Fan Shape Model for Object Detection

Xinggang Wang*[1] Xiang Bai[1] Tianyang Ma[2] Wenyu Liu[1] Longin Jan Latecki[2]
[1] Department of Electronics and Information Engineering, Huazhong University of Science and Technology
[2] Department of Computer and Information Sciences, Temple University
wxghust@gmail.com,xiang.bai@gmail.com,tianyang.ma@temple.edu,liuwy@hust.edu.cn,latecki@temple.edu

Abstract

We propose a novel shape model for object detection called Fan Shape Model (FSM). We model contour sample points as rays of final length emanating for a reference point. As in folding fan, its slats, which we call rays, are very flexible. This flexibility allows FSM to tolerate large shape variance. However, the order and the adjacency relation of the slats stay invariant during fan deformation, since the slats are connected with a thin fabric. In analogy, we enforce the order and adjacency relation of the rays to stay invariant during the deformation. Therefore, FSM preserves discriminative power while allowing for a substantial shape deformation. FSM allows also for precise scale estimation during object detection. Thus, there is not need to scale the shape model or image in order to perform object detection. Another advantage of FSM is the fact that it can be applied directly to edge images, since it does not require any linking of edge pixels to edge fragments (contours).

1. Introduction

In this paper, we present a flexible shape model, named Fan Shape Model (FSM). As shown in Fig. 1, a folding fan is composed of slats (connected with a thin material, e.g., fabric or paper) that revolve around a pivot. Similarly, FSM is composed of rays that may revolve around a center point. Each ray corresponds to a part of the object, and the spatial distribution of the parts is modeled by restricting the range of flexibility of their rays. As is the case for a folding fan, the neighborhood structure of rays, and consequently their order, is preserved during the deformation. This is one of the key features that provides the discriminative power of FSM. On the other hand, the ray flexibility offers FSM high tolerance to shape variation within class.

FSM is learned from segmented objects in training images. Learning starts from shape matching. Across different shapes belonging to the same object category, the correspondence of parts is first established by matching. Shape deformation is modeled with distributions of ray parameters. We stress that learning FSM only requires positive training images, which makes the learned models universally applicable. We demonstrate this fact in Fig. 2, where we use a bottle FSM trained on ETHZ dataset [11] to detect bottles on PASCAL dataset [7].

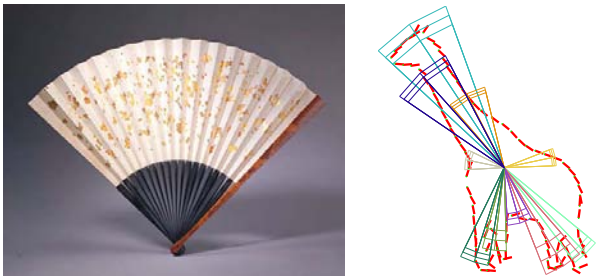

Figure 1. Left image shows a folding fan. Right image shows our fan shape model for giraffe. Each part of giraffe is modeled as a flexible ray. Ten rays are plotted in different colors together with their flexibility regions.

Scale variance is big challenge for object detection. To deal with this problem, the current methods either scale their shape models or the testing images. Either way, the computational cost is multiplied by the number of scales. In contrast when using FSM for object detection, we use local edge pixel voting for scale estimation. Then the final evaluation is only performed at the estimated best scale at a given image location.

Recently, most shape based object detection methods use bottom-up image contours, such as [10], [22], [25] and [15]. Given a gray scale image, edge pixels are obtained by edge detector, such as Canny [3] or Pb [18]. Based on the edge pixels, image contours are obtained using an edge-linking algorithm, and the obtained contours are utilized as basic elements in these methods. Shape descriptors are used to compare the shape of selected contours to known model contours. Ferrari et al. [10] introduce Pair of Adjacent Segments (PAS) feature. Srinivasan et al. [22] and Zhu et

*Part of this work was done while the author was at Temple University.

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

al. [25] both use shape context [2] to encode the shape of an image contour. In [15], Lu et al. consider relations between triples of points and compute a histogram for each contour according to the angles between all triple points.

Grouping edge pixels into contours can increase the discriminative power compared to only considering an unstructured set of edge pixels. However, edge-linking algorithm, such as [13], can only provide good image contours if edge quality allows the connectivity between edge pixels is able to be identified. For images with bad edge quality, edge-linking algorithm may fail to give any meaningful images contours. This significantly limit the applications of image contour-based object detection methods. This fact is illustrated in Fig. 2. Therefore, the proposed object detection framework with FSM does not require any edge linking.

Direct template matching like chamfer matching [21] also has this property. However, the matched templates are not flexible. Therefore, a large number of edge templates is needed for object detection. In contrast, we utilize only one flexible template.

To summarize, there are five main advantages of the fan shape model: (1) FSM is very flexible, and has the ability to tolerate substantial shape variance; (2) learning FSM requires only positive training images; (3) FSM can fast infer object scale, which could also be used by other object detection systems; (4) FSM is robust to broken edges, it can obtain very impressive detection results even when the edge quality is bad; (5) FSM can easily combine both shape and texture descriptors for object detection.

