Active Skeleton for Non-rigid Object Detection

Xiang Bai *
Huazhong Univ. of Sci.&Tech.
xbai@hust.edu.cn

Xinggang Wang
Huazhong Univ. of Sci.&Tech.
wxghust@gmail.com

Longin Jan Latecki
Temple University
latecki@temple.edu

Wenyu Liu
Huazhong Univ. of Sci. & Tech.
liuwy@hust.edu.cn

Zhuowen Tu
University of California, Los Angeles
ztu@loni.ucla.edu

Abstract

We present a shape-based algorithm for detecting and recognizing non-rigid objects from natural images. The existing literature in this domain often cannot model the objects very well. In this paper, we use the skeleton (medial axis) information to capture the main structure of an object, which has the particular advantage in modeling articulation and non-rigid deformation. Given a set of training samples, a tree-union structure is learned on the extracted skeletons to model the variation in configuration. Each branch on the skeleton is associated with a few part-based templates, modeling the object boundary information. We then apply sum-and-max algorithm to perform rapid object detection by matching the skeleton-based active template to the edge map extracted from a test image. The algorithm reports the detection result by a composition of the local maximum responses. Compared with the alternatives on this topic, our algorithm requires less training samples. It is simple, yet efficient and effective. We show encouraging results on two widely used benchmark image sets: the Weizmann horse dataset [7] and the ETHZ dataset [16].

1. Introduction

Non-rigid object detection is a challenging problem in computer vision, due to the large deformation and intra-class variation of an object class. Recent approaches in the literature can be roughly divided into: (1) image appearance-based [9, 31]; (2) shape-driven [5, 16, 23, 4]; (3) and mixture of shape and appearance models [22, 32, 33]. Our work is focused on shape-based representation, but it performs object detection in real images. Moreover, appearance-based features can also be added to our representation.

*Part of this work was done while the author was at University of California, Los Angeles

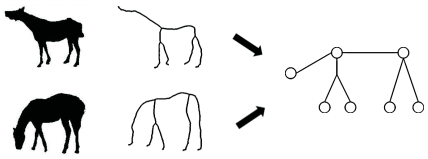

Figure 1. Illustration of two horses in different poses but with the same skeleton structure.

Objects under non-rigid deformation/articulation often observe large variation globally. However, their local structures are somewhat more invariant to the changes. A successful algorithm should be able to take advantage of the local invariance, account for the deformation, and perform effective and efficient detection. A recent benchmark study [11] shows that we are still far from being able to produce a practically useful system for detecting articulated objects, such as pedestrians. Without a mechanism to explicitly model the configuration and deformation (mostly through generative models) of an object class, classic discriminative models [9, 30] on a set of features are facing major bottlenecks due to limitation in modeling objects of complex configuration and high variation.

Blum [6] defined skeleton (medial axis) as a set of medial loci of the maximal disks inside object boundary. Skeleton captures certain degree of intrinsic shape information about objects and has been successfully applied in shape matching and classification on silhouettes [2, 36, 24]. Skeleton-based shape matching algorithms are robust against articulation since the skeleton topology is more-or-less stable (see the example in Fig. (1)). However, medial axis representation has been mostly used on already extracted shapes or binary images, and its advantage has not been successfully translated into detecting objects in cluttered natural images. The main obstacle is probably due to the difficulty of reliably extracting object skeletons from cluttered images.

In this paper, we introduce a new approach, *active skele-*

ton, for detecting non-rigid objects. We utilize the skeleton representation to capture non-rigid deformation of an object; the configuration difference of various skeleton instances is managed by a tree-union structure, similar to [28, 27]; each branch on the skeleton is associated with a few part-based templates, modeling the contour information. In training, we use a few examples for learning the skeleton tree-union structure whose branches are associated with contour parts. The contour part level is exemplar-based and the parts are connected and guided by the skeleton structure, allowing non-rigid deformation. In the detection stage, we first compute an edge map for an input image. Similarities between object parts in the model and the edge map are computed using Oriented Chamfer matching [23]. Inferring the overall object is then done by combining the local responses to match with the model guided by skeleton structure.

2. Related work

The idea of using skeleton (medial axis) to model the object shape has been extensively studied in the literature [6, 2, 36, 24, 19, 21, 18]. Most of the existing works, however, are focused on matching different shapes or silhouettes, which are already extracted from the images. Since extracting the skeletons for objects in cluttered background is a very difficult task, it is not clear how these methods could generalize to perform object detection in natural images.

Other shape matching algorithms such as Shape Contexts [5], Chamfer matching [26], Inner Distance [17], and Data-driven EM [29], also have not been fully illustrated on real images.

