Object Detection Using A Shape Codebook

Xiaodong Yu[1], Li Yi[1], Cornelia Fermuller[2], David Doermann[2]

[1] Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20740 USA
{xdyu,liyi}@umd.edu

[2] Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20740 USA
fer@cfar.umd.edu,doermann@umiacs.umd.edu

Abstract

This paper presents a method for detecting categories of objects in real-world images. Given training images of an object category, our goal is to recognize and localize instances of those objects in a candidate image.

The main contribution of this work is a novel structure of the shape codebook for object detection. A shape codebook entry consists of two components: a shape codeword and a group of associated vectors that specify the object centroids. Like their counterpart in language, the shape codewords are simple and generic such that they can be easily extracted from most object categories. The associated vectors store the geometrical relationships between the shape codewords, which specify the characteristics of a particular object category. Thus they can be considered as the "grammar" of the shape codebook.

In this paper, we use Triple-Adjacent-Segments (*TAS*) extracted from image edges as the shape codewords. Object detection is performed in a probabilistic voting framework. Experimental results on public datasets show performance similiar to the state-of-the-art, yet our method has significantly lower complexity and requires considerably less supervision in the training (We only need bounding boxes for a few training samples, do not need figure/ground segmentation and do not need a validation dataset).

1 Introduction

Recently, detecting object classes in real-world images using shape features has been explored in several papers. Compared to local features such as SIFT [10], shape features are attractive for two reasons: first, many object categories are better described by their shape than texture, such as cows, horses or cups; second, for objects with wiry components, such as bikes, chairs or ladders, local features unavoidably contain large amount of background clutter [1, 13]. Thus shape features are often used as a replacement of, or complement to local features [2, 6, 17].

One practical challenge for shape features is that they are less discriminative than local features . To overcome this limitation, several methods have been proposed to use a shape codebook for object detection [4, 16, 21]. Inspired by these works, we propose a new structure of the shape codebook for object detection in this paper. In the shape codebook, the shape codewords should be simple and generic such that they can be reused in different object categories. The geometrical relationships between the shape codewords specify the characteristics of a particular object category. Thus they can be viewed as the "grammar" of the shape codebook.

In this paper, we explore a local shape feature proposed by Ferrari *et al*, [4] as the shape codeword and use the *Implicit Shape Model* [7, 8] to define the shape grammar. The shape feature is formed by chains of k connected, roughly straight contour segments (kAS). In particular, we use $k = 3$, which is called Triple-Adjacent-Segments (*TAS*). A *TAS* codebook entry consists of two components. (1) A prototype *TAS* that represents a group of similiar *TAS*s, which is called *TAS* codeword. (2) a group of vectors specifying the associated object centroids, and encode the shape grammar. During detection, we match each *TAS* from the test image to the codebook. When an entry in the codebook is activated, it casts votes for all possible object centroids based on the associated vectors. Finally, candidate object centroids are detected as maxima in the continuous voting space using Mean-Shift Mode Estimation. The object boundary is then refined as the enclosure of the matched *TAS*s associated to the detected object centroid.

The main contributions of this work are:

1. We propose a two-layer structure of the shape codebook for object detection. Simple and generic shape features are used as shape codewords and geometrical constraints are used as the shape grammar. Since the shape codewords are not designed for specific object classes (e.g., cows, horses, cars), they only need to be learned once. Then they can be used in all object categories.

2. We seperate the procedures of learning shape codewords and building shape grammar. With a set of learned shape codewords, shape grammar can be learned for a new object category using a simple nearest neighbor rule. This method significantly reduces the complexity of the codebook and makes our algorithm more flexible.

The paper is structured as follows. The next section reviews related work. The proposed algorithm is described and evaluated in Section 3 and Section 4 respectively. Finally, Section 5 presents conclusions and future work.

2 Related Work

Codebook of local features for object categorization and detection: The idea of learning a codebook for object categorization and detection has widely been used in approaches using local features in recent years [2, 3, 5, 11, 15, 19, 22, 24]. One of the key differences between these algorithms lies in the way the geometric configuration of parts in an object being exploited. The simple "bag-of-words" model is used in [5, 15, 24], where geometrical constraints among visual words are discarded. Loose spatial constraints are used in [22] to detect the co-occurence of pairs of visual words within a local spatial neighborhood. A slightly tighter spatial constraint called "spatial weighting" is decribed in [11], where the features that agree on the position and shape of the object are boosted and the background features are suppressed. Russell *et al* [19] encode the spatial relationship among visual words from the same object using segmentation information. Fergus *et al* [2] adopt a parameterized geometric model consisting of a joint Gaussian over the centroid position of all the parts. Translation and scale information is explicitly built in a pLSA model in [3], and clear improvement using this model is demonstrated on object classes with great pose variability.

