Weakly Supervised Shape Based Object Detection with Particle Filter

Xingwei Yang and Longin Jan Latecki

Dept. of Computer and Information Science
Temple University, Philadelphia, 19122, USA
{xingwei.yang,latecki}@temple.edu

Abstract. We describe an efficient approach to construct shape models composed of contour parts with partially-supervised learning. The proposed approach can easily transfer parts structure to different object classes as long as they have similar shape. The spatial layout between parts is described by a non-parametric density, which is more flexible and easier to learn than commonly used Gaussian or other parametric distributions. We express object detection as state estimation inference executed using a novel Particle Filters (PF) framework with static observations, which is quite different from previous PF methods. Although the underlying graph structure of our model is given by a fully connected graph, the proposed PF algorithm efficiently linearizes it by exploring the conditional dependencies of the nodes representing contour parts. Experimental results demonstrate that the proposed approach can not only yield very good detection results but also accurately locates contours of target objects in cluttered images.

1 Introduction

Object recognition, detection, and localization in real images is a major problem in Computer Vision since its beginning. In the last few years, the majority of existing methods use simple relations of local image patches as basic features, e.g., [24,3]. They can perform very well on high textured objects, but they are unable to identify parts of deformable objects nor precisely localize their boundaries in images. The main reason is that the model fails to represent all available information [25]. However, an improved, richer representation of deformable objects is only useful when it is accompanied by efficient techniques for performing inference and learning [26]. Thus, progress in this area requires to simultaneously develop more powerful representations together with efficient inference algorithms.

In this paper, we propose a single layer fully connected graph to model shape of deformable objects. Each node in the graph is a state variable, which consists of the position and the corresponding part. The relation between nodes is long range and not limited to direct spatial proximity. Our model can be interpreted as a generative prior for the configuration of the state variables. Since our graph is fully connected, we do not need to learn its structure, which simplifies

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part V, LNCS 6315, pp. 757–770, 2010.

the learning significantly. We only need to learn representation of the nodes and their pairwise relations. Since the number of pairwise relations is large, and most of them are not used in our inference process, we do not learn the pairwise relations explicitly. Instead, we learn a representation that allows us to dynamically construct the pairwise relations needed in the inference process.

In our model graph, the nodes represent contour parts and their position in a given shape class. They are learned automatically with partially-supervised learning. While many state-of-the-art approaches construct part models manually [18,27], we limit manual labeling to a single contour. In our approach, only one silhouette is manually decomposed into visual parts in advance. Then, the part decomposition is automatically transferred to silhouettes not only in the same class but also in different classes with similar shape by shape matching. To deal with non-rigid objects, we use Inner Distance Shape Context (IDSC) introduced in [17]. The constructed part bundles (see §3) with proper position in the exemplar shapes form the nodes in the model graph. The relations between the nodes represent the spatial layout between parts. It is described by nonparametric density estimation, which has better discriminative power than methods based on unimodal distributions modeled as Gaussians, e.g., [5,23]. To make the learnt model graph representative, we use the well designed exemplar based clustering by Affinity Propagation [8] to select a set of candidate silhouettes as exemplars for our model learning approach.

According to [26], there are no known algorithms for performing inference for densely connected flat models, e.g., the performance of Belief Propagation (BP) is known to degrade for representations with many closed loops. To address this issue, we propose a Markov chain Monte Carlo (MCMC) approach that is able to efficiently infer the values of the state variables representing nodes of our fully connected model graph. The proposed MCMC approach is based on Particle Filter (PF), but it differs fundamentally, since unlike the standard PF framework, our PF framework can infer an order of random variable (RVs). The inferred order follows the most informative paths in the graph. Thus, we use PF to linearize the structure of the graph, which allows us to avoid the problem of loops. Each particle may explore a different node order in this linearization, which corresponds to the order of contour parts. This fact is illustrated by two different detection examples shown in Fig. 1, where the PF order of detected contour parts is color coded. This property makes our algorithm different from other PF based method [13,12]. As can be seen by examining the relative position of consecutive parts, the proposed inference is not limited to direct spatial proximity of the parts. This fact sets our approach apart from existing approaches, e.g., [26,14].

In order to show the advantages of the proposed approach, we test our method on three widely used data sets, Weizmann horses [2], the ETHZ [6], and the cow dataset from the PASCAL Object Recognition Database Collection (TU Darmstadt Database [16]). Our results measured by bounding box intersection are comparable to state-of-the-art methods. Also, we perform very well in the accuracy of boundary localization, which is evaluated by a recently proposed measure in [7].