Early works for object detection through template matching can be dated to early 90s, e.g., the deformable template by Yuille et al. [32]. Shape-based approaches have the advantage of being relatively robust against lighting and appearance change. Recently, there has been a resurgence of matching-based object detection algorithms [13, 33, 34]. These systems decompose a given contour of a shape into a group of contour parts, and match the resulting contour parts to edge segments in a given edge image. A hierarchical structure is often used to combine these parts and enforce their compatibility. However, without a good mechanism to explicitly account for the articulation of an object, the modeling capability of the existing methods is rather limited. For example, only two legs were assumed in [33] and many horses bending the head were not successfully matched. Our skeleton-based structure improves the capability of modeling non-rigid objects over these algorithms.

In Ferrari et al. [14], a shape codebook of contours is learned, followed by edge linking methods named KAS or TAS to obtain many salient segments before detection. Shotton et al. [23] describes the shape of the entire object using deformable contour fragments and their relative positions.

The tree-union structure is shown to be effective in modeling the configuration variation [28, 27]. In this paper, we use it to represent the skeleton configuration, which is learned through a weakly supervised manner. We associate object parts, learned from training samples, to the branches in the skeleton. This is different from, e.g. [10], where ellipses are assumed. The Active Basis algorithm [31] is primarily focused on learning effective appearance models for the object parts by slightly perturbing their locations and orientations.

In the detection stage, we avoid the difficult steps of performing segmentation, extracting skeletons and then matching the candidates against the learned template. Instead, we match the parts with the edge maps and use the sum-and-max algorithm [20, 31] to perform rapid detection. This makes the algorithm simpler and more computationally efficient than other methods using dynamic programming [33], belief propagation [13], or MCMC [35]. We illustrate our algorithm on two widely adopted benchmark datasets: the Weizmann horse dataset [7] and the ETHZ dataset [16], and show very encouraging results with only a few training samples.

3. Active skeleton model

In this section, we give the active skeleton formulation, which uses skeleton as the back-bone, giving explicit object models. In this regard, our model is generative.

3.1. Skeleton-to-Contour Template

Skeleton, also named medial axis, is a set of points centered at the maximal disks of the object boundaries [6]. In this paper, we call a skeleton point having only one adjacent point an **endpoint** (the skeleton endpoint); a skeleton point having more than two adjacent points a **junction point**; and a skeleton segment between two skeleton points a **skeleton branch**. Given a 2D contour, one can compute its skeleton; likewise, we can obtain the contour from the skeleton with the information about its corresponding maximal disks. Therefore, contour and skeleton observe the duality of a 2D shape, and we can always derive one from the other. However, it is more effective to use the skeleton to represent the deformation; it is more direct to use the contour (boundary) for image measurement since skeleton cannot be directly observed. Therefore, our model has two main variables, (ST, C), where ST is a skeleton instance and C contains contour fragments based on ST.

Given a training sample of segmented object, we can automatically extract its skeleton, ST, using a technique in [3]. In this paper, we represent it in a tree structure. Two examples are shown in Fig. (1). Each ST has its corre-

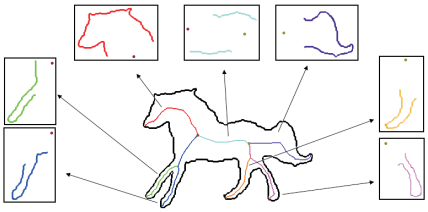

Figure 2. A skeleton structure with each of its branch associated with a contour template. The corresponding contour templates and skeleton branches are shown in the same color.

sponding end points, junction points, and skeleton branches. An illustration is given in Fig. (2), where different skeleton branches are shown in different colors, representing the visual parts of a horse. Therefore, we can denote a skeleton as

$$ST = (B_i; i = 1..N)$$

where B_i is a skeleton branch and N is the total number of branches. The statistical variation of an object class is represented by a tree-union structure, which will be discussed in the next section. Contour C is then dependent on skeleton ST with each skeleton branch, B_i, associated with a contour fragment (segment), F_i.

$$C = (F_i; i = 1..N).$$

Fig. (2) shows some examples of such contour fragments. It is effective and computationally efficient to perform matching-based object detection using our representation, in which contour fragments are attached with a skeleton: (1) skeleton branches represent meaningful parts of an object and they observe consistency across different instances; (2) the local symmetry of the contour fragments can be detected and matched. This is demonstrated by a few recent works [25, 1] on contour grouping based on symmetric segments. Notice that in Fig. (2) the positions of the junction points adjacent to the corresponding skeleton branches are also coded in the model, since these junction points will be used for combining all the templates in the matching phase.

