Object Detection via Structural Feature Selection and Shape Model

Huigang Zhang, Xiao Bai, Jun Zhou, Senior Member, IEEE, Jian Cheng and Huijie Zhao

Abstract

In this paper, we propose an approach for object detection via structural feature selection and part-based shape model. It automatically learns a shape model from cluttered training images without need to explicitly use bounding boxes on objects. Our approach first builds a class-specific codebook of local contour features, and then generates structural feature descriptors by combining context shape information. These descriptors are robust to both within-class variations and scale changes. By exploring pairwise image matching using fast earth mover’s distance, feature weights can be iteratively updated. Those discriminative foreground features will be assigned high weights and then be selected to build a part-based shape model. Finally, object detection is performed by matching each testing image with this model. Experiments show that the proposed method with feature selection and shape model is very effective. It has achieved comparable performance to the state-of-the-art shape-based detection methods, but requires much less training information.

Index Terms

Object detection, Foreground feature selection, Part-based shape model.

I. INTRODUCTION

Object detection is a challenging task in computer vision and pattern recognition. Recent work in this area can be divided into appearance-based and contour-based approaches [1]. Appearance-
based approaches detect interesting points and extract their patch descriptors using features such as color, texture and gradient [2], [3], [4]. Contour-based methods, in contrast, study relationship between contour points and shape formed by them [5], [6], [7].

Objects vary greatly in appearance, not only being influenced by changes in illumination and viewpoint, but also due to intra-class variability in shape and other visual properties. When appearance-based approaches are utilized, intra-class variation problem cannot be easily solved, despite the fact that various feature extraction methods and structural descriptors have been developed. On the other hand, psychophysical studies show that human can recognize objects using fragments of outline contour alone [8]. Compared with other features, contour information is invariant to color, texture, and brightness changes. Moreover, it enables significant reduction in the number of training examples required for model development, without loss of the detection accuracy [9].

During the past decade, several contour-based methods have been proposed and demonstrated excellent performance in object detection and recognition [10], [11], [12]. Shotton and Opelt et al. [5], [13] proposed object recognition frameworks based on contour fragments codebook learning and clutter sensitive chamfer matching. Extending the contour based method, Shotton et al. introduced a bootstrapping technique that augments the sparse set of training examples [8]. In [14], Ferrari et al. built a codebook of pairs of adjacent segments (PAS) from cropped training images, and localized objects in the form of shape boundary. Zhu et al. [9] introduced a contour context selection framework for detecting objects in cluttered images using only one exemplar per-class. Lu et al. [15] used particle filters as inference tool for relevant contour fragments selection from edge images in order to match object models. More recently, sophisticated training process is designed to ensure models be automatically learned from training data instead of using a single category shape prototype [16]. Besides these approaches, contour information is also utilized for various object recognition tasks [17], [18], [19].

Most of these methods use ground truth bounding boxes during the training stage in order to extract features only from the objects themselves while removing the influence of background. This makes the training process inconvenient and less automatic as the bounding boxes are normally manually labeled. Furthermore, local contour features used in these methods are often too generic. They are easily matched to irrelevant parts of an image, while the geometric relationship between different parts of objects are often not quantitatively evaluated. To overcome
these problems, we propose a method for object detection via structural feature selection and part-based shape model. This method commences from building a class specific codebook of local contour features, and then generates feature descriptors by combining context information on object shape. These new descriptors are robust to within-class variations and scale changes. By exploring pairwise image matching via earth mover’s distance, discriminative foreground features can be selected after an iterative feature weight update process. Those features selected from the positive training images are then used to build a part-based shape model. Finally, object detection is performed by matching each testing image with this model. An example on how this method works is shown in Fig. 1.

Fig. 1. Illustration of the proposed method. (a) Training images with raw feature points. (b) Selected foreground features. (c) Part-based shape model for object detection.

The contribution of this framework lies in three aspects. Firstly, a new contour context descriptor is developed, which combines PAS feature [20] and its context information for shape part description. This descriptor is proved to be robust to within-class variation and scale changes, and is suitable for feature selection task. Secondly, we propose a method to eliminate the background responses for object boundary model learning without using ground truth bounding boxes. This is a practical solution to get a prediction model when abundant training information does not exist. Finally, a simple deformable shape model for object identification is proposed. The effectiveness of this framework is validated through the experiments on three widely used datasets: the ETHZ shape dataset, the INRIA horse dataset, and the PASCAL 2007 dataset. This paper extends from an earlier conference version in [21]. Comparing with the conference version, this paper has given complete and detailed descriptions on the proposed method with in-depth analysis on why each step is required and effective. Several extensions have been introduced to
the method. These include 1) introduction of PAS context feature augmentation process so as to extract more effective foreground features; 2) introduction of a new descriptor that treats each feature as an independent component for more precise representation of objects; and 3) adoption of a sparse coding scheme for vectorized object representation instead of using bag-of-words approach as in [21]. Furthermore, we have significantly extended the experiments to demonstrate the effectiveness of each step of the proposed method.

The rest of the paper is organized as follows. After reviewing the related work in Section II, we present the details of the proposed detection approach in Section III. Section IV reports extensive experimental results to validate the effectiveness of each key step of the method. Finally, we conclude the paper in Section V.