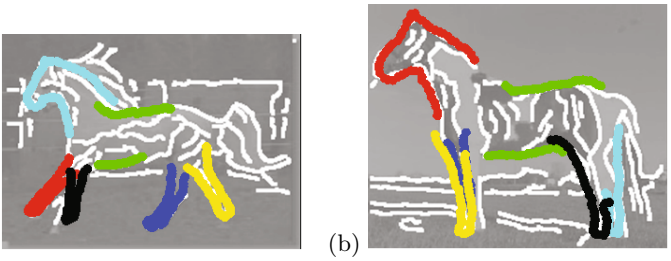


Fig. 1. Examples of two different inferred orders of detected contour parts. Colors represent the order, which is 1=red, 2=cyan, 3=blue, 4=green, 5=yellow, and 6=black.

2 Related Work

Ferrari et al. [7] propose to learn the model from real images with weakly supervision. Given the bounding boxes, the model is considered as the common pattern of objects in the same class. With the same intuition, Lee and Grauman [15] also treat the common pattern in a class as the model. However, their method is totally unsupervised. To utilize the already learnt information, Stark et al. [23] transfer information of the learnt model to better study the new model by a probabilistic framework. They have very similar intuition with our method. However, ours are quite different from theirs. We transfer the structure information by pure shape matching without any statistics. Their method is mainly based on the probabilistic model they construct.

To detect objects in the cluttered image, Ferrari et al. [6] use kAS with Hough voting to estimate the position of objects. Ommer and Malik [20] propose a novel Hough voting strategy to overcome the problem of scales. Zhu et al. [27] treat the detection as a set-to-set matching problem between segments. They simplify the problem into linear programming to reduce the complexity. Ravishankar et al. [21] propose a multi-stage method with manually deformed model. Similar to ours, Trinh and Kimia also learn the model from silhouettes. However, instead of contours, they use a skeleton based generative shape model. Also, their detection stage is using dynamic programming, which is quite different from our method. Besides pure shape based method, Maji and Malik [19] propose a maximum margin hough voting method with SVM to detect objects. Gu et al. [11] combine the region and shape together for object detection.

Particle filter (PF) has been used for object detection previously [13,12]. They mainly utilize PF to reduce the possible assumptions and they have pre-defined the order for PF. However, our method can determine the order of PF on the fly, which is theoretically quite different from the traditional PF. Moreover, we are based on shape features for object detection instead of the binary classifier they defined. Lu et al. [18] also use PF for shape based object detection. However, we are totally different from them at the proposal and evaluation steps, which is essential for PF. Also, the pairwise relation between parts is naturally embedded into our PF framework, which has not been done in the previous PF methods.

3 Partially-Supervised Model Learning

Our approach only requires marking object parts on one exemplar. We then transfer this knowledge to other contours not only in the same shape class but also to similar shape classes. Thus, our approach is able to construct the part models for different classes of objects starting with only one exemplar contour. The constructed model can describe a wide range of objects with different poses.

As we learn the model from exemplars, the first issue is which ones should be chosen from a given training data set. We use Affinity Propagation to select the exemplars, which are cluster centers in AP. These cluster centers are representative, so that they can describe most of the poses of objects. The input pairwise distance between shapes is obtained by Oriented Chamfer Matching (OCM).

3.1 Part Model Construction

In this section we describe a way to automatically decompose the exemplars $E = \{E_1, \ldots, E_{N_e}\}$ into meaningful parts. We first manually segment one selected silhouette, say E_1 into m different meaningful parts $S = \{s_1, \ldots, s_m\}$. For example, for horse, we have six parts: head, two front legs, two back legs, and the body, shown in different colors in top left of Fig. 2(a). We then use shape matching with IDSC [17] to transfer the parts to other exemplars $E_2, \ldots, E_{N_e}$, e.g., to the second horse in Fig.2(a). The corresponding points carry over the part decomposition. To ensure that the part decomposition is transferred correctly, we require that the number of corresponding points for a given contour part s_i is larger than a given threshold, e.g. 80% of the total number of points in the contour part. If this is not the case, the corresponding part is removed from the model.

We define part bundle B_i as a set composed of part s_i on E_1 and all corresponding parts on $E_2, \ldots, E_{N_e}$ transferred by the IDSC matching for $i = 1, \ldots, m$. Each part bundle B_i has at most N_e contour parts. We obtain a set of m part bundles $B = \{B_1, B_2, ..., B_m\}$ that defines the nodes of our part model graph.