2. Related Work

Recent year, a large range of shape-based object detection and recognition methods has been proposed. Many of them achieve state-of-the-art performance by only utilizing edge information in images. For example, Shotton et al. [21] and Opelt et al. [19] learn codebook of contour fragments first, then use Chamfer distance to match learnt fragments to edge images. In [10, 11], Ferrari et al. build a network of nearly straight adjacent segments (kAS), and use them to match between model parts and images. Zhu et al. [25] formulate the shape matching of contour in clutter as a set to set matching problem, and present an approximate solution to the hard combination problem. To address the non-rigid object deformation, Bai et al. [1] use the skeleton information to capture the main structure of an object, and use Oriented Chamfer Matching [21] to match the model parts to images. Lu et al. [15] first decompose the shape model into several part bundles, and use particle filter to simultaneously perform selection of relevant contour fragments in edge images, grouping of the selected contour fragments, and matching to the model contours. Most recently, Srinivasan et al. [22] address the contour grouping problem as many-to-one matching, and use this scheme in both training and testing phases. For purpose of improving detection and score ranking, a training process is designed in which latent SVM is used to guarantee the many-to-one score is tuned discriminatively. Edge information is also utilized in [24, 16] to obtain the state-of-the-art performance on the ETHZ shape dataset [11].

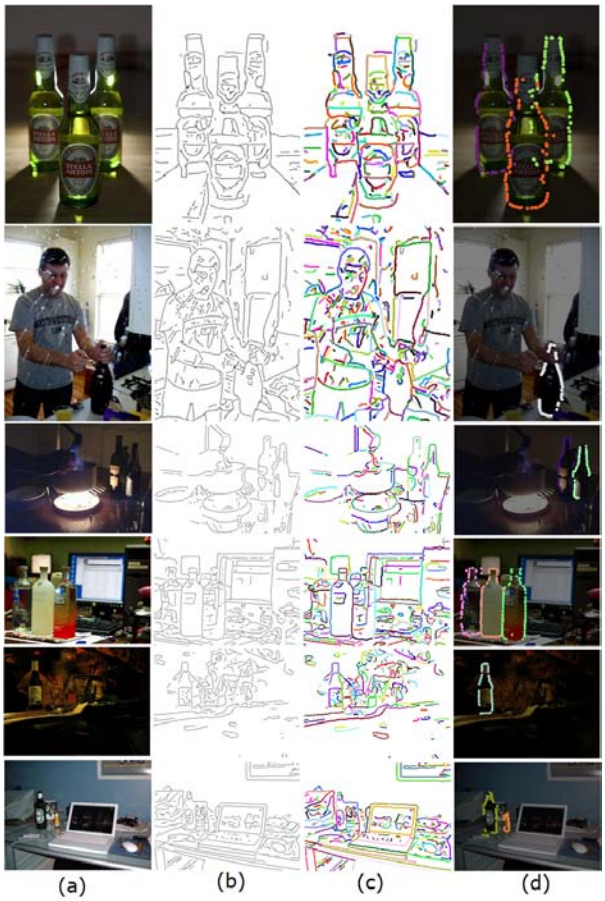

Figure 2. Bottles detection results when edges are severely broken. (a) shows bottle images from PASCAL dataset [7]; (b) shows the edge maps computed using Pb edge detector [18]; (c) shows edge-linking results by [13]; (d) shows our bottle detection results using a bottle FSM trained on ETHZ dataset [11] (shown in Fig. 5).

The proposed fan shape model, as a typical part-based object model, is similar to [9]. Both [9] and our model can be automatically learned from weakly supervised image data. Previous fundamental research of the part-based object model can be found in [6], [5] *etc*.

3. Fan Shape Model

In this section, we give specific details about the proposed fan shape model.

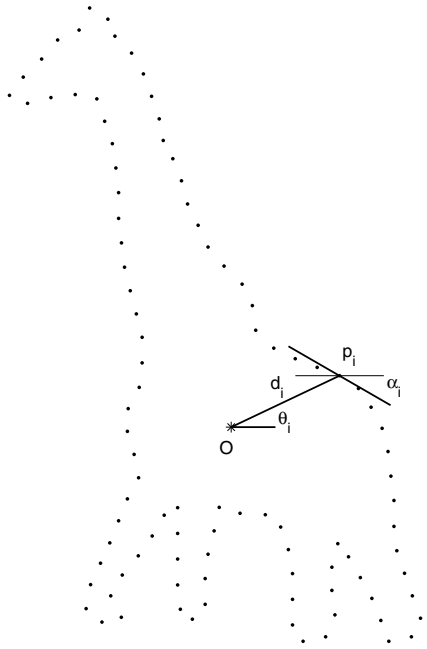

Figure 3. Ray based shape representation

3.1. Ray Based Shape Representation

For a shape, *i.e.* a segmented object S, we can extract its contour, and densely sample the contour into n ordered points $\{p_1\ p_2\ \dots\ p_n\}$. Given a reference point o, we use ray $\overrightarrow{op_i}$ to describe point $p_i, 1 \leq i \leq n$. Parameters of ray $\overrightarrow{op_i}$ is $R_i(S, o) = (\theta_i, d_i, \alpha_i)$. As shown in Fig. 3, θ_i is inclined angle of ray, it ranges in $[0, 2\pi]$, d_i is Euclidean distance between point o and p_i, α_i is edge orientation at the contour point p_i, it ranges in $[0, \pi]$. We represent shape S as

$$\{R_i(S, o), i = 1 \dots n\}$$

There is only one reference point o of a shape S. Therefore, to simplify the notation, we use $R_i(S)$ to replace $R_i(S, o)$ below when possible.