II. RELATED WORK

Shape features have played an important role in object detection. For example, many recent works use locally-normalized histogram of gradient (HOG) [22] to characterize rich object structural information. In many cases, low-level shape features, especially contour features, are needed to construct a shape model. In [23], Belongie et al. proposed a widely used descriptor, the *shape context*, to solve the shape matching problem. This is a globally discriminative feature extraction method that captures not only local object property but also the distribution of the relationship between pairs of contour points. This idea has opened up a new vista to combine local feature with its related surrounding features. In [14], Ferrari et al. presented a new local contour feature, namely *Pairs of Adjacent Segments (PAS)*, for objects detection and localization. PAS is composed of small connected pairwise points, i.e., approximately straight contour segments. It is capable of covering more accurate portion of object boundary without including nearby spurious edges, therefore, has demonstrated good performance in object detection.

Besides developing new features, some methods focus on selecting only a small subset of the training features for object recognition. Turcot et al. [24] presented a method for identification of maximally robust features in a set of images, as well as a method for image feature augmentation in a bag-of-words framework. In this work, a graph is introduced to represent matching relationships between images, which effectively augments the feature set for each image by merging them with maximally robust features from neighboring images. Kim et al. [25] introduced a pairwise image matching method to select discriminative foreground features. Pairwise image
matching scores are calculated to help update the weights of discriminative features via an iterative matching process. The final selected features correspond to the foreground content of the images, which characterize the high-level knowledge of image categories.

After feature extraction and selection, object detection models are required. Deformable part models, such as pictorial structures, provide an elegant framework for this purpose [26]. Pictorial structure model [27] defines a matching problem in which parts have corresponding matching costs in a dense set of locations, while their geometric arrangements are captured by a set of connecting part pairs. For example, parts at different locations can be arranged in a star-structured model with an appended root component [28]. Generative part models form another group of effective object detection models built from parameter estimation methods. Amit et al. formulated a deformable template model for object recognition with a clearly defined mechanism for parameter estimation [29]. The deformation of an object is defined in terms of locations of a moderate number of reference points. Bai et al. treated object parts as nodes of a graph and constructed a generative model for graph structures through the embedding of the nodes of the graph in a vector space [30].

For image representation, the transformation and integration of features is the bond that links feature extraction and object detection. Bag-of-words (BOW) model has been widely used for this purpose because of its simplicity and effectiveness [31], [32], [33], [34]. This method represents an image by a histogram of local features. It is robust against spatial translations of features due to its statistical nature and has demonstrated decent performance in image classification tasks. Beyond BOW, Yang et al. [35] used a sparse coding (SC) method to derive image representations. By combining the state-of-the-art spatial pyramid matching (SPM) [36] method and max pooling, this method has achieved high recognition accuracy. Compared with BOW, SC can achieve a much lower reconstruction error due to the less restrictive constraints in model construction.

III. APPROACH

The goal of the proposed method is to detect objects in images. Given a set of training images with cluttered background, our method first extracts structural features from images. The proposed feature descriptor is an extension of the Pairs of Adjacent Segments (PAS) by adding context information to characterize structural information of object parts (Section III-A). Initially, these feature descriptors are assigned equal weights. Then the descriptors and their
weights are updated by similarity measurement and feature update steps. The similarities are measured between image pairs of the same class in the training set, and are calculated using the fast and robust Earth Mover’s Distances method (fast EMD) [37]. The feature update step then uses the similarity and the matching property to update the weights of each feature descriptor. The similarity calculation and feature update steps run iteratively, which allows the features that are common to a class be assigned large weights, and be selected as the final foreground descriptors (Section III-B). The selected features from the positive training images are then used to form a part-based shape model (Section III-C). In this model, both whole object and parts information are captured and represented using a sparse coding scheme. The final object detection task is performed by matching each testing image with this model (Section III-D). The complete object detection framework is illustrated in Fig. 2. The image in the second panel represents the part-based shape model, with an enlarged version given in Fig. 7.

**Fig. 2.** Framework of the proposed object detection method.

### A. Structural Feature Description

The feature extraction step is based on PAS because of its scale-invariant local property and good performance in object detection [14]. Although being resilient under occlusion and clutters, PAS sometimes suffers from too much locality because features with minimal spatial extent may be too generic and easily matched to anything. To alleviate this problem, we propose a PAS
context descriptor which encodes both local PAS descriptor and the relative locations of other
features in a spatial neighborhood. Therefore, this descriptor combines both local description
ability of PAS and local structure, i.e. context information, around each feature. We describe
how to construct this descriptor in detail as follows.

Firstly, PAS features are extracted from the training set as described in [14]. A PAS feature is
computed over a pair of connected contour segments, which can be represented as a quintuple
\((x, y, s, e, d)\). In this representation, \((x, y)\) is the mean location of the segment pair, \(s\) is the
distance between the segment centers, and \(e\) is a value in \([0, 1]\) which measures the average
edge detection confidence. A descriptor \(d = (\theta_1, \theta_2, l_1, l_2, r)\) is defined to indicate the shape of
the PAS through encoding the orientations \(\theta_1, \theta_2\) and lengths \(l_1, l_2\) of the two contour segments.
\(r\) is a relative location vector, which points from the center of one segment to the center of
another.

Secondly, we construct a codebook via clustering the extracted PAS features [14]. The code-
book is useful for efficient matching since all features similar to a certain type are considered as
a group. As the PAS codebook is class-specific, it has to be built separately for different classes.
In our experiments, we set the codebook dimension to 40.

