Contour Based Object Detection Using Part Bundles

ChengEn Lu[*,a,c,d], Nagesh Adluru[b], Haibin Ling[a],
Guangxi Zhu[c,d], Longin Jan Latecki[a]

[a]*Dept. of Computer and Information Science, Temple University, 19122, USA*
[b]*Dept. of Biostatistics, University of Winsconsin-Madison, 53705, USA*
[c]*Dept. of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China*
[d]*Div Commun and Intelligent Networks, Wuhan National Laboratory for Optoelectronics, Wuhan, 430074, China*

Abstract

In this paper we propose a novel framework for contour based object detection from cluttered environments. Given a contour model for a class of objects, it is first decomposed into fragments hierarchically. Then, we group these fragments into *part bundles*, where a part bundle can contain overlapping fragments. Given a new image with set of edge fragments we develop an efficient voting method using local shape similarity between part bundles and edge fragments that generates high quality candidate part configurations. We then use global shape similarity between the part configurations and the model contour to find optimal configuration. Furthermore, we show that appearance information can be used for improving detection for objects with distinctive texture when model contour does not sufficiently capture deformation of the objects.

Key words: part bundle, shape context, object detection

1. Background

The key role of contours and their shapes in object extraction and recognition in images is well established in computer vision and in visual perception. Extracting edges in digital images is relatively well-understood and there are robust

[*]Corresponding author.
Email addresses: luchengen@gmail.com (ChengEn Lu), adluru@wisc.edu (Nagesh Adluru), hbling@ist.temple.edu (Haibin Ling), gxzhu@hust.edu.cn (Guangxi Zhu), latecki@temple.edu (Longin Jan Latecki)

detectors like [21, 9]. However, it is often difficult to distinguish edge pixels belonging to meaningful object contours. The main challenge is due to the fact that most edge pixels represent background and irrelevant texture, while only a small subset of edge pixels corresponds to object contours. Further, the edge pixels do not simply form occluding contours but broken contour fragments due to noise and occlusion.

Grouping edge pixels into contours using various saliency measures and cues has been studied for long and is still an active research field [37, 30, 28, 15, 27]. Once contours are identified they can be further grouped into objects by performing shape matching with model contours. For example, Shotton *et al.*[26] and Opelt *et al.* [24] use chamfer distance [2] to match fragments of contours learnt from training images to edge images. McNeil and Vijayakumar [23] represent parts learnt from semi-supervised training as point-sets and establish probabilistic point correspondences for the points in edge images. Ferrari *et al.* [6] use a network of nearly straight contour fragments and a sliding window search. Thayananthan *et al.* [29] modify shape context [1] to incorporate edge orientations and Viterbi optimization for better matching in clutter. Flezenswalb and Schwartz [5] present a shape-tree based elastic matching among two shapes and extend it to match a model and cluttered images by identifying contour parts (smooth curves) using a min-cover approach [4]. More recently, Zhu *et al.* [38] formulate the shape matching of contours (identified using [37]) in clutter as a set-set matching problem. They present an approximate solution to the hard combinatorial problem by using a voting scheme ([33, 19]) and a relaxed context selection scheme using linear programming. They use shape context [1] as shape descriptor. Bai *et al.* [40] provide a skeleton-based model decomposition scheme to solve non-rigid object detection problem. Aside from the recent work mentioned above, there are many early studies that use geometric constraints for model-based object and shape matching [13, 11, 12].

In this paper we focus on object detection by grouping the edge fragments in an image according to a contour model. Like many previous studies [33, 19, 26, 38, 7], we follow a two-phase framework: (1) generating object hypotheses, and (2) picking the best one. Our main contributions are summarized below: For the first phase, we propose a novel *part bundle* model that allows us to use local shape similarity of contour parts to efficiently generate accurate hypotheses. Specifically, we decompose a given contour model into several part bundles, where each part bundle contains several overlapping contour fragments (see Fig. 1 for example decompositions). The model naturally handles hierarchical relations between fragments, as well as the AND/OR relations for co-existing and competing frag-