Codebook of shape features for object categorization and detection: The idea of learning a codebook has also been explored for shape features [4, 6, 9, 14, 16, 18, 21]. The different approaches employ diverse methods for building the shape codebook and using the geometrical constraints. Mori *et al* [14] quantize shape context vectors into a small number of canonical shape pieces, called *shapemes*. Liu *et al* [9] apply the "bag-of-words" model to 3D shape retrieval. Neither algorithm stores the spatial information. Kumar *et al* cluster outlines of object parts into a set of exemplar curves to handle variability in shape among members of an object class [6]. A pictorial structure is employed to represent the spatial relationship between parts of an object. Opelt *et al* [16, 18] build a codebook for class-discriminative boundary fragments and use boosting to select discriminative combinations of boundary fragments to form strong detectors. Similarly, a fragment dictionary is built by Shotton *et al* [21]. The differences between them are: the former requires no segmentation mask while the latter does; the former uses the spatial relationship between the boundary segments in a model similar to Leibe's approach [7], while the latter uses grids. Ferrari *et al* [4] build a codebook of *k*AS using the clique-partioning approximation algorithm. Compared to the codebooks used in [6, 16, 18, 21], the *k*AS codebook is generic and not designed for specific object classes (e.g., cows, horses, cars). Thus, once a codebook for a particular *k* has been learned, it can be used in all object classes.

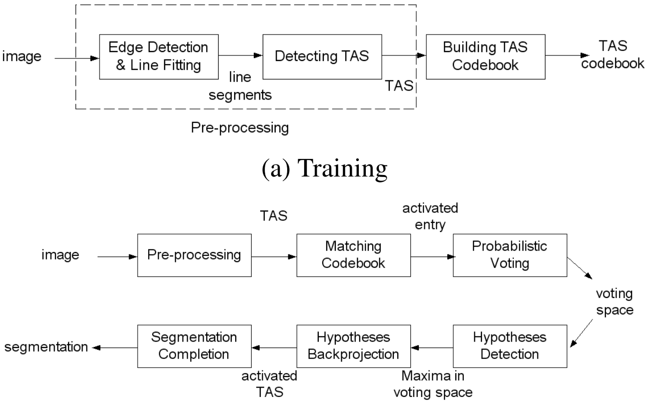
3 The Algorithm

In this section, we present the details of the proposed algorithm (Figure 1). First, the preprocessing steps are described in Section 3.1. Then we discuss the approach for building the *TAS* codebook in Section 3.2. Finally, Section 3.3 explains how to detect objects in a test image.

3.1 Detecting and Comparing *TAS*s

The *TAS* is used as the shape feature in our work. It is a special case of the *k*AS, which is formed by a chain of *k* connected, roughly straight contour segments. It has been shown that *k*AS is very robust to edge clutter. Since only a small number ($k \leq 5$) of connected segments are used, kAS can tolerate the errors in edge detection to some extent. Thus *k*AS is an attractive feature compromising between information content and repeatability. In the work of [4], object class detection is implemented using a sliding window mechanism and the best performance is achieved when $k = 2$.

We choose $k = 3$ in this work. As *k* grows, *k*AS can present more complex local shape structures and becomes more discriminative but less repeatable. Ferrari *et al* [4] point out that *k*AS of higher complexity are attractive when the localization constraints are weaker, and hence the discriminative power of individual features becomes more important. In



(a) Training

(a) Object Category Detection

Figure 1: An overview flowchart of the proposed algorithm.

this work, since we do not apply explicit spatial contraints, such as dividing the sliding window into a set of tiles, it is appropriate to use a *k*AS of higher degree.

The procedure to detect *TAS*s is summarized as follows: first, we detect image edges using the Berkeley Segmentation Engine (BSE) [12]. The BSE supresses spurious edges and has a better performance than the Canny edge detector. Second, small gaps on the contours are completed as follows: every edgel chain c_1 is linked to another edgel chain c_2, if c_1 would meet c_2 after extending n pixels. Contour segments are fit to straight lines. Finally, starting from each segment, every triplet of line segments is detected as a *TAS*.