After the codebook generation, for each image, we construct log-polar coordinates centered
at the location where each PAS feature is extracted. This step forms five circles as shown in
Fig. 3(a). The radiuses of these five circles are used to indicate the distance between pixels to
the PAS feature location, which are set to 0.125, 0.25, 0.5, 1, and 2, respectively. They divide
the neighborhood of a feature location into five regions. For each region, we assign a weight \(\alpha\)
to it using the Gaussian function

\[
\alpha = \sqrt{2\pi} \exp\left(-\frac{r^2}{2}\right)
\]

where \(r\) is the radius of the corresponding circle. In this way, the weights are set to 2.49, 2.43,
2.21, 1.52, 0.34 from inside to outside, which are marked in Fig. 3(b).

Finally, we map all PAS features in the neighborhood to the closest codebook entry, and
extend the PAS features to context descriptors represented as \(f_p = [p_1, p_2, \ldots, p_{40}]^T\) (Fig. 3(c)),
where

\[
p_i = \sum_{j=1}^{5} (\alpha_j \times \#\{q \neq p : q \in \text{codebook}(i) \text{ and } q \in \text{region}(j)\})
\]
is the number of PAS features in region $j$ around $p$ belong to codebook entry $i$ multiplied by their location weight $\alpha$.

PAS is a robust scale invariant local feature. In order to keep this property with the proposed PAS context descriptor, relative distance is used in $q \in \text{region}(j)$. Following [23], the distance between $p$ and $q$ is normalized over the mean distance of the whole feature set. If the normalized distance is smaller than the radius of the $j^{th}$ circle but larger than that of the $j-1^{th}$ circle, we say $q \in \text{region}(j)$.

![Fig. 3. PAS context construction(a), computation(b) and representation(c).](image)

B. Foreground Feature Selection

In feature extraction step, images with background removed or with bounding boxes around target objects are normally preferred in order to get accurate object description. Here we propose an effective method for automatic foreground PAS context feature selection without requiring bounding boxes on training samples. The foreground feature selection rationale is based on the observation that images of the same category normally have high similarity because they share a set of common features. By calculating the pairwise image similarities in both intra- and inter-classes, we can update the weights of discriminative foreground features.

We have noticed that selected foreground features can help the extraction of the category specific information, which generates more accurate pairwise image similarities than using all foreground features. Suppose we have a set of images which contain foreground object from the same class. If we perform pairwise matching among these images, then the features from foreground regions will be matched to each other more frequently than those from the background region.
Based on these observations, in this paper, we propose an iteratively matching strategy to select the foreground features. Each feature is assigned an equal weight initially, then Earth Mover’s Distance (EMD) is used to perform pairwise matching between images. Based on the contribution of the feature towards the EMD calculation, the feature weights are updated. The weights for foreground features becomes larger as the iteration goes on, and those for the background features are gradually reduced, which facilitates explicit feature selection.

1) Pairwise matching via EMD: The first step of the feature selection method is pairwise image matching using fast EMD method [37]. This method is used because of two reasons. Firstly, EMD gives an optimal solution for matching between histograms with different dimensions. This exactly fits into the feature matching case in this paper, in which the number of feature points extracted from two images may not be the same. Secondly, the EMD algorithm considers both the relationship between feature descriptors as well as the weights associated with them, which has greatly facilitated the following feature selection step.

The fast EMD method is defined as follows. Let \( X = \{(f_1, w_1), (f_2, w_2), ..., (f_{|X|}, w_{|X|})\} \) be the representation of an image \( I \), where \( f_i \) is a PAS context feature, and \( w_i \geq 0 \) is the corresponding weight. \(|X|\) defines the number of features in this image. Initially, all feature weights are set to 1. Given representations \( X_p, X_q \) for images \( I_p, I_q \), the fast EMD is

\[
f_{EMD}(X_p, X_q) = \left( \min_{\{f_{ij}\}} \sum_{i,j} f_{ij}d_{ij} \right) / \left( \sum_{i,j} f_{ij} \right)
\]

s.t \( f_{ij} \geq 0 \quad \sum_j f_{ij} \leq w_i \quad \sum_i f_{ij} \leq w_j \)

\[
\sum_{i,j} f_{ij} = \min (\sum_i w_i, \sum_j w_j)
\]

where \( \{f_{ij}\} \) defines a flow matrix with the element \( f_{ij} \) indicating the flow between features \( f_i \in X_p \) and \( f_j \in X_q \). \( \{d_{ij}\} \) is a thresholded distance matrix with the element \( d_{ij} \) being defined as \( d_{ij} = \min(d(i, j), t) \), where \( d(i, j) \) is the Euclidean distance between features \( f_i \) and \( f_j \), \( t > 0 \) is a fast EMD parameter that controls the speed of the EMD computation process. The smaller the \( t \) is, the faster the process is.

2) Feature weights update: The second step is updating the feature weights according to the contribution of each feature towards the fast EMD calculation. Notice that the flow matrix
\{f_{ij}\} reflects the correspondence between features extracted from two images. We define the contribution of feature \(f_i\) to image \(I_q\) as

\[
    c_q(i) = \sum_j f_{ij} \times \delta_j / d_{ij}
\]

where

\[
    \delta_j = \frac{n \times w_j}{\sum_{k=1}^{n} w_k}
\]

\(n\) is the number of PAS context features in image \(I_q\). The mean of all related contributions in one category is used as the updated weight of feature \(f_i\). If \(N\) is the number of images in one category, the new weight of feature \(f_i\) in images \(I_p\) is

\[
    w_i = \frac{1}{N-1} \sum_{q=1}^{N-1} c_q(i)
\]

By doing so, we highlight the features which have strong matches across all images within the same category.