3.2. Shape Matching with Dynamic Programming

Let $\{R_i(S_1), i = 1 \dots n\}$ and $\{R_j(S_2), j = 1 \dots m\}$ represent two shapes S_1 and S_2 with their corresponding reference points o_1 and o_2. The shape matching problem is then formulated as finding a mapping ϕ from $\{i = 1, 2, \dots, n\}$ to $\{j = 0, 1, 2, \dots, m\}$, where $R_i(S_1)$ is mapped to $R_{\phi(i)}(S_2)$, if $\phi(i) \neq 0$, otherwise $R_i(S_1)$ is unmatched. ϕ should minimize the matching cost $C(\phi)$ defined as

$$C(\phi) = \sum_{1 \leq i \leq n} c(R_i(S_1), R_{\phi(i)}(S_2)), \qquad (1)$$

where $c(R_i(S_1), R_{\phi(i)}(S_2)) = \tau$, if $\phi(i) = 0$. τ is the penalty for leaving ray $R_i(S_1)$ unmatched. Otherwise, $c(\cdot)$ is the cost for matching two rays $R_i(S) = (\theta_i, d_i, \alpha_i)$ and $R_{i'}(S') = (\theta_{i'}, d_{i'}, \alpha_{i'})$ defined as

$$c(R_i(S), R_{i'}(S')) = \lambda_t * \min(|\theta_i - \theta_{i'}|, 2\pi - |\theta_i - \theta_{i'}|) + \lambda_d * |d_i - d_{i'}| + \lambda_a * \min(|\alpha_i - \alpha_{i'}|, \pi - |\alpha_i - \alpha_{i'}|), \qquad (2)$$

where λ_t, λ_d, and λ_a are weights.

Since the contours provide the order of the rays on two shapes, it is natural to restrict the matching ϕ obeying this order. Therefore, dynamic programming can be used to solve the matching problem. It has been widely used for contour matching task, and is proved to be able to obtain a stable matching result between two different shapes. We use the standard dynamic programming method [4] with cost function defined in Eq. (1) and Eq. (2).

3.3. Learning a Fan Shape Model

In order to learn the parameters of FSM for a given shape class $\mathcal{S}$, we only need a set of positive training images $I_1, I_2, \dots, I_M$. For each image, we also need a segmented contour of the target shape $S_i \subset I_i$ for $i = 1, \dots, M$.

There are some existing approaches [1, 23] to learn structural shape model for object detection. Two biggest differences between our shape model and the existing models are: (1) shape models in [1, 23] are built on skeleton based shape representation, while fan shape model is built on contour point representation with rays. (2) shape models in [1, 23] rely only on contour matching in detection phase, while fan shape model can utilize several kinds of appearance descriptors rather than only shape contour.

Definition of fan shape model: The fan shape model FSM of a given shape class $\mathcal{S}$ consists of a set of ordered rays

$$FSM(\mathcal{S}) = \{F_i, i = 1 \dots N\}.$$

Different from Section 3.1, the rays are now characterized by distributions of values (as opposed to values of the parameters)

$$F_i = (\mu_\theta^i, k_\theta^i, \mu_d^i, \sigma_d^i, \Lambda^i)$$

The distributions represent two classes of parameters. One class of parameters is used to model spatial distribution of object parts, i.e., μ_θ^i and k_θ^i describe the distribution of angles of ray i while μ_d^i and σ_d^i describe the distribution of lengths of ray i. Another class of parameters is used to represent local appearance of object part, i.e., Λ^i, which include the edge orientation. Details of these distributions are introduced in the following section.

Parameters estimation in fan shape model: Given a set of segmented objects (training shapes) $S_1, S_2, \dots, S_M$ in the same class $\mathcal{S}$, to learn fan shape model, we start with shape matching. First, reference point of first training shape S_1 is manually labeled. For all the other training shapes

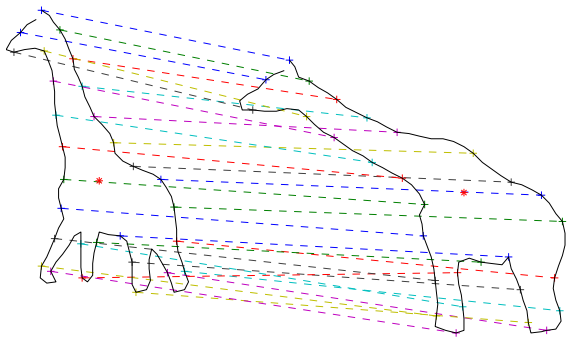

Figure 4. Shape matching result. Red stars mark the reference points. The dashed, colored lines connect pairs of matched points in two shapes.