Let θ_i, $l_i = \|s_i\|$ be the orientation and the length of s_i, where s_i for $i = 1,2,3$ denote the three segments in a *TAS P*. Two *TAS*s P^a and P^b are compared using the following measure $D(a,b)$

$$D(a,b) = w_\theta \sum_{i=1}^{3} D_\theta(\theta_i^a, \theta_i^b) + \sum_{i=1}^{3} |log(l_i^a/l_i^b)|, \tag{1}$$

where $D_\theta \in [0,1]$ is the difference between segment orientation normalized by π. Thus the first term measures the difference in orientation and the second term measures the difference in length. A weight $w_\theta = 2$ is used to emphasize the difference in orientation because the length of the segment is often inaccurate.

3.2 Building the *TAS* codebook

Building the *TAS* codebook consists of two stages: learning *TAS* codewords and learning *TAS* grammar. They are discussed in Section 3.2.1 and Section 3.2.2 respectively.

3.2.1 Learning *TAS* codewords

The *TAS* codewords are learned from *TAS*s in a training image set. First, we compute the distance of each pair of training *TAS*s. Then, we obtain a weighted graph $G = (V,E)$, where the nodes of the graph are the training *TAS*s, and an edge is formed between every pair of *TAS*s. The weight on each edge, $w(a,b) = \exp(-D(a,b)^2/\sigma_D^2)$ is a function of the distance between two *TAS*s P^a and P^b, where σ_D is set to 20 percent of the maximum of all $D(a,b)$. Then clustering the training *TAS*s is formulated as a graph partition problem, which can efficiently be solved using the Normalized Cut algorithm [20].

After obtaining the clustering results from the Normalized Cut algorithm, we select a *TAS* codeword, J_i, from each cluster i. The *TAS* codeword is selected as the *TAS* closest to the cluster center (i.e., it has the smallest sum of distances to all the other *TAS*s in this cluster). Each codeword J_i is associated with a cluster radius r_i, which is the maximum distance from the cluster center to all the other *TASs* within this cluster.

Figure 2.a shows the 40 most frequent *TAS* codewords in the codebooks learned from 10 images in the Caltech motorbike dataset. We can observe that the most frequent *TAS* codewords have generic configurations of three line segments. Quantitatively, we compared the codewords from variant datasets and found 90% to 95% of the *TAS* codewords are similiar. This confirms that the *TAS* codebooks are generic. In the following experiments, we apply the codewords learned from the Caltech motorbike dataset to all datasets.

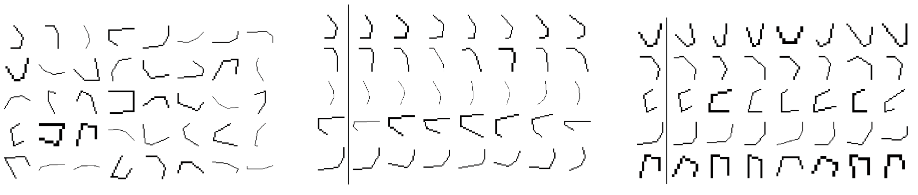


Figure 2: Examples of *TAS* codewords. (a) shows the 40 most frequent *TAS* codewords learned from 10 images in the Caltech motorbike dataset. (b) and (c) illustrate the 5 most frequent *TAS* codewords (the first column) and their associated members in the clusters for the Caltech motorbikes dataset and the cows dataset respectively.

3.2.2 Learning *TAS* Grammar

To learn the *TAS* grammar, we need training images with the object delineated by a bounding boxes. First, we apply the nearest neighbor rule to quantize the *TAS*s within the bounding boxes using the *TAS* codewords. Let's denote e_k a *TAS* and J_i the nearest neighbor in the codebook. The *TAS* e_k is quantized as J_i if $D(J_i,e_k) < r_i$. Figure 2.b and 2.c show the 5 most frequent *TAS* codewords in two datasets and their associated members in the cluster. We found that only less than 2% of the *TASs* in all datasets can not be found in the *TAS* codewords learned from the motorbike dataset. This further confirms the generality of the *TAS* codebook.