Having given the notation of skeleton and contour fragment, we are ready to discuss our image model. An object we want to detect is (ST, C, Θ), where Θ contains the detailed parameters such as $rotation$ and $scaling$. Given an input image I, we model it in a Bayesian framework by

$$\begin{aligned} p(ST, C, \Theta|I) &\propto p(I|ST, C, \Theta)p(C|ST)p(ST) \\ &\propto p(I|C, \Theta)p(C|ST)p(ST), \end{aligned} \quad (1)$$

where $p(I|ST, C, \Theta) = p(I|C, \Theta)$ is the likelihood appearance model, $p(ST)$ is a prior on the skeleton tree, and $p(C|ST)$ models the variation of different contour fragments on ST. We assuem equal prior on the Θ for simplicity. Two key observations we make are: (1) appearance model $p(I|C, \Theta)$ does not depend on ST; and (2) contour fragments C are dependent on the skeleton ST. In this paper, we directly work on the Canny edge map extracted from image I, and, therefore, the likelihood model is computed on edges. However, our algorithm does not prevent the direct use of appearance models (often requires a learning procedure such as the active basis work [31]). Thus,

$$p(I|C, \Theta) \propto \prod_i \exp D(Eg(I), F_i(\Theta)),$$

where $Eg(I)$ denotes the edge map of image I, and $D(Eg(I), F_i(\Theta))$ measures the similarity between transformed F_i and a portion of $Eg(I)$ at a specific location and at a certain scale. We use Chamfer distance measure in this paper.

We use a tree-union structure to learn the prior $p(ST)$ and the details are given in the next section. Once ST is modeled, then

$$p(C|ST) = \prod_i p(F_i|B_i),$$

where we directly represent $p(F_i|B_i) = \sum_j \alpha_j \cdot p(D(F_i, F_j(B_i))$ using an exemplar-based mixture model, since we only have a few training samples. $F_j(B_i)$ are the contour fragments in the training samples, and $D(F_i, F_j(B_i))$ measure the Chamfer distance between F_i and exemplar $F_j(B_i)$. Since the other measures in eqn. (1) are standard, we focus on $p(ST)$, the prior on the skeleton configuration, in the next section.

Here we use oriented Chamfer matching (OCM) proposed in [23] as the distance measure for $D(Eg(I), F_i(\Theta))$. A simple linear interpolation between the distance and orientation terms is used for OCM. For each point on a template, the OCM distance is: $d_\lambda = (1 - \lambda) \cdot d_{cham,\tau} + \lambda \cdot d_{orient}$. For the experiments in this paper, $\lambda = 0.3$ and $\tau = 20$. The details about OCM can be found in [23].

3.2. Modeling skeleton instances using tree-union

The computer vision field has recently witnessed a resurgence of using grammar to represent objects of high variation and complexity [35, 33], which is a very promising direction. In this paper, we focus on the practicality of representing and learning the skeleton structure and performing efficient detection. Thus, we adopt a simpler scheme, tree-union, for the representation. More sophisticated models may be needed with more complex structures and variations.

The junction points/endpoints can be looked at as the **critical points** for the topology of a skeleton, since we can build a graph/tree directly by using these critical points as the nodes and the skeleton branches as the edges between them. As shown in Fig. (3.a), we select a junction point A on the skeleton of a horse as the root node, then

a skeleton tree can be built easily. Given a skeleton tree $ST = ST(V, E)$ rooted in A, we denote V as the set of nodes and E as the set of edges.

Now we define a skeleton tree-union. Its actual construction is described in Section 3.3. We are given a group of skeleton trees $\{ST_1, ST_2, ...ST_k\}$ that belong to the same shape class. The tree-union T_U can be looked as their common prototype or an abstraction of them. A tree-union $ST_U = ST_U(V_U, E_U)$ is defined as the largest common tree of $\{ST_1, ST_2, ..., ST_k\}$ such that there exists mappings $F_i : ST_i \rightarrow ST_U$ $(i = 1, \ldots, k)$ that preserve the tree topology. Consequently, for any edge $e_u \in E_U$, $F_i^{-1}(e_u)$ can be viewed as a union of all branches that map to e_u. For example, in Fig. (3.a,b), all three sample skeletons in (a) have their skeleton branches (E_i) and critical points (V_i) corresponding to the tree-union in (b), where the corresponding components are in the same colors.

There have been some mature approaches [28, 10] for unsupervised learning of the class structure of the samples. However, in our experiments, we build the tree-unions in a simpler way introduced in Section 3.3. As the correspondences between critical points and branches of different skeletons can be obtained by the existing skeleton matching methods, we assume that the correspondences between the nodes of their skeleton trees $\{ST_1, ST_2, ..., ST_k\}$ are known. Since skeleton matching approaches allow us to match empty nodes, we allow adding empty branches and endpoints during the process of construction of tree-union. However, we do not allow matching junction points to empty nodes in order to preserve topological tree structure.