The above two steps are performed iteratively. We measure the differences of the average feature weights computed from all images between two consecutive iterations. If this difference is lower than 10\% of the previous weight, the iteration terminates. The PAS context features with high weights, e.g., larger than 1, are selected as the foreground contour features to build the shape model. Examples of the selected features are shown in (Fig. 1(b)). Note that ground truth bounding boxes are not required here. The above two steps form an iterative optimized process which increases the weights of features shared by intra-class targets. Meanwhile, background features will be given low weights because they can not reflect category information, so their weights diminish over the matching process.

3) Discussion: In the feature selection process, the PAS context descriptors are not updated. To make the descriptors more robust and effective, we modify the proposed PAS context as follows:

\[
    p_i = \sum_{j=1}^{5} (\alpha_j \times \sum_{q \in \text{codebook}(i), q \in \text{region}(j)} w_q)
\]

where the difference to the method in [21] lies in the introduction of a feature weight \(w_q\). In this way, when the feature weights are updated after an iteration, the PAS context descriptor can also be updated. As a consequence, the contribution from weak features become lower in the feature
descriptor construction, which makes the foreground information representation more accurate. The effectiveness of this feature augmentation step is validated in the experiment section.

Another question arises on why circles are necessary in the construction of descriptors. To answer this question, we propose another descriptor $f_p = [p_1, p_2, ..., p_{40}]^T$ and compare it with the proposed PAS context. Here

$$p_i = \sum_{q \neq p, q \in \text{codebook}(i)} \alpha_q \times w_q$$

where $\alpha_q = \sqrt{2\pi} \exp(-r_q^2/2)$ and $r_q$ is the normalized distance between $q$ and $p$. This new descriptor treats each feature as an independent element instead of as the member of a group in circles. Therefore, each feature has a different distance weight $\alpha$ and a similarity weight $w$. Despite the initial impression that this precise representation should be more effective than the group counterpart, experiments show that it does not perform as well as the PAS context feature. The reason lies in that such descriptors are sensitive to outliers. When the features are updated in the two-step iterations, some false information may also be reinforced. We will further verify this point in the experiment section.

The proposed feature selection method is akin to feature space partition process in the multiple instance learning (MIL) approaches as both paradigms can detect representative foreground objects [38], [39]. The main difference is that in MIL, the learning goal is on modelling the distribution of both positive and negative instances so that they can be well-separated. In our method, we only concern the foreground features that are shared by object classes. We do not care about the distribution of negative instances. Furthermore, an advantage of our method is that class-specific feature representation can be built for multiple classes in the training stage. However, most MIL methods can only deals with binary classes, and their extension to multi-class are not straightforward.

C. Part-based Shape Model

Having selected the foreground features from the training images, a model is needed for object detection. Here we adopt a part-based shape model inspired by the structural method in [28]. Our method is composed of a root model that approximately covers an entire object and some fixed part models that cover fine parts of the object, as exemplified in Fig. 1(c). The root model can
be easily obtained by incorporating all foreground features, but construction of the part model consists of three steps.

Fig. 4. Foreground features after clustering based on their locations.

In the first step, representative parts have to be found from the training images of the same category. To do so, we use the location information of the descriptors in an image. For a training image with its foreground features selected, it can be represented as $M = \{f_1, f_2, ..., f_m\}$, where $m$ is the number of foreground features. We group the $m$ features into $k$ clusters according to their locations using the K-means clustering algorithm, as shown in Fig. 4. In this figure, we have assigned each cluster with a different color in order to highlight different parts of an object. The k-means clustering algorithm is executed for three times on each image to get multiple part candidates. The initial k centers of the data are randomly selected. Therefore, the k-means algorithms may generate different grouping results for each time. As a consequence, each image is divided into $3 \times k$ clustered parts with possible overlaps. Some part examples are displayed in Fig. 5.

The purpose of the second step is to quantize parts into vectorized representations. These parts are grouped into $K$ clusters using the appearance information, based on the intuition that these clusters should have semantic meanings such as head, neck, back, abdomen, and tail in the swan example. We use PAS context descriptor to represent appearance information and adopt a sparse coding method to assemble them instead of using the traditional BOW model. The idea of sparse coding is inspired by the research in image statistics which has revealed that image patches are sparse signals [35].
This method is expressed as follows:

$$
\min_{U,V} \sum_{m=1}^{M} \|x_m - u_m V\|^2 + \lambda |u_m| \tag{10}
$$

subject to $\|v_k\| \leq 1, \quad \forall k = 1, 2, \ldots, K$

where $X = [x_1, x_2, \ldots, x_M]^T \in \mathbb{R}^{M \times 40}$ is a 40 dimensional PAS context feature space, $V = [v_1, v_2, \ldots, v_K]^T$ are the $K$ cluster centers of the codebook to be found, and $U = [u_1, u_2, \ldots, u_M]^T$ are the cluster membership indicators. Here, $\| \cdot \|$ and $| \cdot |$ denote the L2-norm and the L1-norm of vectors, respectively. Normally, $K$ is larger than the dimension of the feature space. Therefore, in our experiment, we set $K = 50$, which is larger than the PAS context feature dimension 40.