We can also employ shape matching to transfer the part structure to different but similar object classes. As illustrated in Fig. 2(a), our part decomposition of the horse contour transfers easily to contours of giraffes. As long as the objects in different classes have similar structure, the proposed approach can transfer the structure knowledge from the known class to the other classes and obtain the part bundle models. There are three advantages of the proposed approach: 1) It requires very little manual labeling. 2) The constructed model composed of part bundles can handle the intra-class variations as long as the training silhouettes can represent the possible poses of objects. 3) The structural knowledge can be easily transferred to different classes.

3.2 Relation between Model Parts

After learning the model from silhouettes, in order to make the model more flexible, we permit the rotation for each part and also some shift. However, with

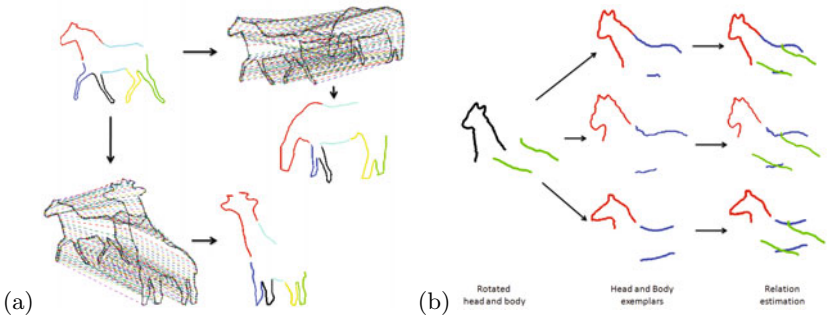


Fig. 2. (a) Six manually labeled parts on the horse in top left are marked with different colors. The point correspondence obtained by shape matching allows us to transfer the part structure to a different horse and to a giraffe. (b)The horse head and horse body shown on the left hand side are very different from our perception of a horse. Our measure of this fact is illustrated in the rest of this figure.

the increasing flexibility, the obtained model can be very different from shapes in a given object class. To reduce the negative effect of flexible models, we propose a soft way to constrain the flexibility. We allow the flexibility in a range determined by shape similarity to example shapes in a given object class. Here the shape similarity is described by spatial layout of model parts, i.e., a new rotated spatial layout of parts is allowed if it is similar to a layout previously seen for this class. An example is shown in Fig. 2(b). The horse head and horse body shown on the left hand side are very different from our perception of a horse. The head and body are too far away from each other and their arrangement due to rotation is really strange. With the method described below, we can offer a soft constraint on possible spatial layout of parts.

The key idea is to construct a distribution describing the spatial layout between different parts. In particular, given a part bundle B_i, the spatial relation between it and another part bundle B_j forms a distribution. This kind of distribution has been used in object detection to help describe the model [23,5], but the distribution is assumed to be Gaussian, whose parameters can be easily learned from training samples. However, obviously, the distribution of part relation is very complex and expressing it as Gaussian or any other parametric distribution does not seem to be a good approximation. Instead, we propose to learn the underlying distribution in a non-parametric setting.

We employ kernel density estimation, which is one of the most popular non-parametric methods. Given are two rotated parts p'_i and p'_j that come from different part bundles B_i and B_j respectively. Our goal is to find how is p'_j located with respect to p'_i. For example, we want to find out how well the green body is positioned with respect to the black horse head in Fig. 2(b). For part p'_i, we use $OCM_{p'_i}$ to find the top k most similar exemplar parts $(p_i(1), \ldots, p_i(k))$ in part bundle B_i (the bundle of p'_i). For these original parts in B_i, we know the exemplar contours they came from. From these contours, we extract parts

$(p_j(1), \ldots, p_j(k))$ that belong to the same bundle as p'_j, i.e., to part bundle B_j. In Fig. 2(b), OCM retrieves the 3 red horse heads $(p_i(1), p_i(2), p_i(3))$ as most similar to the black head, which in turn carry over from their original contours 3 blue horse bodies $(p_j(1), p_j(2), p_j(3))$. Finally, we measure the spatial layout between parts p'_i and p'_j by estimating the fitness of p'_j to the distribution described by $(p_j(1), \ldots, p_j(k))$:

$$f(p'_j|p'_i) = \frac{1}{C_c} \sum_{t=1}^{k} \frac{1}{h} K(\frac{OCM_{p'_j}(p_j(t))}{h}) \tag{1}$$

where K is a kernel function with bandwidth h, which is Gaussian in the paper and C_c is a constant value. The computation of $f(p'_j|p'_i)$ in our example is illustrated in the right column of Fig. 2(b). It is a function of the OCM distance between the green horse body and the 3 blue horse bodies.