The *TAS* grammar is defined using the *Implicit Shape Model* [7]. For the member *TAS*s in cluster i of size M_i, we store their positions relative to the object center $(v_m, m = 1, ..., M_i)$. Thus, a codebook entry records the following information: $\{J_i; (v_m, m = 1, ..., M_i)\}$. For simplicity, we might also use J_i to denote the codebook entry.

3.3 Detecting Object Category by Probabilistic Voting

The procedure for detecting object category is illustrated in Figure 1.b. First, we match each test image *TAS* e_k located at l_k to the codebook. A codebook entry J_i is declared to be matched (activated) if $D(J_i, e_k) < r_i$. For each matched codebook entry J_i, we cast votes for possible locations of the object centers $(y_m, m = 1, ..., M_i)$, where y_m can be obtained from l_k and v_m. Then, object category detection is accomplished by searching for local maxima in the probabilistic voting space after applying Parzen window probability density estimation. Formally, let x_n be a candidate position in the test image and $p(x_n)$ be the probability that object appears at position x_n. Candidate object centers x^* defined as follows,

$$x^* = arg \max_x \sum_{x_n \in W(x)} p(x_n), \tag{2}$$

where $W(x)$ is a circular window centered at x. The probability $p(x_n)$ is obtained by observing evidence e_k in the test image. Thus, conditioned on e_k, we marginalize $p(x_n)$ as follows

$$p(x_n) = \sum_k p(x_n|e_k)p(e_k). \tag{3}$$

Without any prior knowledge on $p(e_k)$, we assume it is uniformly distributed, i.e., $p(e_k) = 1/K$, where K is the number of TASs in the test image.

Let $\mathbf{S}$ be the set of matched codewords, $p(x_n|e_k)$ can be marginalized on $J_i \in \mathbf{S}$

$$\begin{aligned} p(x_n|e_k) &= \sum_{J_i \in \mathbf{S}} p(x_n|J_i, e_k)p(J_i|e_k) & (4) \\ &= \sum_{J_i \in \mathbf{S}} p(x_n|J_i)p(J_i|e_k). & (5) \end{aligned}$$

After matching e_k to J_i, the voting will be performed by members within J_i. Thus $p(x_n|J_i, e_k)$ is independent of e_k and Equation 4 can be reduced to Equation 5. In Equation 5, the first term is the probabilistic vote for an object position given an activated codebook entry J_i, and the second term measures the matching quality between J_i and e_k. We use $p(J_i|e_k) \propto \exp(-D(e_k, J_i)^2/r_i^2)$ to evaluate the matching quality.

For an activated codebook entry, we cast votes for all possible locations of the object centers y_m. Thus $p(x_n|J_i)$ can be marginalized as

$$\begin{aligned} p(x_n|J_i) &= \sum_m p(x_n|y_m, J_i)p(y_m|J_i) & (6) \\ &= \sum_m p(x_n|y_m)p(y_m|J_i). & (7) \end{aligned}$$

Since the voting is casted from each individual member in J_i, the first term in Equation 6 can be treated as independent of J_i. Then Equation 6 is reduced to Equation 7. Without prior knowledge of y_m, we treat them equally and assume $p(y_m|J_i)$ is a uniform distribution, i.e., $p(y_m|J_i) = 1/M_i$.

The term $p(x_n|y_m)$ measures the vote obtained at location x_n given an object center y_m. Since we only vote at the location of possible object centers, we have $p(x_n|y_m) = \delta(x_n - y_m)$, where $\delta(t)$ is the Dirac delta function.

Combining the above equations, we can compute $p(x_n)$ from the evidence e_k located at l_k. In order to detect instances of the object category, we search for the local maxima x^* in the voting space after applying Parzen window probability density estimation. The score of these candidates is defined as $\sum_{x_n \in W(x^*)} p(x_n)$. If this score is greater than a threshold t_{score}, we classify this image belonging to the training object category. To obtain a segmentation of the object instance, we find the test *TAS*s voting within $W(x^*)$ for an x^*. Then we obtain a smooth contour from these *TAS*s using the Gradient Vector Flow snake algorithm [23]. Also a bounding box is obtained in this procedure for each object instance. Figure 3 shows some detection examples for the Caltech motorbikes dataset and the cows dataset.