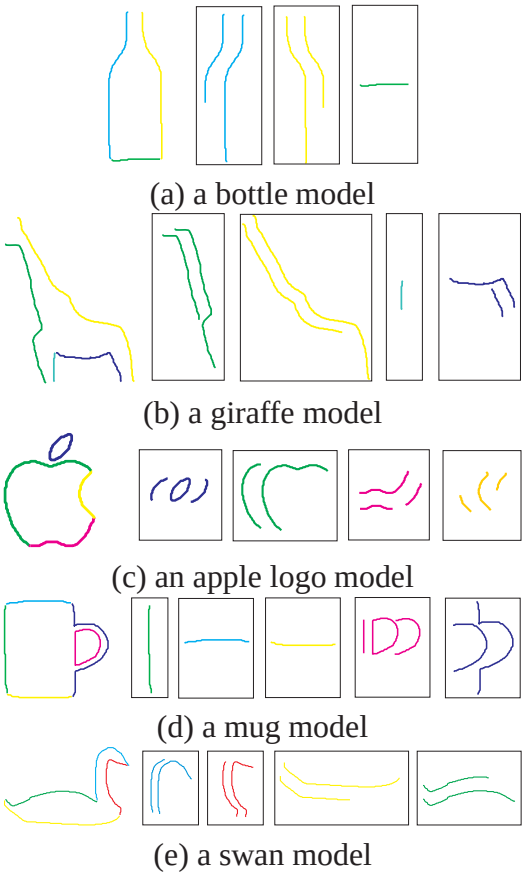


Figure 1: The part bundles of the hand-drawn contour models. Different parts and the part bundles are shown in different colors.

ment pairs. By matching the contours composed of fragments in the part bundles to that of the edge fragments in a given image we can generate concise and accurate hypotheses for part-configurations. The part bundle model is closely related to the hierarchical Random Filed model [18] and the AND/OR graphs [14, 39, 3]. While our model avoids explicit graph-theoretic parsing it keeps all the necessary properties for inferring part-configurations.

In the second phase, we compute global shape similarity of the part-configurations to the model contour using shape context [1] to pick the best part-configuration. Global shape matching is an expensive operation but we can afford to do this thanks to the concise and accurate hypotheses generated in the first phase using local shape matching of edge fragments to the part bundles. The illustration of

our framework is shown in Fig. 2.

Another contribution we made for the second phase is to investigate if appearance can compensate for the lack of model deformability for objects with distinctive texture. Using the bag-of-features [36, 35, 32, 10] model, we show that the detection performance can be significantly improved for objects with distinctive texture.

The rest of the paper describes the details of our framework followed by experimental evaluation. In §2 we describe the first phase of hypotheses generation. §3 describes how we select the best hypothesis in the second phase. In §4 we present experimental evaluation of our method, followed by conclusions and discussions in §5.