One way to solve the above problem is to alternatingly optimize over $U$ or $V$ while fixing the other parameters. In our experiment, we first train a codebook $V$ from the training set. Then each $u_m$ corresponds to each feature $x_m$ can be solved in the coding phase. For any part or root represented by a set of PAS context descriptors, we can compute a single feature vector based on statistics on the descriptor codes. Similar to [35], we use max pooling to assemble
these descriptors into a single feature vector. After solving the above optimization problem, we have got a set of feature codings $U = [u_1, u_2, \ldots, u_N]^T$ to represent a part or root $z$, where $N$ is the number of descriptors in $z$. The \textit{max pooling} process transforms $z$ into such a vector $z = [z_1, z_2, \ldots, z_{50}]^T$, where each element $z_j$ is calculated as

$$z_j = \max\{|u_{1j}|, |u_{2j}|, \ldots, |u_{Nj}|\}$$  \hfill (11)

In this way, each candidate part can be represented as a histogram using the sparse coding method. By clustering them on their similarities, we can achieve the $k$ semantic clusters, which are illustrated in Fig. 6.

![Fig. 6. Examples of candidate part class after clustering using appearance information. Each row represents a class. Images with red bounding box are not discriminative enough for model construction.](image)

Thirdly, each semantic cluster is filtered using the location information to get more accurate part models. Although the above clusters contain parts of similar appearance, this purely appearance-based clustering does not take the relative location of the parts into account. Furthermore, some parts in the clusters may be too rough to help learning a part model, such as the
examples with red bounding box in Fig. 6. To solve this problem, auxiliary location information is used to filter these non-discriminative parts. We assign each part with a vector translated from the object centroid as illustrated by the red arrows in Fig. 1(c). The similarity of these vectors can be measured by their lengths and angles. Those parts with low similarity values will be eliminated. Contour parts within each cluster now agree both in appearance and location relative to the object centroid, as shown in Fig. 7.

![Part-based shape model](image)

**Fig. 7.** The part-based shape model. The big green bounding box represents the root model, and the green dot in the center of the panel is its centroid. $k$ part clusters around the centroid represent the part model of the image, with $k = 5$ in this example. Both the root and the part models are described using mean value of the sparse coding histograms, which are drawn in green and blue respectively. Blue arrows show the two-dimensional vectors, specifying an anchor position for part $i$ that is relative to the root position.

Up to now, we have got the roots and part semantic clusters of the object, as illustrated in Fig. 7. Each part and the root can be represented as a location center and a histogram calculated by the sparse coding method. The means of the clusters are used to represent the final model of root and parts. Thus, each part model $P_i$ is defined by a 2-tuple $(F_i, v_i)$, where $F_i$ is a 50-dimensional histogram descriptor for the $i^{th}$ part, and $v_i$ is a two-dimensional vector specifying an anchor position for part $i$ that is relative to the root position, $i = 1, 2, ..., k$ ($k$ is the number of semantic clusters). The whole part-based shape model is formally defined as $(F_0, P_1, ..., P_k)$ where $F_0$ is the descriptor for the root.
D. Object Detection

To detect objects in an unseen image, we first extract the PAS context descriptors. Our detection system scans the image using a sliding window, and traverses the testing images at 5 different scales (which are 0.5, 0.8, 1, 1.2, 1.5 times the mean shape model scales respectively). We compute an overall score for each root location according to the best possible placement of the parts according to the part model. By defining a threshold for each possible root location we can detect multiple objects in an image. We retain all the hypotheses whose costs are less than the detection threshold. For different part scales, we perform non-maxima suppression by retaining only the lowest cost hypothesis among the ones which had significant overlap. Given a testing image $T$, let $(F^t_0, P^t_1, ..., P^t_k)$ be its possible detection result at location $t$. An optimal match of the model to image $T$ is defined as

$$t^* = \arg \min_t \left( \sum_{i=0}^{k} d(F_i, F^t_i) + \sum_{i=1}^{k} d(v_i, v^t_i) \right)$$

where $d(\cdot, \cdot)$ is the Euclidean distance measuring the similarity. $F^t_i$ is a 50-dimensional histogram descriptor for the $i^{th}$ part in location $t$, and $v^t_i$ is a two-dimensional vector specifying an anchor position for part $i$ of location $t$ that is relative to the root position.

IV. Experiments

In order to validate the proposed method, we have performed experiments on three widely used datasets [40], [41], [42] as outlined in section IV-A. The results on the proposed PAS context descriptor for feature selection are presented in section IV-B. The performance of the EMD feature selection process is then assessed in section IV-C. Finally, we present the accuracy of the proposed detection method in section IV-D, and compare it with several contour-based recognition approaches.

All experiments are implemented on a desktop with an Intel Core 2 Duo 2.40-GHz processor. In the sparse coding step, we integrate the Matlab codes from Honglak Lee [43] and Yang [35] for the codebook generation and the coding steps.

A. Datasets

1) ETHZ shape dataset: The ETHZ shape dataset consists of 5 different object categories (applelogos, bottles, giraffes, mugs, swans) with 255 images in total [40]. The images contain
at least one and sometimes multiple instances of a class and have a large amount of background clutter. It is highly challenging because of the significant intra-class variations, scale changes, and many severely cluttered images, with objects occupying only a small fraction of the total image area.