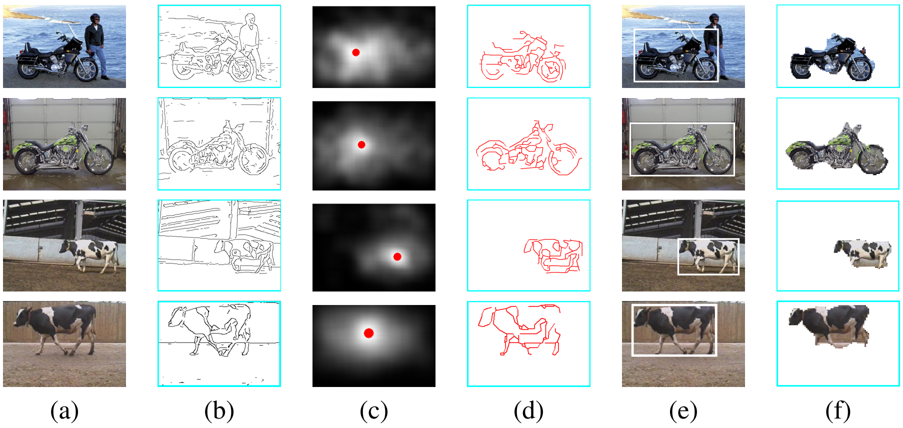


Figure 3: Example detection results for the Caltech motorbikes dataset and the cows dataset. (a) The originial images. (b) The edge maps. (c) The voting spaces and detected centroids. (d) The backprojected *TAS*s. (e) The bounding box of the detected objects. (f) The segmentation

4 Experimental Results

In this section, we evaluate the performance of the proposed algorithm and compare it to the state-of-the-art algorithms that detect object categories using shape features. If a test image has a detection score greater than the threshold t_{score} and the overlap between the detected bounding boxes and the ground truth is greater than 50%, we consider the detection (localization) correct. By varying t_{score} we can obtain different recall/precision values. The performance is evaluated in terms of the Recall-Precision Equal Error Rate (RPC EER). All parameters are kept constant for different experiments.

The training data includes training images with bounding boxes annotating instances of the object class. Compared to the state-of-the-art, we require the least supervision. [16, 18] uses training image with bounding boxes and validation image sets that include both positive and negative images. [4] also requires negative images to train the SVM classifier. [21] requires segmentation masks for 10 positive training images plus a large amount of positive and negative images to train a discriminative classifier.

Cows Dataset: We use the same cow dataset as in [16] and compare to their results: 20 training images and 80 test images, with half belonging to the category cows and half to negative images (Caltech airplanes/faces). But we do not use the validation dataset while [16] uses a validation set with 25 positive/25 negative.

The performance is shown in Table 1. We also shows the variation in performance with the number of training images. The results show that our approach outperforms or

Table 1: Performance (RPC EER) depending on the number of training images with bounding boxes (N_{BB}) on the cows dataset and comparison to other published results.

	N_{BB}=5	N_{BB}=10	N_{BB}=20
Ours	0.93	0.95	0.96
Opelt [16]	0.91	0.95	1.00

Table 2: Performance (RPC EER) on the cups dataset and comparison to other publised results. N_{BB} is the number of training images with bounding boxes; N_V is the numbers of validation images.

	N_{BB}	N_V	RPC EER
Ours	16	-	0.841
Opelt [16]	16	30	0.812

performs as well as Opelt's when the number of training images is small ($N_{BB} = 5, 10$) but is outperformed when the number of training image is large ($N_{BB} = 20$). It shows that our approach is favorable when there are small number of training images available. The reason is that the *TAS* feature is very simple and generic. Thus only a few training images is sufficient to discover the statistical patterns in the training images. In comparison, Opelt's features are more complex and have more discriminative power for a particular object. Hence more training images are needed to fully exploit their advantages.

Cup Dataset: In this test, we evaluate our approach on the cup dataset used in [18]. We use 16 training images and test on 16 cup images and 16 background images. We do not use the validation set with 15 positive/15 negative, which is used in [18].

The performance is summarized in Table 2. It shows that we can achieve slightly better performance than Opelt's algorithm even we use less supervision in the training.

Caltech Motorbikes Dataset: In this test, we evaluate our algorithm using the Caltech motorbikes dataset [2]. Training is conducted on the first 10 images in this dataset. Testing is conducted on 400 novel motorbike images and 400 negative images from Caltech airplane/face/car rear/background images.