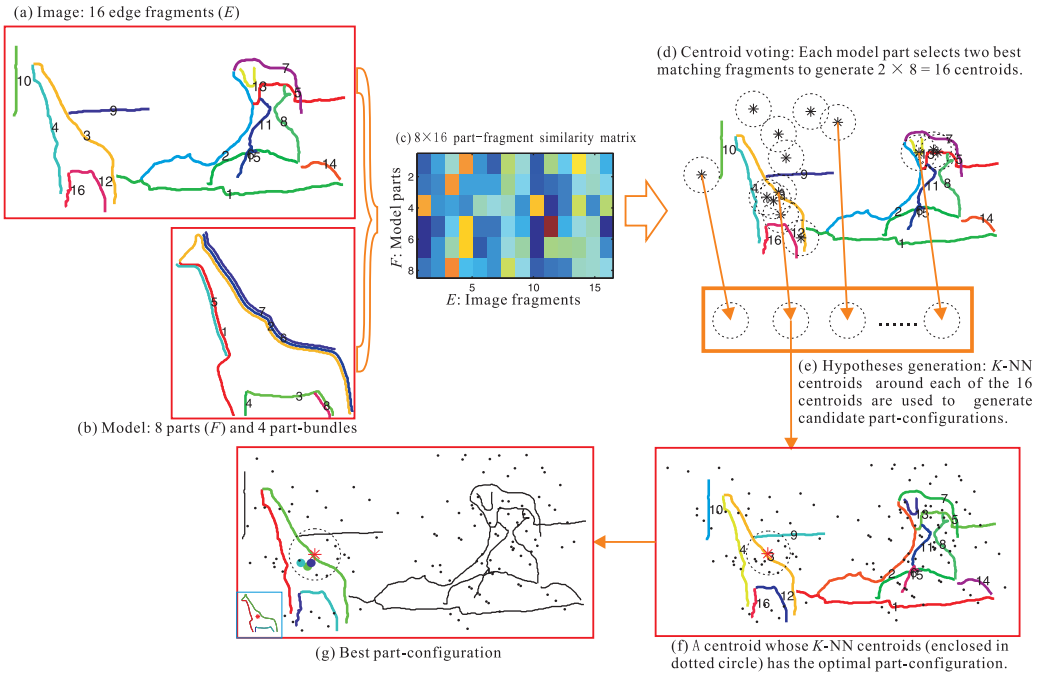


Figure 2: Illustration of our framework: (a) The edge fragments are generated using edge linking software [16] (see §2.2). (b) Contour based model is decomposed into part bundles (see § 2.1). (c) Contour fragments similarity, the color varies from blue to red indicates similarity from weak to strong. (d) Using contour fragment similarity we perform efficient centroid voting (see § 2.3). (e)-(f) The part configurations generated using K-NN centroids around each centroid are evaluated using shape and appearance information and the best part-configuration is picked (see § 2.3 and §3), the black dots are the centroids estimated by mapping a model fragment to a image fragment.

2. Hypotheses Generation Using Part Bundles

We aim to find the objects in the image using a shape matching scheme. The shape matching algorithm by Ferrari [8], called kAS, groups k pieces of adjacent segments (AS) together and treats the adjacent segments as an individual unit and performs shape matching between them. Zhu et al. [38] perform many-to-many matching between sets of model and image segments in their process of contour selection.

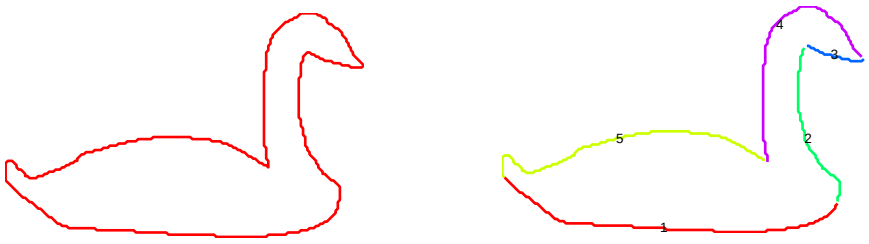
We propose a simple scheme that only allows one-to-one mapping between a model segment and an image segment. Our part bundle based decomposition of the model contour allows us to perform the one-to-one matching. We first explain how a given contour model is hierarchically decomposed into part bundles and then explain how the edge fragments are obtained from a given image. Using the part bundle model and edge fragments we generate good candidate part configurations of edge fragments in the image.

2.1. Part bundles model for shape decomposition

To model an object class, we first get its contour representation by tracing a sample image as in [8, 38]. As stated in the above subsection, we try to apply the same criteria to the shape model as that in the image segments to get fairly straight contour fragments. Since the model contour is a complete contour, we only apply *rule 2* (described below) to break the model contour to several contours at high curvature contour points. Other methods like discrete curve evolution (DCE) [17] also represent a shape contour as linear segments connected by several salient contour points. Salient points are the contour points with high curvature. Fig. 3 shows an example which use our *rule 2* to decompose a shape model to several contour pieces. Thus, both the image fragments and model segments are processed with the same criteria, which increases the probability of their one-to-one matching.