2) **INRIA horse dataset:** The INRIA horse dataset contains 340 images, in which 170 images belong to the positive class showing at least one horse and 170 images do not contain any horses [41]. Horses may appear at several scales, and in cluttered backgrounds.

3) **PASCAL 2007 dataset:** The Pascal VOC 2007 dataset is more challenging than the above two datasets due to the great variation in appearance and pose of objects in the images [42]. This dataset contains objects from 20 classes. The training set consists of 2,501 images with 6,301 objects. The testing set contains 4,952 images with 12,032 objects.

In our experiments, all images are resized to less than $300 \times 300$ pixels with the aspect ratio remain unchanged. Because PAS is a scale invariant feature extraction method, it allows extraction of descriptors with high consistency under different image sizes.

**B. Feature Descriptor Evaluation**

In this experiment, we evaluate the proposed PAS context descriptor and its effectiveness in the foreground feature selection step. Three types of descriptors are proposed for the detection system. They are defined by equations 2, 8, and 9, respectively. Equation 2 defines the original descriptor that combines PAS with context information, in which circular regions are used in the calculation but without weight updating during the feature selection iterations. It is labeled as “descriptor 1” in the experiment. Equation 8 is the modified descriptor with weight updating during the feature selection iteration, which is labeled as “descriptor 2”. Similarly, feature defined by Equation 9 is labeled as “descriptor 3”. It treats each PAS feature as an independent element instead of using circular representation.

In order to compare these three descriptors and the original PAS descriptor, all images in the six object classes from the ETHZ and the INRIA datasets are used. Table I shows the average feature selection accuracies on the datasets. The accuracy is calculated from the percentage of the number of descriptors located in the area of a ground truth bounding box around the foreground object against the total number of descriptors.
TABLE I

COMPARISON OF DIFFERENT CONTOUR FEATURE DESCRIPTORS FOR FOREGROUND FEATURE SELECTION. (%)

<table>
<thead>
<tr>
<th>Category</th>
<th>PAS descriptor 1</th>
<th>descriptor 2</th>
<th>descriptor 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>applelogos</td>
<td>73</td>
<td>83</td>
<td>92</td>
</tr>
<tr>
<td>bottles</td>
<td>76</td>
<td>83</td>
<td>94</td>
</tr>
<tr>
<td>giraffes</td>
<td>84</td>
<td>88</td>
<td>96</td>
</tr>
<tr>
<td>mugs</td>
<td>78</td>
<td>86</td>
<td>94</td>
</tr>
<tr>
<td>swans</td>
<td>85</td>
<td>90</td>
<td>98</td>
</tr>
<tr>
<td>horses</td>
<td>79</td>
<td>87</td>
<td>93</td>
</tr>
</tbody>
</table>

It can be observed that the descriptor 2 achieves the best performance among all four descriptors, with more than 90% accuracy on all six categories and the highest 98% accuracy on swans class in particular. The PAS descriptor has the lowest accuracy due to its local property. In foreground feature selection, too much locality can be problematic because features with minimal spatial extent may be too generic and easily matched to anything. The other three descriptors alleviate this problem by encoding the descriptors and relative locations of other features in a spatial neighborhood.

Although descriptor 1 encodes the shape context information, it does not use the feedbacks during the feature selection iteration. Thus, it is less robust than descriptor 2. Table I also shows that descriptor 3 can achieve reasonable results while it uses both the context and the feedback information. However, it does not perform as well as descriptor 2. The reason is that this descriptor is too precise. When useful information is augmented, some false information may also be reinforced, which makes it sensitive to outliers.

In summary, both theory and experiments have confirmed the effectiveness of the proposed PAS context descriptor (descriptor 2). In the following sections, only this descriptor is used in the experiments.

C. Foreground Feature Selection Performance

Now we turn our attention to the performance of the proposed foreground feature selection process. We have performed experiments on all images in the six classes in the ETHZ and the INRIA datasets. These images are tested separately to get corresponding foreground information. We firstly extract the initial PAS context descriptors of each image, and then use the proposed
feature selection process to update them. Finally, we choose those descriptors whose weights are larger than the initial value 1 as the final selected foreground features. Fig. 8 shows the experimental results.

![Different Iteration Performance](image)

**Fig. 8**. Performance of the foreground feature selection process.

In Fig. 8, the horizontal axis is the number of iterations for the feature weights update. The vertical axis on the left of the graph indicates the percentage of foreground features among all image features, which is an average value from all six categories. The vertical axis on the right corresponds to the average time cost over the iteration number. Therefore, the blue line represents how many correct foreground features are selected when iteration number increases, while the red line shows the time cost change tendency with the growth of iteration number.

From Fig. 8, it can be seen that the foreground feature selection accuracy is not always increasing when the number of iterations increases. After six iterations, the foreground feature selection accuracy is stabilized at around 94%. Nevertheless, the time consumption keeps increasing linearly even after the number of iterations exceeds six. Hence, the update iteration can be terminated after a number of steps to guarantee both efficiency and accuracy of the feature selection step. Notice that the data in Fig. 8 show the average values, thus six is not a fixed number for all images being tested. Meanwhile, we observe that the point of termination has the following property, i.e., the average of the changes on the feature weights is lower than 10% of that from the previous iteration. Hence, in our next detection experiments, we stop the foreground
feature selection process when this condition is met. Some feature selection results are shown in Fig. 9, which demonstrate the effectiveness of the proposed feature selection process. We also show some bad examples of foreground feature selection in Fig. 10. In these examples, our method does not perform well mainly due to the significant intra-class variations and overlapping of multiple objects in a training image.

