features. In some sense, the IStrip features can be considered as a subset of the joint Haar-like features constrained by some curve patterns. Therefore, the proposed features have more powerful discriminability than Haar-like features. Furthermore, by designing different curve patterns, strip features can be tuned to describe various shape characteristics of different objects. Compared with the complex local descriptors such as HOG [4] and covariance descriptors [20], strip features can explicitly capture the shape information shown in Fig. 11. Many complex features are based on the statistics of regions; thus, they may be robust to slight variance of translation, scale, and rotation. Strip features discard some statistical information, which weakens their discriminability. However, it seems the inevitable cost of fast features. Compared with those complex features, strip features are more effective and efficient for detecting the objects with simple and typical shapes, such as side-view cars.

The proposed perturbed version of strip features is a simple trial for deforming strip features, and other effective deformation approaches [13], [15] may be utilized to improve the deformation capability of strip features. Moreover, the performance can be further improved by incorporating with other powerful features [27], [30]. As a kind of local features, strip features are also promising in designing part-based object detection systems for handling the occlusion problem. These topics will be studied in the future.

REFERENCES

[1] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1475–1490, Nov. 2004.

[2] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[3] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, pp. 886–893.

[5] N. Dalal, "Finding people in images and videos," Inst. Nat. Polytech. de Grenoble, Grenoble, France, 2006.

[6] M. Everingham, A. Zisserman, C. Williams, and L. V. Gool, "The Pascal visual object classes challenge 2006 (VOC2006) results," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[7] W. Chang and C. Cho, "Online boosting for vehicle detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 892–902, Jun. 2010.

[8] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 259–289, May 2008.

[9] M. Fritz, B. Leibe, B. Caputo, and B. Schiele, "Integrating representative and discriminative models for object category detection," in *Proc. ICCV*, 2005, pp. 1363–1370.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[11] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. CVPR*, 2003, pp. 264–271.

[12] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool, "Towards multi-view object class detection," in *Proc. CVPR*, 2006, pp. 1589–1596.

[13] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection," *Int. J. Comput. Vis.*, vol. 87, no. 3, pp. 284–303, May 2010.

[14] W. Gao, H. Ai, and S. Lao, "Adaptive contour features in oriented granular space for human detection and segmentation," in *Proc. CVPR*, 2009, pp. 1786–1793.

[15] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.

[18] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *Proc. CVPR*, 2006, pp. 11–18.

[19] T. Ma and L. Latecki, "From partial shape matching through local deformation to robust global shape similarity for object detection," in *Proc. CVPR*, 2011, pp. 1441–1448.

[20] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.

[21] H. Ren, C. Heng, W. Zheng, L. Liang, and X. Chen, "Fast object detection using boosted co-occurrence histograms of oriented gradients," in *Proc. ICIP*, 2010, pp. 2705–2708.

[22] J. Shotton, A. Blake, and R. Cipolla, "Efficiently combining contour and texture cues for object recognition," in *Proc. BMVC*, 2008, pp. 7.1–7.10.

[23] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, Dec. 1999.

[24] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1270–1281, Jul. 2008.

[25] P. Srinivasan, Q. Zhu, and J. Shi, "Many-to-one contour matching for describing and discriminating object shape," in *Proc. CVPR*, 2010, pp. 1673–1680.

[26] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511–518.

[27] B. Wu and R. Nevatia, "Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection," in *Proc. CVPR*, 2008, pp. 1–8.

[28] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *Proc. ICCV*, 2005, pp. 90–97.

[29] C. Waring and X. Liu, "Face detection using spectral histograms and SVMS," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 3, pp. 467–476, Jun. 2005.

[30] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. ICCV*, 2009, pp. 32–39.

[31] Y. Wu, Z. Si, C. Fleming, and S. Zhu, "Deformable template as active basis," in *Proc. ICCV*, 2007, pp. 1–8.

[32] Y. Xu, X. Cao, and H. Qiao, "An efficient tree classifier ensemble-based approach for pedestrian detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 107–117, Feb. 2011.

[33] S. Yan, S. Shan, X. Chen, and W. Gao, "Locally assembled binary (LAB) feature with feature-centric cascade for fast and accurate face detection," in *Proc. CVPR*, 2008, pp. 1–7.

[34] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. CVPR*, 2006, pp. 1491–1498.

[35] W. Zheng and L. Liang, "Fast car detection using image strip features," in *Proc. CVPR*, 2009, pp. 2703–2710.

[36] Z. Zhu, Y. Zhao, and H. Lu, "Sequential architecture for efficient car detection," in *Proc. CVPR*, 2007, pp. 1–8.