Moreover, we obtain a more flexible model structure by manual interference based on above decomposition. We manually decompose contour segments at some salient points and the center of a contour segment to form a hierarchical structure. Let $F = \{f_i\}_{i=1}^{m}$ denote the set of such fragments. A part bundle B is a subset of F, such that at most one fragment in B is allowed to have correspondence to an edge fragment in E. In other words, all fragments in a bundle B compete with each other during shape-to-image matching. With this bundle decomposition, we now can write our contour model as

$$\mathcal{B} = \{B_k\}_{k=1}^{m'} . \tag{1}$$



(a) (b)

Figure 3: A shape model is decomposed to several pieces at salient contour points using *rule 2* of our edge linking criteria.

where m' is the number of part bundles in model $\mathcal{B}$, which is usually very small. For bundles we have the following constraints

$$\begin{aligned} B_k \subset F\ , \quad & k = 1, ..., m' \\ B_i \bigcap B_j = \emptyset\ , \quad & 1 \leq i < j \leq m'\ . \end{aligned} \tag{2}$$

Sample bundle decompositions are shown in Fig. 1, we fix them as the models for ETHZ dataset.[1] It can be seen that a bundle can have fragments representing overlapping parts thus having redundancy and hierarchy. A cognitive motivation behind such decomposition scheme is that an object can be recognized even if some parts of it are missing, as can be observed in Fig. 4. There are several reasons why parts of objects can be missing in real images: missing edge information, occlusion, failures in contour grouping. The main constraint for the bundle design is to ensure shape flexibility, i.e., to accommodate for possible deformations and broken edge fragments in the edge image. Our experimental results presented in §4 show the superiority of our part bundle scheme over the control point scheme.

2.2. Extracting edge fragments from an image

Given is an input image I and its edge map extracted by the edge detection algorithm [22]. We extract a set of edge fragments, $E = \{e_i\}_{i=1}^{n}$, from the edge

[1] We design the part bundles based on the hand drawn models provided in the ETHZ dataset, but they are not optimal. For example, we observe that in the giraffe category, the necks of many giraffes have different bending angles. If we add another giraffe model to this category with a different neck, the performance of our method on that object category improves significantly.

Figure 4: Parts of the objects are missed both due to missing edge information and due to broken edge links. Objects can still be recognized using shape even if some parts are missing.

map in the following way. First, single pixel wide edge chains are obtained using the edge linking method of Peter Kovesi [16]. Then each branching chain from a junction point is treated as an edge segment. An example can be found in Fig. 5(b), where different color represents different edge segments. The connected edges are partitioned to several branches at junction points.

We observe that if edge segments are long and straight, they are more likely to correspond to a contour part of the model. Based on this observation, we further group Peter Kovesi's edge linker result to generate fairly long and straight edge segments. At this step, no shape prior is taken into consideration for contour grouping. We apply following two rules.

- Rule 1 (Linking): Connect two edge segments if the distance between two of their endpoints is less than D_T and the corresponding turn angle is less than K_T, and repeat the process until there are no more edge segments to link. In cases where there are more than two edge segments at a junction, edge segments with smaller turn angle should be connected first.

- Rule 2 (Decomposing): Split a given edge segment at a point with turn angle larger than K_T, and continue doing this till there is no more edge segments to split.

In our work, we set $D_T = 10$ pixels in most cases, and K_T is measured by curvature with a certain scale. K_T is typically set to 0.5. In Fig. 5(c), it can be seen that that short edge fragments are linked to form longer edge segments at junction points following *rule 1*, and in Fig. 5(d), all edge segments are decomposed to fairly straight segments. We apply our edge rules to all the test images in the ETHZ dataset.