The experimental results are compared to other publised results on object localization in Table 3. We also compared the degree of supervision in the training in terms of the number of variant types of training images. It is shown that we can achieve performance compariable to Shotton's method but are slightly worse than Opelt's. This should be attributed to the class-discriminative contour segments used by Opelt *et al.*

Table 3: Comparison of the proposed algorithm to other publised results on the Caltech motorbikes dataset. Column 2 through 5 are the numbers of variant types of training images: N_S for images with segmentations; N_U for images without segmentations; N_{BB} for images with bounding boxes; N_V for validation images.

	N_S	N_U	N_{BB}	N_V	RPC EER
Ours	-	-	10	-	0.921
Shotton [21]	10	40	-	50	0.924
Opelt [16]	-	-	50	100	0.956

Discussion: The advantage of the proposed method lies in its low complexity. The *TAS* codewords only need to be learned once. Thus the learning procedure for a new object category can be reduced to a simple nearest neighbor search for the training *TAS*s and the time-consuming clustering can be skipped. Furthermore, There are a limited number of possible configurations of three line segments. In our experiments, the *TAS* codebook has 190 entries. Ferrari *et al* [4] reported a *TAS* codebook with 255 entries because they used more complex descriptors. Nevertheless, the number of the shape codewords is bounded, rather than increasing linearly with the number of class categories as in the codebook used in [18, 16].

5 Conclusion

We have presented a two-layer structure of the shape codebook for detecting instances of object categories. We proposed to use simple and generic shape codewords in the codebook, and to learn shape grammar for individual object category in a seperate procedure. This method is more flexible than the approaches using class-specified shape codewords. It achieves similiar performance with considerable lower complexity and less supervision in the training. And thus it is favorable when there is a small number of training images available or the training time is crucial.

Currently we are investigating methods to combine several shape codewords in the voting. We will also try other clustering methods, e.g., k-means, aggolomerative clustering, etc., and compare the *TAS* codebooks to those used in this paper. Finally we plan further evaluation of the proposed method in more challenging datasets and over more categories.

References

[1] Owen Carmichael and Martial Hebert. Shape-based recognition of wiry objects. In *IEEE Conference On Computer Vision And Pattern Recognition*. IEEE Press, June 2003.

[2] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 02, page 264, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[3] Robert Fergus, Fei-Fei Li, Pietro Perona, and Andrew Zisserman. Learning object categories from google's image search. In *ICCV*, pages 1816–1823, 2005.

[4] Vittorio Ferrari, Loic Fevrier, Frederic Jurie, and Cordelia Schmid. Groups of adjacent contour segments for object detection. *Technical Report*, 2006.

[5] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005.

[6] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *Proceedings of the British Machine Vision Conference*, 2004.

[7] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, Czech Republic, May 2004.

[8] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference (BMVC'03)*, pages 759–768, Norwich, UK, Sept. 2003.

[9] Yi Liu, Hongbin Zha, and Hong Qin. Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2025–2032, Washington, DC, USA, 2006. IEEE Computer Society.

[10] D. Lowe. Distinctive image features from scale-invariant keypoints. 20:91–110, 2003.

[11] Marcin Marszalek and Cordelia Schmid. Spatial weighting for bag-of-features. In *CVPR*, pages 2118–2125, Washington, DC, USA, 2006. IEEE Computer Society.

[12] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004.

[13] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 779–788, 2003.

[14] Greg Mori, Serge Belongie, and Jitendra Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[15] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[16] Andreas Opelt, Axel Pinz, and Andrew Zisserman. A boundary-fragment-model for object detection. In *ECCV*, pages 575–588, 2006.

[17] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Fusing shape and appearance information for object category detection. In *BMVC*, pages 117–127, Edinburgh, UK, 2006.

[18] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR*, pages 3–10, Washington, DC, USA, 2006. IEEE Computer Society.

[19] Bryan C. Russell, William T. Freeman, Alexei A. Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. volume 02, pages 1605–1614, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[20] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[21] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Contour-based learning for object detection. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 503–510, Washington, DC, USA, 2005. IEEE Computer Society.

[22] Josef Sivic, Bryan Russell, Alexei A. Efros, Andrew Zisserman, and Bill Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV 2005)*, October 2005.

[23] C. Xu and J. Prince. Gradient vector flow: A new external force for snakes. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '97)*, pages 66–71, San Juan, Puerto Rico, June 1997.

[24] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. Computer Vision*, 73(2):213–238, 2007.