4 Framework for Object Detection

Our goal is to infer the maximum of a posterior distribution $p(B_1, \ldots, B_m \mid Z)$, where $(B_1, \ldots, B_m)$ is a vector of random variables (RVs) representing part bundles, which are nodes of our shape model graph (§3). In our application $Z = (I, C)$ is a set of observations, where I is a RV ranging over binary edge images and C ranges over classes of target objects including background. Thus, Z is static, since the target edge image and the class of object are fixed for a given detection process. The possible values of each RV B_i are vectors of two elements, one is the location x_i in the image and the second is the part s_i chosen from the part bundle B_i in the model. In the case of a correct detection, we expect part s_i to be located at x_i in the image. We stress that even though each part bundle has many parts, only one of them is chosen for a given location in the image. To simplify the notation, we use b to represent the pair of values (x, s) for each random variable, i.e., $b_l = (x_l, s_l)$. Consequently, our goal is to find value assignments to RVs $B_t = b_t$ for $t = 1, \ldots, m$ that maximize the posterior

$$\widehat{b}_{1:m} = \underset{b_{1:m}}{\operatorname{argmax}}\ p(b_{1:m} \mid Z), \tag{2}$$

where $b_{1:m}$ is a shorthand notation for $(b_1, \ldots, b_m)$. We will achieve our goal by approximating the posterior distribution with a finite number of particles in the framework of Particle Filter (PF). Besides, only a small subset of the search space is considered in the framework, which reduces the complexity significantly compared to exhaustive search with sliding windows, e.g., [22].

Unlike the standard PF framework, the observations Z in our approach do not arrive sequentially, but are available at once, i.e., Z is static. Therefore, the observations have no natural order. Consequently, the states $b_{1:m}$ also do not have any natural order, i.e., the order of indices $1, \ldots, m$ does not have any particular meaning. Therefore, we need to extend the PF framework to infer an order of RVs, which may be different for each particle. Intuitively, we want to

determine such an order of RVs so that the corresponding order of observations is most informative, which makes the particle reaches optimal solution faster and more accurate. This makes the proposed PF fundamentally different from classical PF. To represent the order of RVs we need a symbol of a bijection (onto and one-to-one function) $< \cdot >^{(i)}$: $\{1, \ldots, m\} \rightarrow \{1, \ldots, m\}$. Although we may have a different bijection for each particle (i), we will drop the index (i) from $< 1 : t >^{(i)}$, since the state variables already carry the particle index. For example, we denote $(b_4^{(i)}, b_5^{(i)}, b_2^{(i)})$ as $b_{<1:3>}^{(i)}$, where $< 1 : 3 >= (4, 5, 2)$.

We first present the proposed PF algorithm followed by a discussion of its major differences to standard PF approaches. As it is often the case in PF applications, we assume the proposal distribution to be $q(b|b_{<1:t-1>}^{(i)}, Z) = p(b|b_{<1:t-1>}^{(i)})$. For each particle (i), where $i = 1, \ldots, N$, the proposed PF algorithm in each iteration $t = 2, \ldots, m$ performs the following three steps:

1) **Importance sampling / proposal:** Sample followers of particle (i) for $l \in \{1, \ldots, m\} \backslash < 1 : t-1 >$

$$b_l^{(i)} \sim p(b_l|b_{<1:t-1>}^{(i)}) \tag{3}$$

and set $b_{<1:t-1>,l}^{(i)} = (b_{<1:t-1>}^{(i)}, b_l^{(i)})$. In particular, in the first iteration $(t = 1)$ we generate samples from each dimension of the state space, i.e., we sample for $l \in \{1, \ldots, m\}$

$$b_{<1>}^{(i)} = b_l^{(i)} \sim p(b_l) \tag{4}$$

2) **Importance weighting/evaluation:** An individual importance weight is assigned to each follower of each particle by

$$w(b_{<1:t-1>,l}^{(i)}) = p(Z|b_{<1:t-1>,l}^{(i)}). \tag{5}$$

3) **Resampling:** At the sampling step we have generated more samples than the number of particles. Thus we have a larger set of particles $b_{<1:t-1>,l}^{(i)}$ for $i = 1, \ldots, N$ and $l \in \{1, \ldots, m\} \backslash < 1 : t-1 >$ from which we sub-sample N particles and assign equal weights to all of them as in the standard Sampling Importance Resampling (SIR) approach. We obtain a set of new particles $b_{<1:t>}^{(i)}$ for $i = 1, \ldots, N$. The resampling is not performed in the last step, i.e., when $t = m$.