$S_2, \ldots, S_M$, the reference points are automatically determined by shape matching so that they maximize the similarity to S_1. For each of the training shapes, we densely set many reference points. Then we use dynamic programming described in Section 3.2 to match the first shape for each of the reference points. Since we focus on learning variance of the inclined angle of rays, we set $\lambda_t = 0$ in (2). After all the set reference points have been tested, the one with minimum matching cost is chosen. A matching example is shown in Fig. 4. As revealed in the example, our designed feature with dynamic programming can provide a good points correspondence even though there is considerable variance between these two giraffe shapes.

We estimate parameters for each ray independently. For the ith ray in S_1 $(1 \leq i \leq N)$, after shape matching, there are $M-1$ rays from $S_2, \ldots, S_M$ matched to it. Therefore, M rays in total belong to the ith part of object in the training shapes, and they are given by $\{R_i(S_1), \ldots, R_{\phi(i)}(S_j), \ldots, R_{\phi(i)}(S_M)\}$, where we recall that $R_{\phi(i)}(S_j) = (\theta_{\phi(i)}, d_{\phi(i)}, \alpha_{\phi(i)})$ for $1 \leq j \leq M$.

We consider the distribution of distances $d_{\phi(i)}$ as a Gaussian distribution, i.e., $\mathcal{N}(d_{\phi(i)}, \mu_d^i, \sigma_d^i)$, and the distribution $\mathcal{M}$ over inclined angle $\theta_{\phi(i)}$ as the von Mises distribution [12],

$$\mathcal{M}(\theta_{\phi(i)}, \mu_\theta^i, k_\theta^i) \propto e^{k_\theta^i cos(\theta_{\phi(i)} - \mu_\theta^i)}$$

where μ_θ^i denotes mean value of all $\theta_{\phi(i)}$, k_θ^i is a measure of concentration of all $\theta_{\phi(i)}$. μ_d^i and σ_d^i are estimated via maximum likelihood estimation (MLE). Similarly, there is known method for finding the ML parameters $(\mu_\theta^i, k_\theta^i)$ of a von Mises distribution.

In this paper, we use two kinds of local descriptor. One is the edge orientation, another is the SIFT feature [14] of local image patches. Hence we represent Λ^i as

$$\Lambda_i = (\mu_\alpha^i, k_\alpha^i, T^i)$$

The edge orientation α ranges in $[0, \pi]$, and the distribution of 2α is a von Mises distribution $\mathcal{M}(\alpha, \mu_\alpha^i, k_\alpha^i)$.

We use SIFT features to estimate the appearance distribution of ith part. T^i is a collection of all the SIFT features at positions matching to the ith part in the training images. Hence T^i represents a discrete, empirical distribution of the SIFT features. A fixed patch size is used when computing SIFT. The probabilities according to T^i are computed based on the nearest neighbor (NN) classifier.

Fig. 5 illustrates our trained models of the five classes in ETHZ dataset [11].

4. Object Detection using Fan Shape Model

Object detection with FSM is composed of the following two main steps: 1. Fast scale estimation at a given image location and 2. Scoring of the detection hypothesis at the estimated scale. Both steps are described below and performed at every image location. The final detection is obtained after non maxima suppression of the score map.

4.1. Fast Object Scale Estimation

After we have learned fan shape model FSM of a given shape class $\mathcal{S}$, object detection is carried out by matching FSM to a test image. Unlike other methods searching among several scales via simply scaling model or testing image, FSM allows for a fast object scale estimation method. Given a test image I, we first compute edge image E of I. For every edge pixel, we compute edge orientation and a SIFT feature vector. Then, given a candidate location l of the reference point in image I, we collect all the edge pixels whose distance to l is smaller than the maximal expected radius r of the object. The edge pixels and their features are denoted as $Es = \{d_j, \theta_j, \alpha_j, t_j; 1 \leq j \leq ne\}$, where d_j and θ_j are the Euclidian distance and inclined angle of the jth edge pixel $e_j \in Es$ relative to l, α_j is the edge orientation, and t_j is a SIFT vector at the jth edge pixel. ne is the total number of edge pixels in Es. The probability of edge pixel e_j belonging to ray F_i at scale s is

$$\begin{aligned} p(e_j|F_i, s) = \mathcal{N}(d_j/s, \mu_d^i, {\sigma_d^i}^2) \cdot \mathcal{M}(\theta_j, \mu_\theta^i, k_\theta^i) \\ \cdot \mathcal{M}(2\alpha_j, \mu_\alpha^i, k_\alpha^i) \cdot NN(t_j, T^i) \end{aligned} \quad (3)$$

where $NN(t_j, T^i)$ is probability output by a nearest neighbor classifier which uses T^i as positive training instances and t_j as testing example. To estimate the object scale at l, we create a discrete scale space S_s containing ns possible scales $S_s = [s_1, ..., s_{ns}]$ and a voting space V_s that has the same size as S_s:

$$V_s[m] = \prod_{i=1}^{N} (\max_j (p(e_j|F_i, s_m))), \quad (4)$$

where $m = 1, \ldots, ns$ and we recall that N is the number of rays in FSM. The best scale s^* at l is given by

$$m^* = \arg\max_m (V_s[m]) \text{ and } s^* = s_{m^*}$$

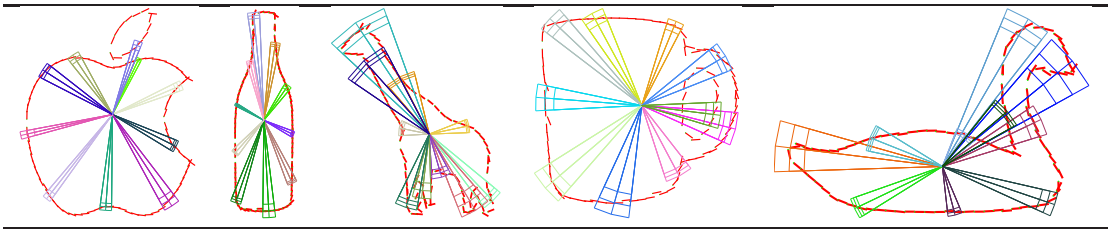

Figure 5. Trained models on ETHZ dataset. We only show a subset of the fan rays. The end of each red line indicates the mean position of one part given by μ_d^i and θ_d^i, while its direction shows the mean edge orientation, i.e., μ_α^i. Sectors with different colors are used to display the variations of the rays. For each sector, its angle shows the k_θ^i, and the difference between two radii of the same ray indicates σ_d^i.

Thus, for each location l, the best scale s^* is identified, and it will be used in the detection process described in Section 4.2.

We can efficiently compute V_s in (4). For edge pixel $e_j = \{d_j, \theta_j, \alpha_j, t_j\}$, first, it is unnecessary to compute $p(e_j|F_i, s_m)$ for every F_i. According to the probability of normal distribution, given F_i and e_j, if $|\theta_j - \mu_\theta^i|$ is larger than three times standard deviation of θ_i, $p(e_j|F_i, s_m)$ will be close to 0. Therefore, for those F_i, $p(e_j|F_i, s_m)$ will be set to be 0 directly. Second, given e_j, for F_i left, it is also unnecessary to compute $p(e_j|F_i, s_m)$ for every s_m. Only for s_m, which falls between $\frac{d_j - 3\sigma_d^i}{\mu_d^i}$ and $\frac{d_j + 3\sigma_d^i}{\mu_d^i}$, $p(e_j|F_i, s_m)$ is computed, otherwise it can be simply set to 0. We also observe that the computation cost of V_s will not increase with the size of scale space if the scale step is fixed.

For each location l and its estimated best scale s^*, we can obtain an object detection hypothesis which is a set of edge pixels $e^*(l) = (e_1^*, ..., e_N^*)$ such that

$$e_i^* = \arg\max_{e_i}(p(e_i|F_i, s^*)) \tag{5}$$

The final evaluation of hypothesis is presented in the next section.

4.2. Evaluation of Detection Hypothesis

During the process of finding an object detection hypothesis, each ray F_i selects the edge pixel e_i^* in image according to Eq. 5. All rays do this selection simultaneously, and has no interaction with choices of the other rays. This makes this process very efficient. In fact, our fan shape model encodes not only the spatial distribution and texture information for each ray, but also the order of all fans and their relative spatial relationship. We use this additional information about the adjacency of the rays in final evaluation.

Supported contour: If two neighboring rays choose two edge pixels connected by contour fragment in image, it is more likely to be a correct detection. Otherwise, the hypothesis may be an accidental matching due to clutter background. We define a contour support score as

$$\eta_1 = \frac{\Sigma_{i=1}^{N-1} slen(e_i^*, e_{i+1}^*)}{\Sigma_{i=1}^{N-1} len(e_i^*, e_{i+1}^*)}$$

let sp denotes a set of sample points between e_i^* and e_{i+1}^* in the line $\overline{e_i^* e_{i+1}^*}$, $len(e_i^*, e_{i+1}^*)$ is the number of points in sp and $slen(e_i^*, e_{i+1}^*)$ is the number of points in sp around which there is at least one edge pixel within a small distance from it. The distance is set to 2 pixels.

Distance consistency of points on adjacent rays: The distances between neighboring parts should be consistent, as in the training phase object contour is sampled with equidistant points. The sequence of distances between adjacent ray points (parts) is given by $d_{neib} = [|\overrightarrow{e_1^* e_2^*}|, ..., |\overrightarrow{e_{N-1}^* e_N^*}|, |\overrightarrow{e_N^* e_1^*}|]$. The distance consistency score is defined as $\eta_2 = \exp(-std(d_{neib}/s^*))$. The final score of a detection hypothesis at image location l is defined by

$$score(l) = \eta_1 \, \eta_2 \prod_{i=1}^{N} (p(e_i^*|F_i, s^*))$$

5. Experiments

We present results on the ETHZ shape classes [11] which features five diverse classes (Applelogos, Bottles, Giraffes, Mugs, Swans) and contains a total of 255 images. For all categories, there is significant inner-class variance, shape deformation, scale change, and some of images have very clustered background.