As each skeleton branch has been mapped to an edge of the tree-union, each contour template (reconstructed from the skeleton branch) will also be mapped into the tree-union. Imagine that we randomly take only one template for each edge of the union-tree from the group of templates mapped to that one, we can generate many different horses shown in Fig. (3.c)) according to the structure of the tree-union, which are all generated by the contours from the only three horses in Fig. (3.a). Thus, using a tree-union to combine the contour templates from each "part" according to human perception and it is natural to represent the shape deformations, which can be considered a large active template with several moving or deformable parts.

3.3. Learning a tree-union

As pointed out by [28, 10] skeleton abstraction can be learned by utilizing skeleton matching. First skeletons are computed for binary (segmented) shape images. Then skeleton abstraction is learned for a given set of shapes in the same class. The learned tree-unions by Torsello and Hancock [28] are very complicated to accommodate for all possible skeleton variations in a given shape class. While

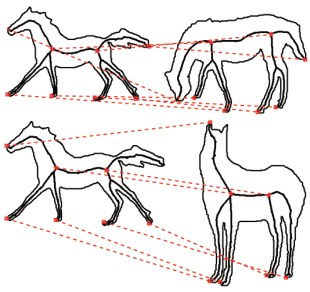

Figure 4. Skeleton matching using improved path similarity measure [2] for two horses in different poses.

this is shown to be beneficial when dealing with binary shapes, it is too general to guide object detection and recognition in edge images in that it will lead to large numbers of false positive, i.e., the object detection will hallucinate the target object in clutter.

The skeletal tree abstraction by Demirci et al. [10] appears to be more suitable for object detection in edge images. However, it still seems to over generalize in that the obtained tree abstractions are not specific enough to prevent hallucinating of target object in clutter. We stress again that tree abstraction [10] as well as [28] were designed for dealing with binary shapes, and we are focused on their usage as models for object search in edge images, which often are cluttered.

In the proposed approach we utilize the main idea of skeleton matching to construct shape class models but, at the same time, we ensure that our models are sufficiently specific by restricting the learned union tree to having the same topology. Thus, in our approach each tree union is represented by a single tree. This does not mean that all trees combined to form a given union-tree have the same topology. This means that we can abstract their topology by ignoring some junction points, and that after abstraction the topology of all trees combined to a single tree union is identical. This fact is illustrated in Fig. (4), where two skeleton junction points of the horse top right are ignored. At the same time the junction point in the rear legs of the horse in top left is ignored too.

To learn our tree-union, we start with one manually labeled skeleton, e.g., the first horse in Fig. (3). Then, we perform skeleton matching with skeleton paths [2]. We added a constraint that junction points on the paths must match junction points. This allows us to abstract junction points. If a matching score is below a given threshold, we extend our tree union by adding the new tree. This simple incremental learning approach is guaranteed to persevere topology due to the constraint that junction points on the paths must match junction points. In the resulting tree-union, each edge represents a set of contour parts including possibly an empty part (it represents like a missing leg). Some edges do not al-

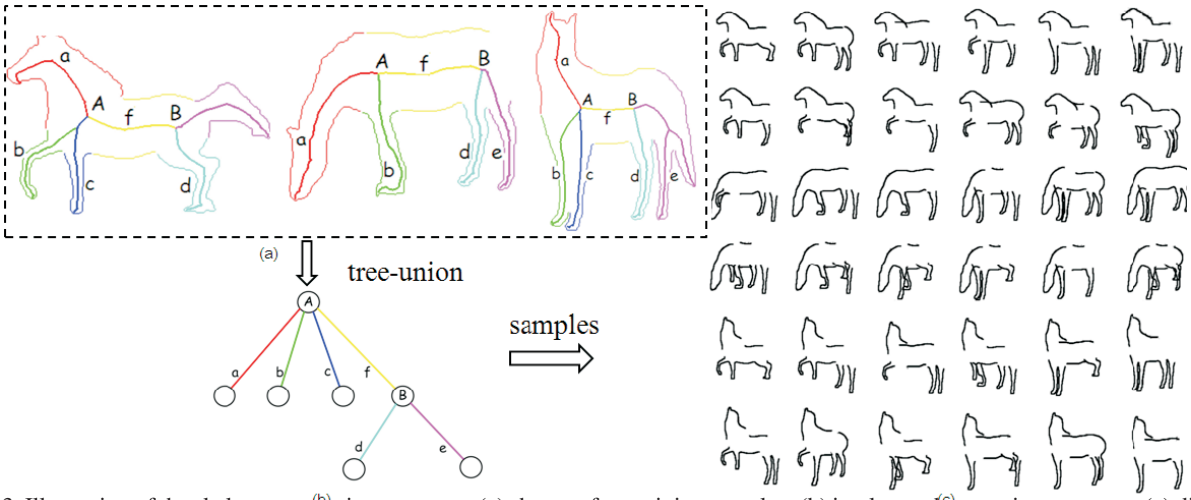

Figure 3. Illustration of the skeleton tree-union structure. (a) shows a few training samples. (b) is a learned tree-union structure. (c) displays some samples drawn from the learned model.