Algorithm discussion:
1) This step provides our main extension of the classical PF framework. In the classical PF framework, followers of each particle are selected from only one conditional distribution, i.e., from the conditional distribution of RV at dimension t given by $p(b_t|b_{1:t-1}^{(i)})$, since the dimension index t represents a real order of RVs $1 : t = 1, \ldots, t$. In contrast we sample the followers from each dimension $l \in \{1, \ldots, m\}$ that is not already included in $< 1 : t-1 >$.

The fact that one can consider more than one follower of each particle and reduce the number of followers by resampling is known in the PF literature and

is referred to as prior boosting [10]. It is used to capture multi-modal likelihood regions. However, all followers are selected from the conditional distribution of the same RV (the same dimension t) in the classical PF framework.
2) We take the weight formula from [18], where it has been derived for PF with static observations.
3) We stress that the resampling plays in our framework an additional and a very crucial role. It selects the the most informative random variables (i.e., state space dimensions) as followers of particles. Since the weight of $b^{(i)}_{<1:t-1>,l}$ is determined by the observations Z, and the resampling uses the weights to selects a follower $b_{<t>} = b_l$ from not yet considered dimensions $l \in \{1, \ldots, m\} \setminus < 1 : t-1 >$, the resampling determines the order of RVs, i.e., the bijection $< t >$ for $t = 1, \ldots m$. Consequently, the order of RVs is heavily determined by Z, and this order may be different for each particle (i). This is in strong contrast to the classical PF, where observations Z have no influence on the order of RVs, which is fixed.

In order to execute the derived PF algorithm, we need to define the proposal distribution $p(b_l|b^{(i)}_{<1:t-1>})$, and the evaluation pdf $p(Z|b^{(i)}_{<1:t-1>,l})$. As stated in Eq. 4, the initial proposal distribution is defined by $p(b_l)$, where l is an index of a RV representing a part bundle and $b_l = (s_l, x_l)$. In our implementation, $p(b_l)$ is simply the probability of finding model part s_l at location x_l, and it measures how well model part s_l fits the edges in the image. We compute it as a Gaussian of the oriented chamfer distance. Similarly, $p(b_l|b^{(i)}_{<1:t-1>})$ is the probability of finding model part s_l at the location x_l, but now the location is constrained, since parts $s_{<1:t-1>}$ have already been placed in the image. Thus, this conditional probability is picked around the expected location x_l determined by the locations $x_{<1:t-1>}$ of the previously added parts. While the initial proposal distribution is computed at every image location, the conditional proposal distribution is only computed at regions of interest determined by the previously placed model parts.

As $Z = (I, C)$, and I and C can be viewed as independent conditioned on $b^{(i)}_{<1:t-1>,l}$, we obtain:

$$p(Z|b^{(i)}_{<1:t-1>,l}) = p(I|b^{(i)}_{<1:t-1>,l})p(C|b^{(i)}_{<1:t-1>,l}) \tag{6}$$

We recall that in our detection framework, both I and C are instantiated, since they are given prior to the detection, i.e., $I = im$, where im is a given binary edge image and $C = 1$, which represents the class of the target object. The first factor $p(I = im|b^{(i)}_{<1:t-1>,l})$ in Eq. 6 describes the goodness of fit to the edge image im of the partial shape model determined by $b^{(i)}_{<1:t-1>,l}$, i.e., how likely the edges in im come from a picture of a shape like the shape of $b^{(i)}_{<1:t-1>,l}$. The second factor $p(C = 1|b^{(i)}_{<1:t-1>,l})$ represents the probability of the target class given the model $b^{(i)}_{<1:t-1>,l}$. Hence it can be viewed as shape class constraints on the model. The conditional pdfs describing both factors are defined in § 5.

5 Evaluation Based on Shape Similarity

As $b^{(i)}_{<1:t-1>,l}$ consists of the parts $s^{(i)}_{<1:t-1>,l}$ and their locations $x^{(i)}_{<1:t-1>,l}$, we construct a partial shape model μ by putting parts $s^{(i)}_{<1:t-1>,l}$ at locations $x^{(i)}_{<1:t-1>,l}$ on the edge map im. The probability that the edge map im is an image of a real object looking like our partial model μ is given by

$$p(I = im|b^{(i)}_{<1:t-1>,l}) = \exp(-\beta \cdot OCM_{im}(\mu)), \tag{7}$$

where $OCM_{im}(\mu)$ returns the Oriented Chamfer distance between im and μ and β is set to 10. Consequently, $OCM_{im}(\mu)$ measures how well the constructed partial model matches to the edge map.