Fig. 9. Examples of foreground feature selection results on the INRIA horse dataset. The upper row lists the original feature extraction results while the lower row shows their corresponding feature selection results.

D. Object Detection

The performance of the object detection method is validated on all three datasets mentioned in Section IV-A. For the ETHZ and the INRIA datasets, we follow the same training and testing strategy in [14] for fair comparison. We randomly select half of the positive examples and an
equal number of negative examples for training, while all the remaining examples are used for testing.

Note that each image in the datasets has a ground truth bounding box $B_{gt}$. The proposed method can also generate a bounding-box $B_m$ around the detected objects. Therefore, the accuracy of the proposed object detection method can be evaluated by the extent of overlap between two bounding boxes. To do so, we use the intersection-over-union (IoU) ratio to measure their overlapping area, which can be calculated by $\frac{B_{gt} \cap B_m}{B_{gt} \cup B_m}$. A detection is counted as correct if the score of IoU is greater than $20\%$, which is the PASCAL $20\%$ criterion and is abbreviated as $20\%-IoU$. Similarly, the PASCAL $50\%$ criterion ($50\%-IoU$) can also be defined. We have used the PASCAL $50\%$ criterion for evaluation purpose.

In this experiment, we compare the proposed shape model, which is learned without ground truth bounding boxes, with several alternative methods that require bounding boxes [28], [15], [10], [20]. Fig. 11 shows the detection rate vs. false positive per image (FPPI/DR) curves for the methods based on the above criteria. FPPI/DR curve is a standard that measures the detection performance, just like the ROC curve measures classification performance. The detection rate is the number of correct detections divided by the total number of objects, while FPPI represents the average number of false positives per image. As different thresholds give different detection results, the values of DR and FPPI can vary substantially. In these panels, the proposed method has shown comparable performance as the alternatives.

Table II shows the comparison of the detection rates at 0.3/0.4 FPPI using the 50%-IoU criterion on the ETHZ and the INRIA datasets. The proposed method has achieved comparable performance as the other methods. Specifically, our method outperforms all other methods in the swan class and in the mean accuracy at 0.4 FPPI. It also achieves the second best performance in the mean accuracy at 0.3 FPPI, with a slight margin behind the best method in [10]. We have to point out here that our method does not require bounding boxes during the training stage. It can automatically select useful foreground features. However, all other methods under comparison require bounding boxes in the training stage to explicitly get the locations of the foreground objects for feature extraction. This is the main advantage of the proposed method. If ground truth bounding boxes are used with the proposed method, it is expected that better performance can be achieved. The reason is that the feature selection process cannot provide 100% accuracy. Thus, the selection outcome may contain features from background, and in turn,
generates negative influence to the detection accuracy.

To highlight the effectiveness of our method, we also list some detection examples in Fig. 12. Note that our method can achieve good detection results on all the six classes regardless of background variation.

To show how our method can handle more difficult datasets, we run experiments on the Pascal
TABLE II
Comparison of detection accuracy for 0.3/0.4 FPPI on ETHZ and INRIA shape classes.

<table>
<thead>
<tr>
<th></th>
<th>Apples logos</th>
<th>Bottles</th>
<th>Giraffes</th>
<th>Mugs</th>
<th>Swans</th>
<th>Horses</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>0.917/0.917</td>
<td>0.953/0.953</td>
<td>0.787/0.909</td>
<td>0.815/0.886</td>
<td>1/1</td>
<td>0.6/0.839</td>
<td>0.845/0.917</td>
</tr>
<tr>
<td>Felz. et al [28]</td>
<td>0.95/0.95</td>
<td>1/1</td>
<td>0.729/0.729</td>
<td>0.839/0.839</td>
<td>0.588/0.647</td>
<td>0.572/0.667</td>
<td>0.779/0.805</td>
</tr>
<tr>
<td>Lu et al [15]</td>
<td>0.9/0.9</td>
<td>0.792/0.792</td>
<td>0.734/0.77</td>
<td>0.813/0.833</td>
<td>0.938/0.938</td>
<td>0.641/0.796</td>
<td>0.803/0.841</td>
</tr>
<tr>
<td>Maji et al [10]</td>
<td>0.95/0.95</td>
<td>0.929/0.964</td>
<td><strong>0.896/0.896</strong></td>
<td><strong>0.936/0.967</strong></td>
<td>0.882/0.882</td>
<td><strong>0.765/0.833</strong></td>
<td><strong>0.893/0.915</strong></td>
</tr>
<tr>
<td>Ferrari et al [20]</td>
<td>0.777/0.832</td>
<td>0.798/0.816</td>
<td>0.399/0.445</td>
<td>0.751/0.8</td>
<td>0.632/0.705</td>
<td>0.587/0.721</td>
<td>0.657/0.72</td>
</tr>
</tbody>
</table>

VOC 2007 dataset. The proposed method is compared with the maximum and median results of Pascal VOC 2007 [42]. To maintain consistency with PASCAL detection criterion, we used Average Precision (AP) with an overlap threshold of 50% to measure the results. Table III lists the AP for all the 20 categories in this dataset. For each class, the method obtaining the highest AP is marked in bold.