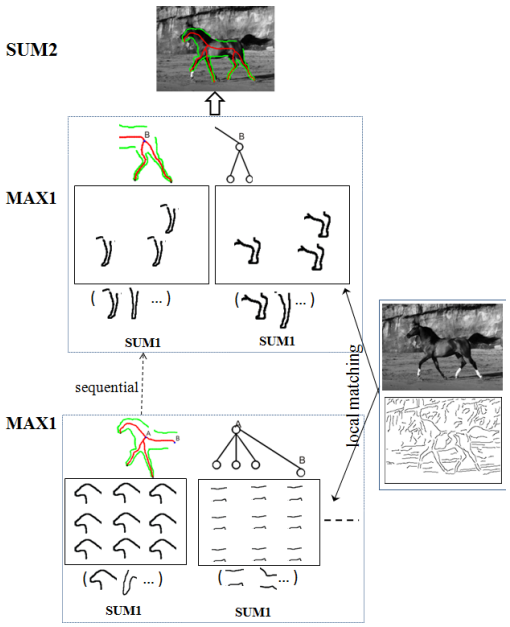

Figure 5. Illustration of the detection process using the sum-and-max algorithm. An edge map is first extracted from an input image. Starting from the root node of the tree-union, local contour templates are matched against the edge map (**SUM1**). Maximum responses of the candidate templates connecting to the current node are kept (**MAX1**). The process moves to the next node in the tree-union to match with other segments (**MAX1**). The overall detection is based on a composition of these local maximum responses (**SUM2**).

low an empty part, e.g., the edge for horse head, which is enforced by the requirement of good matching score.

4. Detection

The choice of the inference algorithm critically decides the quality of an object detector. Object detection using Markov chain Monte Carlo [35] is often slow, though it guarantees to find the global optimal asymptotically. Local pruning procedures are used in a bottom-up and top-down process [33, 34], resulting in quite complex algorithms. In this paper, we emphasize enhancing the representation power of the underlying model. If the model is powerful enough, its energy function may not have many local minima. Thus, a simpler algorithm may be sufficient to perform rapid object detection.

The sum-and-max algorithms are used in [20, 31]. They are easy to implement, effective, and fast to compute. In [20, 31], local evidences, e.g. Gabor function filtering results, are pooled together for some tests; the maximum responses are then kept among promising locations; these kept maximum responses are verified in composition. Depending upon the complexity of the model itself, one can build multiple levels of sum-and-max to pool evidences hierarchically. In this paper, we adopt the basic idea of the sum-and-max algorithm. However, our algorithm differs from [20, 31] due to the tree-union structure, which allows us to perform sequential tests. Given an image, our goal is to detect an object at a particular location and scale with a low energy defined in eqn. (1). The outline of our detection algorithm is the following.

1. For an input image I, we compute its edge map using Canny operator (one can use other favorite algorithms).

2. We start from the root node, which is now our current node. All the skeleton branches connected by the current node are matched independently. For example, several horse heads (exemplars learned in the training phase) are matched with the edge maps at different locations and scales. We use Oriented Chamfer matching (OCM) to compute the shape similarity, and this is

a "sum" operations since each contour fragment itself contains a sequence of points.

3. The exemplars with the maximal score are kept (this is a "max" operation).
4. All the matched fragments are gathered together to compute the local probability on the current node.
5. After the branches of the current node are all checked, we then move to the child node in the tree-union structure, which is our new current node.
6. Once all the nodes are explored, the matched contour segments decide the overall probability (eqn. (1)) in composition. This is another "sum" operation to piece the local evidences together.

Fig. (5) illustrates the detection procedure using our sum-and-max algorithm. In the experiment part, we implement a coarse-to-fine strategy.