[37] G. Casalino, N. Del Buono, and M. Minervini, "Nonnegative matrix factorizations performing object detection and localization," *Appl. Comput. Intell. Soft Comput.*, vol. 2012, Jan. 2012, Article ID 781987, 19 pages, [Online]. Available: http://www.hindawi.com/journals/acisc/2012/781987/cta/

[38] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, Winter 1991.

[39] W. Liu and N. Zheng, "Non-negative matrix factorization based methods for object recognition," *Pattern Recognit. Lett.*, vol. 25, no. 8, pp. 893–897, Jun. 2003.

[40] J. Yang, D. Zhang, and J. Yang, "Constructing PCA baseline algorithms to reevaluate ICA-based face recognition performance," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 4, pp. 1015–1021, Aug. 2007.

[41] X. Chen, L. Gu, S. Z. Li, and H. J. Zhang, "Learning representative local features for face detection," in *Proc. CVPR*, 2001, pp. 1126–1131.

Wei Zheng received the Bachelor's degree from Tsinghua University, Beijing, China, in 2006. He is currently working toward the Ph.D. degree with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing.

In July 2013, he will be with Beijing Samsung Telecom R&D Center, Beijing. His research interests include object recognition, object tracking, and shape representation.

Hong Chang received the Bachelor's degree from Hebei University of Technology, Tianjin, China, in 1998; the M.S. degree from Tianjin University, Tianjin, in 2001; and the Ph.D. degree from Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2006, all in computer science.

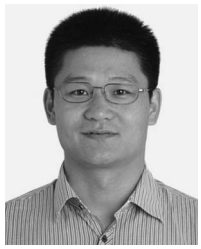
She was a Research Scientist with Xerox Research Centre Europe. She is currently an Associate Researcher with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her main research interests include algorithms and models in machine learning, and their applications in pattern recognition, computer vision, and data mining.

Luhong Liang (M'01) received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1997 and 2001, respectively, all in computer science.

From 2001 to 2007, he worked as a Senior Researcher and a Staff Researcher with Intel China Research Center. From 2008 to 2011, he was an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. From 2009 to 2011, he was also a Manager with the Codec Research Group, National Engineering Laboratory for Digital Video Technology. He is currently a Principal Engineer with the IC Design Group, Hong Kong Applied Science and Technology Research Institute, Hong Kong. He is the author of more than 30 publications in conferences and journals and a holder of ten U.S. patents and patent applications. His current research interests include image and video enhancement, video coding, video surveillance, and video processing systems.

Haoyu Ren received the B.S. degree from Tsinghua University, Beijing, China, in 2007 and the M.S. degree from the Institute of Computing Technology, Chinese Academy of Science, Beijing, in 2010.

He is currently a Researcher with the Beijing Samsung Research Center, Beijing, His research interests include object detection and tracking, segmentation, and pattern recognition.



Shiguang Shan (M'04) received the M.S. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1999 and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences, Beijing, China, in 2004.

Since 2002, he has been with ICT, where is currently a Full Professor. He is the author of more than 120 papers in refereed journals and proceedings in the areas of computer vision, pattern recognition, and image processing. His research interests include face-recognition-related research topics.

Dr. Shan has served as an Area Chair for a number of international conferences, such as the 13th International Conference on Computer Vision, the 21st International Conference on Pattern Recognition, and the 11th Asian Conference on Computer Vision. He has served on the Editorial Board of *Neurocomputing* since 2012. He was a recipient of the China's State Scientific and Technological Progress Awards in 2005 for his work on face recognition technologies, the Best Student Poster Award Runner-up at the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, and the Silver Medal of Scopus Future Star of Science Award in 2009.

Xilin Chen (M'00–SM'09) received the B.S., M.S., and Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 1988, 1991, and 1994, respectively, all in computer science.

From 1999 to 2005, he was a Professor with the Harbin Institute of Technology. From 2001 to 2004, he was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA. Since August 2004, he has been with the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, China. He is the Director of the Key Laboratory of Intelligent Information Processing, CAS. He is the author or coauthor of one book and over 200 papers in refereed journals and proceedings in the areas of computer vision, pattern recognition, image processing, and multimodal interfaces.

Dr. Chen has served as a program committee member for more than 30 conferences. He is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the *Chinese Journal of Computers*, and an Area Editor for the *Journal of Computer Science and Technology.* He was a recipient of several awards, including the China's State Scientific and Technological Progress Award in 2000, 2003, and 2005, for his research work.