$p(C = 1|b^{(i)}_{<1:t-1>,l})$ expresses the probability of the target shape class given partial shape model $\mu = b^{(i)}_{<1:t-1>,l}$. We obtain by Bayes rule

$$p(C = 1|\mu) = \frac{p(\mu|C = 1)p(C = 1)}{\sum_{c=1,0}\ p(\mu|C = c)p(C = c)}. \tag{8}$$

$p(\mu|C = 1)$ measures the similarity between the constructed model and the target class. Similarly, $p(\mu|C = 0)$ measures the similarity between the constructed model and the background. Eq. 8 helps to prevent accidental match to the background, since it eliminates shape models with both high similarity to a given object class and to the background, and favors models with high similarity to a given object class and low similarity to the background. We utilize a recursive computation in our PF framework to obtain

$$\begin{aligned} p(\mu|C = c) &= p(b^{(i)}_{<1:t-1>,l}|C = c) \\ &= p(b^{(i)}_{l}|b^{(i)}_{<1:t-1>}, C = c)\ p(b^{(i)}_{<1:t-1>}|C = c) \\ &= p(b^{(i)}_{l}|b^{(i)}_{<t-1>}, C = c)\ p(b^{(i)}_{<1:t-1>}|C = c) \\ &= f(b^{(i)}_{l}|b^{(i)}_{<t-1>})\ p(b^{(i)}_{<1:t-1>}|C = c), \end{aligned} \tag{9}$$

where f is defined in Eq. 1, and a given shape class $C = c$ is modeled as a set of exemplars $E = \{E_1, \ldots, E_{N_e}\}$, which are selected from training examples by affinity propagation.f describes the pairwise relation between nodes in the graph, which is naturally utilized in our PF framework. When $C = 0$, we randomly select some background edge configurations as training examples. In the transition from 2nd to 3rd row in Eq. 9, we make a Markov assumption that the new model part $b^{(i)}_{l}$ only depends on the previously added part $b^{(i)}_{<t-1>}$ conditioned that we know the shape class $C = c$. This simplifies the computation and makes the shape model more flexible in that the pose of the new model part is only evaluated with respect to the pose of previously added part. Finally, $p(b^{(i)}_{<1:t-1>}|C = c)$ is remembered from the previous iteration of particle (i).

6 Experimental Results

We have tested our algorithm on three widely used data sets: the extended Weizmann Horses [2,22], the ETHZ shapes [7] and the TU Darmstadt Database [16]. During the testing for Weizmann Horses, only 12 automatically selected horse silhouettes with one hand decomposed horse are used to learn the shape model. All the other images are used for testing. The edge maps for this dataset are obtained by Canny edge detector. We also test our method on the class of giraffe in ETHZ shape dataset [7]. The reason why we only select the category giraffes from ETHZ is that our model learning method can only transfer between objects with similar structure and giraffe is the only object in ETHZ having similar structure to horse. Only one hand decomposed horse and 6 automatically selected giraffe silhouettes are used to learn the giraffe model. Further, we work on the cow dataset the TU Darmstadt Database [16], since cows have similar structure with the above two classes. It contains 111 images. Only one hand decomposed horse and 6 automatically selected cow silhouettes are used to learn the cow model. The edge maps for this dataset are obtained by Canny edge detector.

To adapt to large scale variance, we generate multiple models by resizing the original ones to 5 to 8 scales, and choose as the final result from the best score in all the scales. We not only report our results on the commonly used bounding box intersection, but also the accuracy of our boundary localization.

6.1 Detection according to Bounding Boxes

We first evaluate the ability of the proposed approach to localize objects in cluttered images using bounding-box intersection, which is widely used in traditional object detection task. We adopt the strict standards of PASCAL Challenge criterion: a detection is counted as correct only if the intersection-over-union ratio with the ground-truth bounding-box is greater than 50%.

Fig. 3 reports precision-recall (P/R) curve and detection rate vs false positive per image (DR/FPPI) curve for the class Giraffes in ETHZ dataset. In P/R, we compare to Lu et al. [18], Zhu et al. [27], Ommer and Malik [20] and Ferrari et al. [7], whose results are quoted from [18]. In DR/FPPI, as Ferrari et al. [7,6], Ommer and Malik [20] and Lu et al. [18] provide their results, we compare to them. As Ravishankar et al. [21] do not give their curves, we do not compare to them in Fig. 3. According to the curves, we are better than Lu et al. [18], Ommer and Malik [20], Ferrari et al. [7,6] and perform equally well as Zhu et al. [27]. The performance of the proposed method illustrates its ability to cope with substantial nonrigid deformations, which are present in the class Giraffes. This is demonstrated by our example results in Fig. 4(a).