1. A coarse-level detection is performed. We have sample locations on each edge map for detection as the root node for a tree union is not very sensitive. The templates for detection was set in 5 scales for computing a optimal coarse confidence for each location. (In our experiments, we set a detection point every 10 pixels on each image.)
2. Once the positive positions for the root node are computed, we apply a mean-shift algorithm [8] to allow it drift to a more promising place globally;
3. Based on the roughly detected root node, we perform fine-level detection, whose procedure is slightly different. We allow each template to rotate according to its relative junction point in different at its optimal scale, then we obtain a more robust position for each template. Fig. (6) give an example for fine-level detection. As shown in Fig. (6.a), the rotation of the templates will generate a new shape that is still very similar to the same class. Fig. (6.b) is the detected results at the coarse-level, and Fig. (6.c) is the final detected result based on Fig. (6.b) by rotation. The final average distance (using OCM) between the points on the detected templates and the edge points will be returned as the final confidence for the whole detection.

The detection algorithm finally keeps the solutions for an image at the places that $p(V|T, B, \Theta)$ pass a certain threshold.

5. Experiments

We provide experimental evaluation of the proposed approach on two datasets: the Weizmann horses [7] and the ETHZ shapes [16]. The Weizmann horse dataset has 328

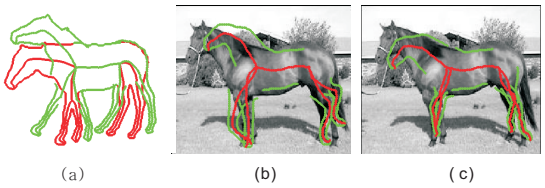

Figure 6. Illustration of matching contour segments at the fine level.

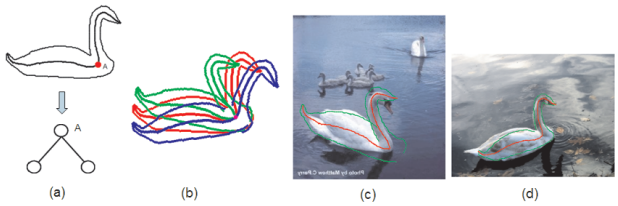

Figure 7. Detection results for two test images in the ETHZ dataset.

images viewed from the side, covering different poses, articulations, deformations, and scales. We compare our method directly to one using matching-based detection algorithm, Shotton et al. [23]. Same as in [23], 328 Caltech-101 background images [12] were used for evaluation as the negatives. In our work, only 15 horse images were used for training, and the remaining 313 were used for testing, whereas 50 training image were used in [23]. We applied Canny edge detector to obtain the edge maps.

As in [23], a successful detection is measured as the following: the inferred bounding box b_{in} (the smallest bounding rectangle containing the detected templates) must agree with the ground truth bounding box b_{gt}, based on an overlap criterion as $\frac{area(b_{in} \bigcap b_{gt})}{area(b_{in} \bigcup b_{gt})} > 0.5$. We consider recall against the average number of false positive per image (RFPPI), the saqme as in [23]. The area under curve (AUC) measure of RP is reported in Table (1) for a comparison. Our method approaches AUC of 80.32%, whereas it was 84.98% in [23]. However, their method used many more training samples than ours and were consequently tested on fewer test images. More importantly, our algorithm is generative and it explicitly explores the intrinsic variation of an object due to non-rigid deformation. Our method also appears to be simpler and it points to a promising direction on this topic.

The detection results of some examples are shown in Fig. (8) in which matched segments are overlayed on top of the original gray-scale images. The extracted Canny edge maps are also shown on the left of Fig. (8). All examples were taken from the Weizamann dataset except for one on the right of the fourth row. Both the detected contour templates (in green) and their corresponding skeleton branches (in red) are displayed. Even though the horses

	Training Size	Testing Size	Detection RP AUC	Time (per image)	Platform
The method in [23]	50	228	84.98%	10 seconds	C#
Our method	15	313	80.32%	2 seconds	Matlab & C

Table 1. Performance comparison for the Weizmann horse images. [7]

Methods	[15]	[34]	Ours
P/R	23.3%/93.9%	31.3%/93.9%	**61.22%/93.9%**

Table 2. Comparison of precision on the swan images of the ETHZ dataset. [16]

in these images exhibit large appearance change and shape deformation, the detection results are still quite stable. The fourth row demonstrates the effectiveness of our algorithm on very cluttered backgrounds. The last row illustrates the robustness of the proposed method against heavy occlusions, which are created by hand (white rectangles).

We also tested our system on the ETHZ dataset [16], which is widely used in the literature. It has 5 different object categories of 255 images in total. All categories have significant intra-class variation, scale change, and illumination difference. Moreover, many objects are surrounded by extensive background clutter and have interior contours. There are only two classes in the ETHZ dataset that exhibit non-rigid deformation: swan and giraffe. We tested our approach on the swan images, which are very dissimilar with horses. Only one training sample is used in this case, shown in Fig. (7.a). Point A is selected as the root node. Then a simple tree-union at the bottom of Fig. (7.a)) is built, which allows the templates to roate the neck and the body, as shown in Fig. (7.b). Fig. (7.c,d) show two results of detected swans.