We follow the train/test split described in [22]. The first half of images in each class is used for training the models. The outline images in these halves are used to do shape matching and learning parameters of the fan shape models, grayscale images are used to extract SIFT features. Our learned models are shown in Fig. 5.

To convert the gray level edge map to binary edge map, we set all pixels with their values larger than 0.02*255 as edge pixels. This means we do not adjust the threshold to get better edges. During detection, since the time complexity of our algorithm does not increase as the scale space in-

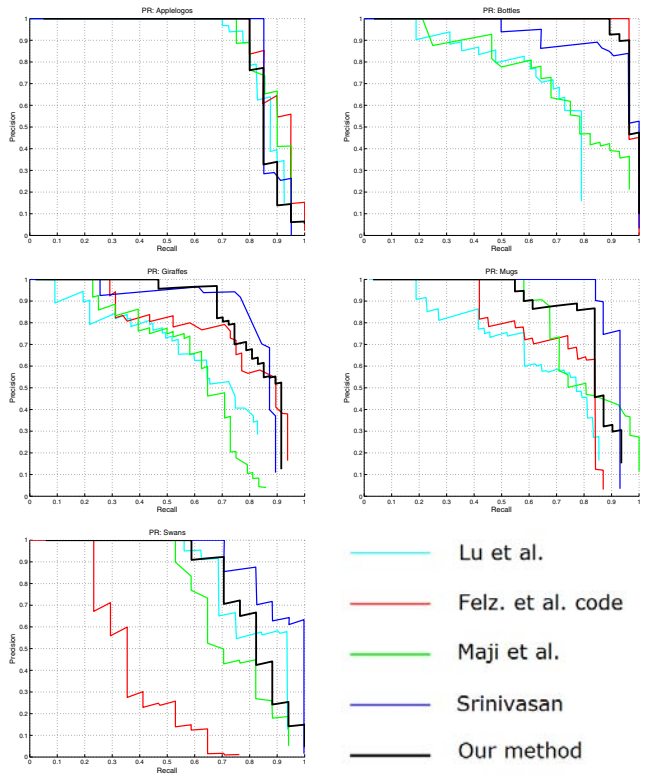

Figure 6. Precision/Recall curves of our method compared to Lu *et al*. [15], Felz *et al*. [8], Maji *et al*. [17], and Srinivasan *et al*. [22] on ETHZ shape classes.

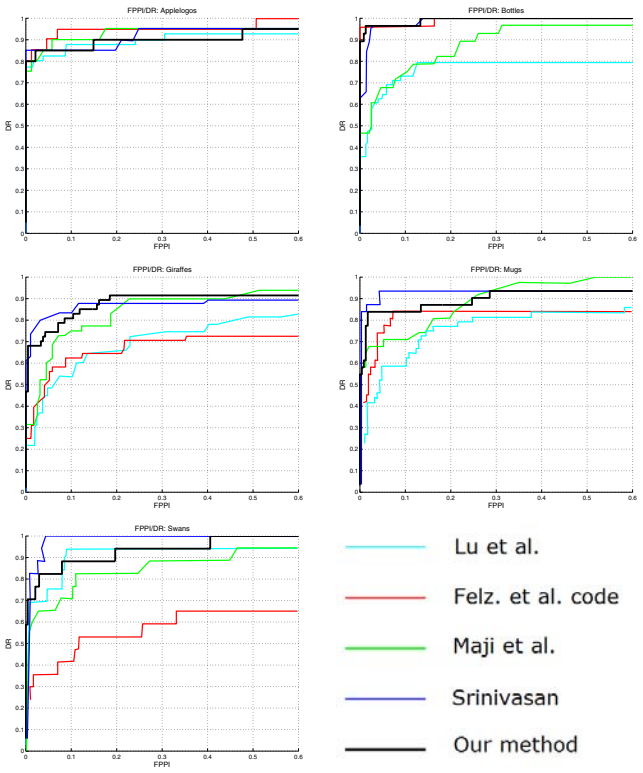

Figure 7. DR/FPPI curves of our method compared to Lu *et al*. [15], Felz *et al*. [8], Maji *et al*. [17], and Srinivasan *et al*. [22] on ETHZ shape classes.

creases, up to 15 scales have been searched. Non-maximum suppression is used to remove duplicate hypothesis.

For the purpose of detection evaluation, we follow the PASCAL criteria, *i.e*., a detection is deemed as correct if the intersection of detected bounding box and ground truth over the union of the two bounding boxes is larger than 50%.

We compare our method with the popular contour based object detection methods [10, 22], and the texture based discriminative part model [8], all these methods use the same training and testing images as we do. We plot the precision/recall (PR) curves in Fig. 6. We use the toolbox in [7] to calculate average precision (AP), Table 1 shows the AP value for 5 methods. The mean AP of our method is comparable to the state of the art method [22] and much better than the other methods.

Among the compared methods, we have achieved best AP for category Giraffes, which is the most difficult class in this dataset. In particular, we significantly outperform [8]. Both [8] and our method are part based methods, where parts are described with texture features. This results demonstrate that our fan shape model is a more flexible part model than the part model in [8].