Table 1 compares our detection rate to [26,22] on Weizman Hores and TU Darmstadt Cows. The detection rate on horses is estimated from the DR/FPPI curve in [22]. The DR/FPPI curve for cows is not available in [22]. The method in [26] is also matching based, while [22] is a classification method. Some examples of our horse and cow detection results are shown in Fig. 4(b). The detection precision/recall area under curve (AUC) is a standard performance measure on

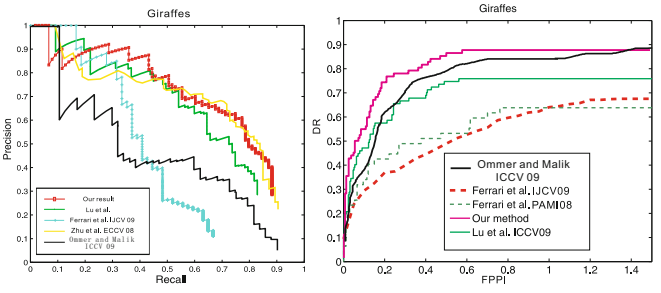
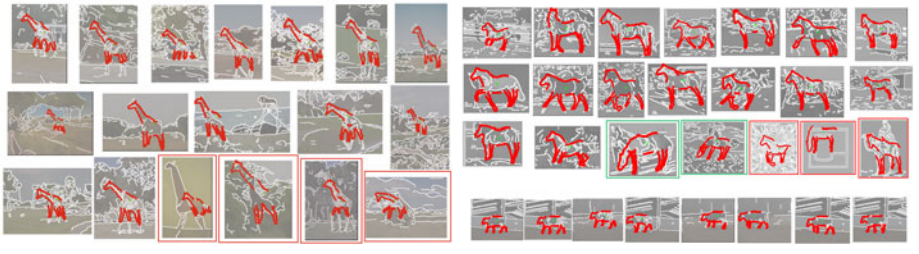


Fig. 3. Precision-recall curve and detection rate (DR) vs false positive per image (FPPI) curve for the class Giraffes in ETHZ dataset



(a) (b)

Fig. 4. Examples of detection results for Giraffes, horses and cows

the Weizmann Horses dataset. The AUC for our approach is 79.84%, which is comparable to the result 80.32% in Xiang et al. [1]. We compare to them as they also use the explicit shape model and matching based method for object detection. The AUC of classification based methods [22,9] is 84.98% and 96%, respectively. We observe that classification based methods are bounding box classifiers and utilize significantly more information than matching based methods as ours. This explains why our detection rate and AUC is lower than [22,9].

The proposed approach can not only succeed in extensive cluttered images, but also handles the problem of large range of scales and intra-class variability. This is demonstrated by several examples in Fig. 4. The images in the bottom right of Fig. 4(a) with red rectangles are the ones we fail to detect. The images of horses in Fig. 4(b) with red rectangles are false positives in the negative images provided by Shotton et. al. [22] to complement the Weizmann horse dataset.

Table 1. Detection rate

	Our method	Zhu et al. [26]	Shotton et al. [22]
Horses	93.97%	86.0%	95.20%
Cows	90.38%	88.6%	N/A

They show that the false positives in the negative set are caused by really very cluttered edges or by the structure of edges happening to match to the model very well. Interestingly, the rightmost false positive of horses is due to a camel, whose shape is very similar to that of a horse.

6.2 Localizing Object Boundaries

The method presented in this paper offers one important advantage compared to texture based and classification methods like [3,9,4]. It can localize object boundaries, rather than just bounding-boxes.

In order to quantify how accurately the output shapes match to true boundaries, we use the coverage and precision measures defined in [7]. Coverage is the percentage of points from ground-truth boundaries closer than a threshold t to the output shapes of the proposed approach. Reversely, precision is the percentage of points from output shapes closer than t to any point of ground-truth boundaries. As in [7] t is set to 4% of the diagonal of the ground-truth bounding box. The measures are complementary. Coverage captures how much of the object boundary has been recovered by the algorithm, whereas precision reports how much of the algorithm's output lies on the object boundaries. These measurements are really useful and suitable for evaluating shape based approaches. In comparison, bounding-box evaluation cannot represent how accurate the detected shapes match the ground-truth boundary. It is possible to have bounding-box intersection larger than 0.5 without having correctly identified the ground-truth object boundaries. Two examples of horse detection are shown in Fig. 4(b) with green rectangles.

The first two columns of Table 2 show coverage and precision averaged over all images of the class giraffes in ETHZ dataset in comparison to the results in [7]. We measure the coverage and precision for the correct detections at 0.4 FPPI, following [7]. The coverage of the proposed approach is over 11% better than [7], which shows that our approach can efficiently recover the true boundary of objects. The precision is a little lower than [7]. More importantly, the detection rate at our 0.4 FPPI is 86.75%. However, even for 20% bounding box intersection, the detection rate at 0.4 FPPI in [7] is only around 60% , which is much less than us. It demonstrates that our approach can correctly localize object's boundary on more images.