Precision vs recall (P/R) curve is used for quantitative evaluation. There are 32 swan images in the ETHZ dataset, thus the other 223 images are used as negative images. We plot our PR curve in Fig. (9) with the comparison to two recent methods [15, 34] which use contour fragments only. We also compare the precision to [15] and [34] at the same recall rates in Table (2). The precision / recall pairs of [15] and [34] are quoted from [34]. The significant improvement on the results demonstrate that our model is more flexible than those using contour fragments only, which do not have the guidance from skeleton information. With skeleton capturing the main deformation field of an object, we end up with using a more compact representation, but with significantly improved results.

6. Conclusion

In this paper, we introduce a new representation, *active skeleton*, for object detection and recognition. Using skeleton has the particular modeling advantage of capturing articulation and non-rigid deformation. As the traditional classification-based approaches using generic features such as Haar and HOG features are reaching the limits, modeling the object variation with explicit representations (generative aspect) holds the special promises. Many of the existing generative-based object detection/recognition algorithms, unfortunately, are either very limited with the modeling power, too complicated to learn, or too computationally expensive to compute in practice. Most of the skeleton-based shape algorithms are not addressing the important problem of detecting objects in cluttered images. Our algorithm is shown to be effective and efficient, with a simple representation. Adding more learning components to account for large variations in configuration and the appearance change might further improve our algorithm.

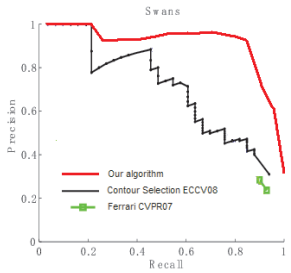

Figure 9. RR curve showing the performance for the swan images of the ETHZ dataset.

Acknowledgements

This project was supported by ONR N000140910099 and NSF CAREER award #0844566. We also thank the support from China 863 2008AA01Z126 and NSFC (No.60873127). Xiang Bai was supported in part by MSRA Fellowship. We would like to thank Jamie Shotton for providing his test images. We also want to thank Yingnian Wu for his useful comments and suggestions.

References

[1] N. Adluru, L. Latecki, R. Lakämper, T. Young, X. Bai, and A. Gross. Contour grouping based on local symetric. In *ICCV*, 2007. 3

[2] X. Bai and L. J. Latecki. Path similarity skeleton graph matching. *IEEE Trans. PAMI*, 30(7):1282–1292, 2008. 1, 2, 4

[3] X. Bai, L. J. Latecki, and W.-Y. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. PAMI*, 29(3):449–462, 2007. 2

[4] X. Bai, Q. Li, L. Latecki, W. Liu, and Z. Tu. Shape band: A deformable object detection approach. In *CVPR*, 2009. 1