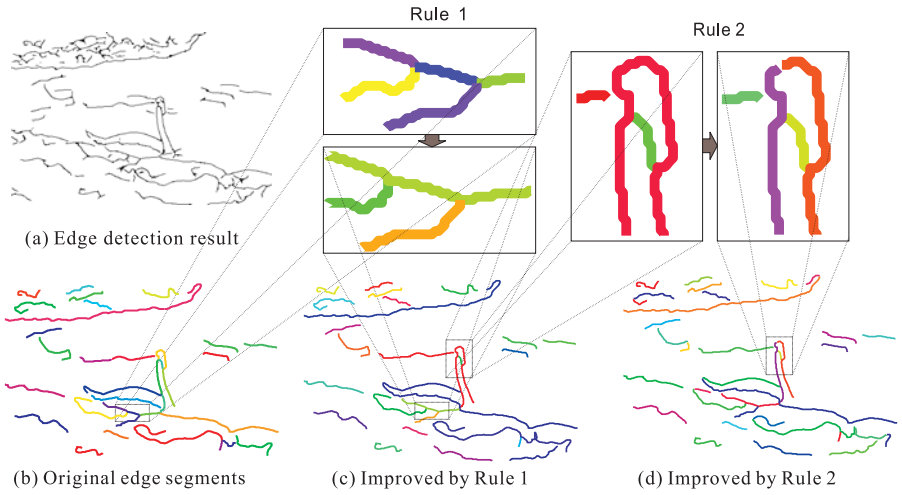


Figure 5: Decomposing and linking of edge segments in a real image with our two rules described in the text.

To summarize, we obtain a set of edge fragments $E = \{e_i\}_{i=1}^{n}$ by applying Peter Kovesi's edge linker [16] and our two rules. A sample set of resulting edge fragments can be seen in Figs. 5(d) and 2(a).

2.3. Efficient centroid voting with part bundles

Given a set of model fragments F and edge fragments E, the task now is to find a candidate subset of E such that the global shape composed by it matches to that of $\mathcal{B}$. Matching F and E is a hard combinatorial matching problem. In [38], *many-to-many* matching is utilized and a context selection scheme is introduced. They solve a relaxed version (that utilizes additivity of shape context representation) instead of the discrete version with linear programming.

In contrast by taking advantage of the exclusion property of part bundles and our edge linking rules, we can model the mapping from $\mathcal{B}$ to E as a *one-to-one mapping*. Denote such a mapping as $\pi : \{1, ..., m'\} \rightarrow \{0_1, 0_2, \ldots, 0_{m'}, 1, ..., n\}$, such that $B_i \in \mathcal{B}$ is mapped to edge fragment $e_{\pi(i)} \in E$ or empty edge when $\pi(i) = 0_i$[2]. It can be seen that each such mapping can be considered as a part configuration obtained by the $\{e_{\pi(i)}\}_{i=1}^{m'}$ in the image and our task is to find an optimal one $\widehat{\pi}$ that maximizes global shape similarity between the part-configuration and

[2] 0_i denotes a dummy fragment that B_i maps to for handling missing parts.

the model contour. Searching among all configurations is very expensive since there are $\prod_{i=0}^{m'-1}(n+m'-i) = O((n+m')^{m'}) \approx O(n^{m'})$ possible configurations.

As mentioned in the previous section, our part bundle generates a very concise model decomposition, with $m' \approx 4$ for most objects. Taking advantage of this fact, we design an efficient voting scheme for searching in the configuration space. This allows us to explore the configurations explicitly using *any* high-level shape matching tool.

The basic idea is to use fragment correspondences between F and E to vote for the object centroid. Specifically, for an image fragment $e \in E$ and a model fragment $f \in F$, we first build the mapping between the sample points. Using sequences of tangent directions along e and f as their shape descriptors, we perform sequence matching with the Smith-Waterman algorithm[34]. Then, using the part-centroid relation from the model, we can project the expected centroid of the object to the image. The scale transformation is estimated by the length ratio of the image fragment e and the model fragment f. For each $f \in F$, we rank all fragments in E according to their similarities to f as computed above. Then we choose the best two matches for centroid voting. Consequently, the whole voting procedure generates $2m$ centroid votes (see Fig. 2(c) & (d)).