For horses and cows, the coverage and precision are obtained over all correct detections. The third column of Table. 2 shows the coverage and precision of the proposed method on the Weizmann Horse dataset. As the edges are significantly worse than the ones provided for the giraffes, both measures are worse than the results on giraffes. The coverage and precision results for cow are shown in the fourth column of Table. 2. Due to less intra-shape variance, the precision is 92.02%, which is much higher than giraffes and horses. However, the coverage is only 73.86%. The main reason for the difference between these two values is that our model has a gap, since we removed the contour part representing the horse tail from the horse contour used for part decomposition. Thus, even if the model and object match perfectly, the coverage score cannot be perfect (see examples in Fig. 4).

Table 2. Accuracy of the boundary localization

	Ours on giraffes	Results in [7] on giraffes	Ours on horses	Ours on cows
Coverage	79.4%	68.5%	77.5%	73.86%
Precision	74.6%	77.3%	61.7%	92.02%

7 Conclusion and Discussion

This paper mainly contains two contributions: shape model learning through shape matching and a novel framework for shape based object detection. The proposed model learning method can not only learn the model for non-rigid or articulated objects with partially-supervised learning, but also transfer the structure information to different kinds of objects. More importantly, the spatial layout between parts is also modeled.

We extend the classical particle filter framework in order to be able to infer an optimal label assignment to RVs whose dependencies are described by a complete graph. The values of RVs represent contour parts of our shape model and their locations. In our framework each particle explores a different order of detected contour parts, and the most informative order is selected by particle resampling.

Acknowledgments

The work has been supported by the NSF Grants IIS-0812118, BCS-0924164, the AFOSR Grant FA9550-09-1-0207, and the DOE Award 71498-001-09.

References

1. Bai, X., Wang, X., Latecki, L.J., Tu, Z.: Active skeleton for non-rigid object detection. In: ICCV (2009)
2. Borenstein, E., Sharon, E., Ullman, S.: Combining top-down and bottom-up segmentation. In: POVC (2004)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
4. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV (2009)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61(1) (2005)
6. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: From images to shape models for object detection. IEEE Trans. PAMI 30(1), 36–51 (2008)
7. Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. IJCV 87(3) (2010)
8. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315(5814), 972–976 (2007)
9. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: CVPR (2009)

10. Gordon, N., Salmond, D., Smith, A.: Novel approach to nonlinear/non-gaussian bayesian state estimation. IEE Proceedings of Radar and Signal Processing 140, 107–113 (1993)
11. Gu, C., Lim, J.J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR (2009)
12. Ioffe, S., Forsyth, D.: Finding people by sampling. In: ICCV (1999)
13. Ioffe, S., Forsyth, D.: Probablistic methods for finding people. IJCV (2001)
14. Kokkinos, I., Yuille, A.: Hop: Hierarchical object parsing. In: CVPR (2009)
15. Lee, Y.J., Grauman, K.: Shape discovery from unlabeled image collections. In: CVPR (2009)
16. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Proceedings of the Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic (May 2004)
17. Ling, H., Jacobs, D.: Shape classification using the inner-distance. IEEE Trans. PAMI 29, 286–299 (2007)
18. Lu, C., Latecki, L.J., Adluru, N., Yang, X., Ling, H.: Shape guided contour grouping with particle filters. In: ICCV (2009)
19. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR (2009)
20. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: ICCV (2009)
21. Ravishankar, S., Jain, A., Mittal, A.: Multi-stage contour based detection of deformable objects. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 483–496. Springer, Heidelberg (2008)
22. Shotton, J., Blake, A., Cipolla, R.: Multi-scale categorical object recognition using contour fragments. IEEE Trans. on PAMI 30(7), 1270–1281 (2008)
23. Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. In: ICCV (2009)
24. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
25. Yuille, A., Coughlan, J., Wu, Y., Zhu, S.: Order parameters for detecting target curves in images, when does high-level knowledge help? IJCV 41(1/2), 9–33 (2001)
26. Zhu, L., Chen, Y., Yuille, A.: Learning a hierarchical deformable template for rapid deformable object parsing. IEEE Trans. PAMI 99(1) (2009)
27. Zhu, Q., Wang, L., Wu, Y., Shi, J.: Contour context selection for object detection: a set-to-set contour matching approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 774–787. Springer, Heidelberg (2008)