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002. 1, 2

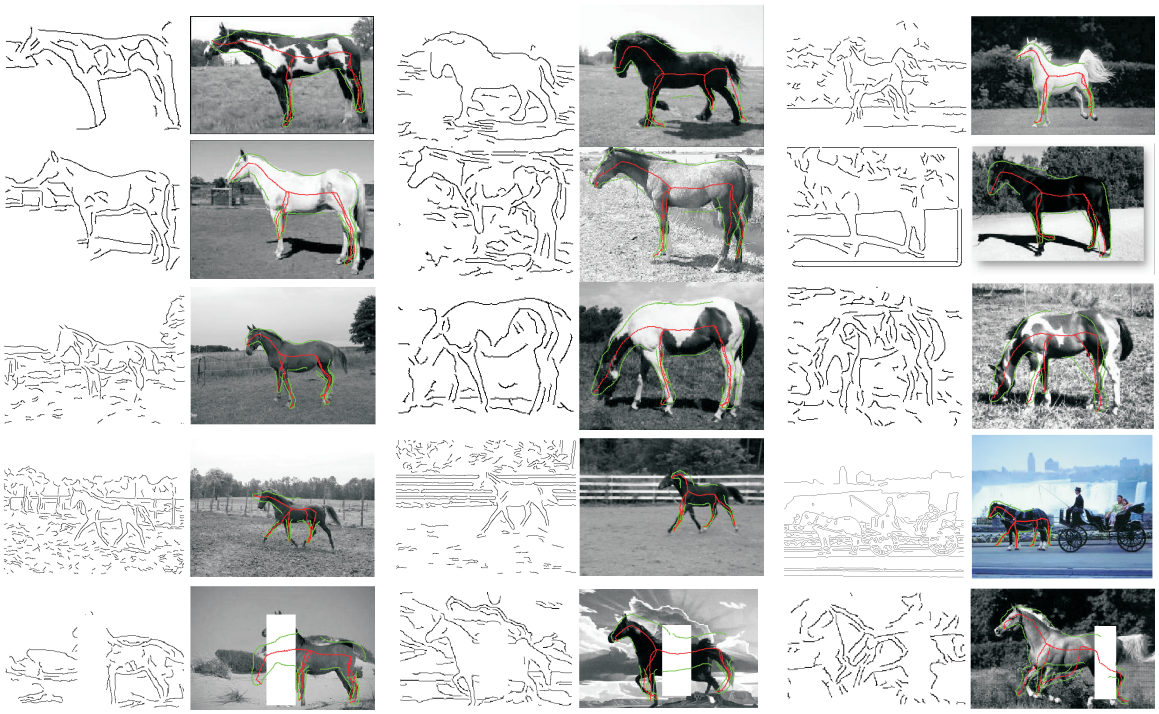

Figure 8. Detection and recognition results for some test images in the Weizmann dataset. The bottom row gives several examples where severe occlusions exist.

[6] H. Blum. Biological shape and visual science (part i). *J. Theoretical Biology*, 38:205–287, 1973. 1, 2

[7] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *POCV*, 2004. 1, 2, 6, 7

[8] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. PAMI*, 24(5):603–618, 2002. 6

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of CVPR*, pages 886–893, 2005. 1

[10] F. Demirci, A. Shokoufandeh, and S. Dickinson. Skeletal shape abstraction from examples. *IEEE Tran. on PAMI*, 2009. 2, 4

[11] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, 2009. 1

[12] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of edges and object boudaries. *IEEE Trans. PAMI*, 28(4):594–611, 2006. 6

[13] P. F. Felzenszwalb and J. D. Schwartz. Hierachical matching of deformable shapes. In *CVPR*, 2007. 2

[14] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *PAMI*, 30(1):36–51, 2008. 2

[15] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection combining recognition and segmentation. In *CVPR*, 2007. 7

[16] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. In *ECCV*, 2006. 1, 2, 6, 7

[17] H. Ling and D. Jacobs. Using the inner-distance for classification of articulated shapes. *PAMI*, 29(2):286–299, 2007. 2

[18] D. Macrini, K. Siddiqi, and S. Dickinson. From skeletons to bone graphs: Medial abstraction for object recognition. In *CVPR*, 2008. 2

[19] S. Pizer and et al. Deformable m-reps for 3d medical image segmentation. *IJCV*, 55(2):85–106, 2003. 2

[20] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuro-science*, (2):1019–1025, 1999. 2, 5

[21] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *PAMI*, 26(5):550–571, 2004. 2

[22] J. Shotton, A. Blake, and R. Cipolla. Efficiently combining contour and texture cues for object recognition. In *BMVC*, 2008. 1

[23] J. Shotton, A. Blake, and R. Cipolla. Multi-scale categorical object recognition using contour fragments. *PAMI*, 30(7):1270–1281, 2008. 1, 2, 3, 6, 7

[24] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *IJCV*, 35(1):13–32, 1999. 1, 2

[25] J. S. Stahl and S. Wang. Globally optimal grouping for symmetric boundaries. In *CVPR*, 2006. 3

[26] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Shape contexts and chamfer matching in cluttered scences. In *CVPR*, 2003. 2

[27] S. Todorovic and N. Ahuja. Unsupervised category modeling, recognition, segmentation in images. *IEEE Trans. PAMI*, 30(12):2158–2174, 2008. 2

[28] A. Torsello and E. Hancock. Learning shape-classes using a mixture of tree-unions. *IEEE Trans. PAMI*, 28(6):954–966, 2006. 2, 4

[29] Z. Tu and A. Yuille. Shape matching and recognition using generative models and informative features. In *ECCV*, 2004. 2

[30] P. A. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 1

[31] Y. Wu, Z. Si, H. Gong, and S. Zhu. Active basis for deformable object modeling, learning and detection. *IJCV*, 2009. 1, 2, 3, 5

[32] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *IJCV*, 8(2):99–111, 1992. 1, 2

[33] L. Zhu, Y. Chen, X. Ye, and A. L. Yuille. Structure-perceptron learning of a hierarchical log-linear model. In *CVPR*, 2008. 1, 2, 3, 5

[34] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV*, 2008. 2, 5, 7

[35] S. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Comp. Graphics and Vis.*, 2(4):259–362, 2006. 2, 3, 5

[36] S. C. Zhu and A. L. Yullie. Forms: A flexible object recognition and modeling system. *IJCV*, 20(3):187–212, 1996. 1, 2