We also show the false positives per image (FPPI) vs. detection rate (DR) in Fig. 7. Table 2 compares our detection rates at 0.3/0.4 FPPI with [22, 17, 8, 15, 20, 10, 25]. Our method achieves a comparable result to [22]. We observe that our method is the only one with no difference in detection rates at 0.3 FPPI and 0.4 FPPI. The curve of our method increases sharply at the beginning and reaches the peak of the detection rate before 0.3/0.4 FPPI.

Fig. 8 shows some of our detection results, both true positives and false positives are showed. Both internal and external contours (*e.g*. mug handle/outline) can be detected.

6. Conclusions and Future Work

As can be observed from our experimental evaluation, the detection rate of the proposed fan shape model is very good. The excellent detection results on selected images from PASCAL dataset [7] also confirm this fact. However, the final evaluation of the detection results requires further work. In particular, the constraints of supported contours and distance consistency are only evaluation after the detection has been completed. Thus, if part of the detected object has been corrupted by wrong edge pixels, the assigned detection score is low. It would be desirable to include these constraints in the detection phase, e.g., with dynamic pro-

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Our method	0.866	**0.975**	**0.832**	0.843	0.828	0.869
Srinivasan et al. [22]	0.845	0.916	0.787	**0.888**	**0.922**	**0.872**
Maji et al. [17]	0.869	0.724	0.742	0.806	0.716	0.771
Felz et al. code [8]	**0.891**	0.950	0.608	0.721	0.391	0.712
Lu et al. [15]	0.844	0.641	0.617	0.643	0.798	0.709

Table 1. Comparison of average precision (AP) on ETHZ Shape classes.

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Our method	0.90/0.90	**1 / 1**	**0.92/0.92**	**0.94**/0.94	0.94/0.94	0.940 / 0.940
Srinivasan et al. [22]	**0.95/0.95**	**1 / 1**	0.872/0.896	0.936/0.936	**1 / 1**	**0.952 / 0.956**
Maji et al. [17]	0.95/0.95	0.929 / 0.964	0.896/0.896	0.936/**0.967**	0.882 / 0.882	0.919 / 0.932
Felz et al. code [8]	0.95/0.95	**1 / 1**	0.729/0.729	0.839/0.839	0.588 / 0.647	0.821 / 0.833
Lu et al. [15]	0.9/0.9	0.792 / 0.792	0.734/0.77	0.813/0.833	0.938 / 0.938	0.836 / 0.851
Riemenschneider et al. [20]	0.933/0.933	0.970 / 0.970	0.792/0.819	0.846/0.863	0.926 / 0.926	0.893 / 0.905
Ferrari et al. [10]	0.777/0.832	0.798 / 0.816	0.399/0.445	0.751/0.8	0.632 / 0.705	0.671 / 0.72
Zhu et al. [25]	0.800/0.800	0.929 / 0.929	0.681/0.681	0.645/0.742	0.824 / 0.824	0.776 / 0.795

Table 2. Comparison of detection rates for 0.3/0.4 FPPI on ETHZ Shape classes.

gramming. However, this significantly increases the detection time. Therefore, our future work will focus on efficient methods for incorporating such constraints.

Acknowledgement

This work received support from the National Natural Science Foundation of China (grant No. 60903096, 61173120, 60903172), the NSF under Grants IIS-0812118, BCS-0924164, OIA-1027897, and the AFOSR Grant FA9550-09-1-0207.

References

[1] X. Bai, X. Wang, L. J. Latecki, W. Liu, and Z. Tu. Active skeleton for non-rigid object detection. *ICCV*, 2009. 2, 3

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. 2

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986. 1

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms. *MIT Press, 2nd edition*, 2001. 3

[5] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, pages 10–17, 2005. 2

[6] D. J. Crandall and D. P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV*, pages 16–29, 2006. 2

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html. 1, 2, 6

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *CVPR*, 2008. 6, 7

[9] P. Felzenszwalb and R. Zabih. Dynamic programming and graph algorithms in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4), April 2011. 2

[10] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *Int Journal of Computer Vision*, 2010. 1, 2, 6, 7

[11] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. *ECCV*, 2006. 1, 2, 4, 5

[12] E. J. Gumbel, J. A. Greenwood, and D. Durand. The circular normal distribution: Theory and tables. *Journal of the American Statistical Association*, 1953. 4

[13] P. D. Kovesi. Matlab and octave functions for computer vision and image processing., 2008. 2

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int Journal of Computer Vision*, 2004. 4

[15] C. Lu, L. J. Latecki, N. Adluru, X. Yang, and H. Ling. Shape guided contour grouping with particle filters. *ICCV*, 2009. 1, 2, 6, 7

[16] T. Ma and L. Latecki. From partial shape matching through local deformation to robust global shape similarity for object detection. In *CVPR*, pages 1441 –1448, june 2011. 2

[17] S. Maji and J. Malik. A max-margin hough tranform for object detection. *CVPR*, 2009. 6, 7