TABLE III
Detection results on Pascal VOC 2007 dataset. AP measure (%) is shown for each category.

<table>
<thead>
<tr>
<th>Category</th>
<th>plane</th>
<th>bicycle</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASCAL 07 Max</td>
<td><strong>26.2</strong></td>
<td><strong>40.9</strong></td>
<td>9.8</td>
<td>9.4</td>
<td>21.4</td>
<td><strong>39.3</strong></td>
<td><strong>43.2</strong></td>
<td><strong>24.0</strong></td>
<td>12.8</td>
<td><strong>14.0</strong></td>
</tr>
<tr>
<td>PASCAL 07 Median</td>
<td>15.6</td>
<td>26.2</td>
<td>4.3</td>
<td>2.5</td>
<td>3.4</td>
<td>19.7</td>
<td>29.4</td>
<td>9.9</td>
<td>9.6</td>
<td>10.0</td>
</tr>
<tr>
<td>Proposed method</td>
<td>24.1</td>
<td>36.9</td>
<td><strong>10.0</strong></td>
<td><strong>10.4</strong></td>
<td><strong>23.8</strong></td>
<td>32.7</td>
<td>35.0</td>
<td>19.5</td>
<td><strong>13.4</strong></td>
<td>13.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASCAL 07 Max</td>
<td><strong>9.8</strong></td>
<td><strong>16.2</strong></td>
<td><strong>33.5</strong></td>
<td><strong>37.5</strong></td>
<td><strong>22.1</strong></td>
<td><strong>12.0</strong></td>
<td><strong>17.5</strong></td>
<td>14.7</td>
<td><strong>33.4</strong></td>
<td>28.9</td>
</tr>
<tr>
<td>PASCAL 07 Median</td>
<td>6.3</td>
<td>9.5</td>
<td>19.8</td>
<td>22.4</td>
<td>12.1</td>
<td>4.5</td>
<td>4.7</td>
<td>9.6</td>
<td>16.3</td>
<td>23.9</td>
</tr>
<tr>
<td>Proposed method</td>
<td>8.2</td>
<td>14.7</td>
<td>29.2</td>
<td>31.5</td>
<td><strong>22.4</strong></td>
<td>11.1</td>
<td><strong>17.5</strong></td>
<td><strong>15.2</strong></td>
<td>28.8</td>
<td><strong>30.4</strong></td>
</tr>
</tbody>
</table>

As shown in Table III, the proposed method outperforms the median results of all the 20 categories without using a bounding box for training. Furthermore, our method achieves the highest AP score in 8 categories such as bird, bottle, sheep, monitor etc. These classes share the common property that they all have clearly predictable visual properties, such as distinctive parts and relatively fixed spatial arrangement of parts. Thus our part-based method can effectively characterize these properties and achieve good detection performance. For those categories with significant variation in structure (e.g. table and plant), the proposed method is less effective.
It should be noted that the proposed method is based on the structural shape feature. Therefore, it is somehow influenced by the appearance and viewpoint of an object. For example, our method does not achieve the best accuracy on the car and bicycle categories due to their varying poses.
Method based on a BOW/spatial pyramid representation [42] delivers better performance because it can better capture the texture feature in these categories. When it comes to categories in which shape plays an important role in detection, such as boat and bottle, our method shows clear advantages over the methods that can only capture coarse shape information. In general, the proposed method performs well on all categories in the PASCAL dataset comparing with the average performance on the dataset, and it works especially well on those objects with clear structure information.

Finally, we analyze the computational complexity of the proposed method and report its time cost in both training and testing stage. The most time consuming steps are the pairwise image matching by fast EMD method and the feature weight update steps. Given a training set with $T$ images, assume by average each image has $N$ features. The complexity of the fast EMD method over the whole training set is $O(T^2 N^2 U \log N)$, where $U$ is a bound on the integral operation [37]. Furthermore, the complexity of feature weight update iteration is $O(T N^3)$. In the testing stage, however, the computational complexity is much lower. This is because each testing image shall be compared with the object models in the training set for only once. The most time consuming step is the building of object models, whose time complexity is $O(wKN)$, where $w$ is the number of image patches generated by sliding windows and $K$ is the total number of clusters in the root and part object models. In both training and testing stage, extraction of PAS context feature and feature pooling take by average 0.2 and 0.3 seconds per image, respectively. Feature selection and building of part-based model take by average 19.8 seconds and 2.0 seconds per image, respectively. These two steps take fairly long time due to the high complexity of the steps. Note that the proposed method shares the important property of contour context models that do not depend on large number of training samples [9]. Therefore, the training time is not a major concern in this case. In the object detection step, object detection in each testing image takes by average 1.1 seconds.

V. Conclusion

In this paper, we have introduced a part-based shape model for object detection. We have shown how our proposed contour feature descriptor can be used for foreground feature selection by iteratively computing pairwise fast EMDs. Moreover, we have built a simple but effective shape model using the selected features for the detection process, which does not require ground
truth bounding boxes during the training stage. The results indicate that the performance of the proposed method is comparable to other methods reported in the literature. Future work will deal with a more precise framework for the shape part description.

REFERENCES