[18] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):530 –549, May 2004. 1, 2

[19] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment model for object detection. *ECCV*, 2006. 2

[20] H. Riemenschneider, M. Donoser, and H. Bischof. Using partial edge contour matches for efficient object category localization. *ECCV*, 2010. 6, 7

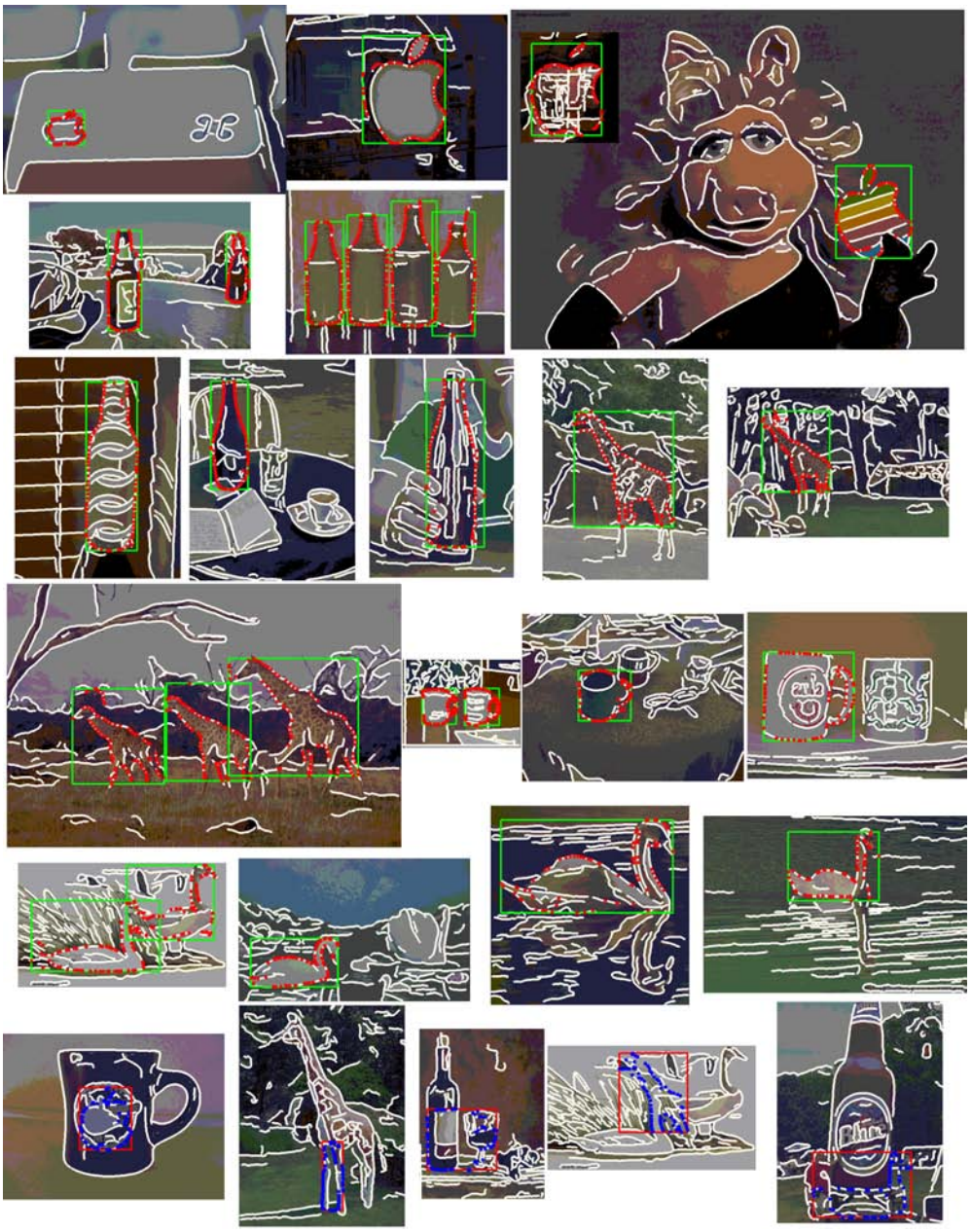

Figure 8. Some of our detection results on the ETHZ shape categories dataset. Each image shows detected edge pixels and bounding boxes for one or more detections. Bottom row shows false positives for Applelogos, Bottles, Giraffes, Mugs and Swans (l-to-r); Edge pixels are plotted in white, true positives are plotted with red points and green bounding boxes, false positives are plotted with blue points and red bounding boxes.

[21] J. Shotton, A. Blake, and R. Cipolla. Multi-scale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 90(7):1270–1281, 2008. 2

[22] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. *CVPR*, 2010. 1, 2, 5, 6, 7

[23] N. H. Trinh and B. B. Kimia. Category-specific object recognition and segmentation using a skeletal shape model. *BMVC*, 2009. 3

[24] W. Zhang, P. Srinivasan, and J. Shi. Discriminative image warping with attribute flow. In *CVPR*, pages 2393 –2400, june 2011. 2

[25] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. *ECCV*, 2008. 1, 2, 6, 7