The correspondences in the optimal configuration $(\widehat{\pi}(1), \ldots, \widehat{\pi}(m))$ should consistently estimate the centroid of the object. Assuming there exists a centroid whose K-NN centroids correspond to the optimal fragment-bundle correspondences, we just need to examine part-configurations obtained from K-NN centroids around all $2m$ centroids (see Fig. 2(e)). A group of K-NN centroids gives us $\left(\frac{K}{m'}\right)^{m'}$ part-configurations. So we have to examine a total of $2m\left(\frac{K}{m'}\right)^{m'}$ hypotheses or part configurations. We denote the set of these hypotheses as $\mathcal{H}$. The number of configurations to be explored is exponential in the number of part bundles but usually $m' \approx 4$ for most of the models. Also since $m << n$, $2m\left(\frac{K}{m'}\right)^{m'} << O(n^{m'})$ as long as $K << nm'$. Theoretically, a larger K implies higher probability to get the optimal configuration, thus, the larger the value of K, the better the performance is. However, the search time increases significantly with the increasing of the value of K, and we observe that the performance improves only slightly when K is larger than 20. This is explained by the fact that we have a concise model structure and the improved edge linking method allows one-to-one mapping. We chose $K = 20$ empirically to strike a balance between performance and time complexity.

Note that the proposed searching approach uses a brute force scheme, which provides a more robust configuration estimation than traditional solutions that usu-

ally rely on mode seeking or relax the matching problem. Though more reliable, brute force solutions are usually forbidden due to their high computational complexity. The key factor that makes our solution computationally trackable is the part bundle model and the one-to-one mapping scheme. In particular, we benefit from the fact that the number of part bundles m' is small. This fact is a general property that is rooted in human visual perception, since humans decompose a given shape only in a small number of parts of visual form [25].

3. Hypotheses Evaluation

3.1. Utilizing local and global shape information

Our part configurations are groups of image edge fragments and our models are based on contour parts. Hence it is natural to use shape constraints for evaluating the part configurations. Instead of just using geometric relationship between contour parts, we use global shape similarity to judge the quality of candidates.

We perform a coarse-to-fine shape matching to evaluate the hypotheses (part configurations). We first eliminate hypotheses $h_i \in \mathcal{H}$ whose cumulative part-to-fragment similarities measured using [34] is below a certain similarity threshold. Then for the remaining hypotheses we use Shape Context [1] which is a well known global shape descriptor and has demonstrated excellent performance for shape analysis tasks. We use it to estimate the similarity between a model and a hypothesis. The shape distance between them is defined by the their shape context distance ([1]) $d_{sc}(h_i, \mathcal{B})$. We then pick the best candidate according to d_{sc}:

$$\underset{h_i}{\operatorname{argmin}}\, d_{sc}(h_i, \mathcal{B}) \tag{3}$$

3.2. Combining shape and appearance

In addition to shape, appearance information often provides distinctive cues for object detection and recognition. Appearance can be helpful in cases of objects with distinctive texture and where the object model does not capture the variance in the shape. In this subsection, we describe our investigation along this line.

We use the popular bag-of-features [36, 32] framework for appearance analysis. First, local patch descriptors are generated from training images using SIFT [20], which are used to build a dictionary $\mathcal{D} = \{w_i\}_{i=1}^{M}$ using k-means. Second, all local features in a window are mapped to visual words in $\mathcal{D}$, and the window is then represented by its word frequency vector. Third, we collect all such word frequency vectors to form a training set, and use support vector machine (SVM) [31] to learn a classifier.

For object class, Giraffes, the neck in the model does not capture the variance in the neck poses of the test images. Further, Giraffes have distinctive texture. For this class we evaluate the hypotheses by combining appearance and shape. Again we first eliminate part-configurations whose cumulative part-fragment similarities is below a certain threshold. Then for each of the remaining hypothesis h, we use the texture inside the smallest bounding box enclosing the fragments to get its visual word frequency representation. Then we measure its relevance to a class using the confidence of the learned SVM model, denoted as $d_{app}(h)$. Combining shape context and appearance similarity, we have the following weighted evaluation criterion for picking the best hypothesis:

$$\underset{h_i}{\operatorname{argmin}}\, d_{sc}(h_i, \mathcal{B}) + \lambda\, d_{app}(h_i) \tag{4}$$

where λ balances the weight between shape and appearance.

4. Experimental Results

We present results on the ETHZ shape classes ([8]). This dataset has 5 different object categories with 255 images in total. Many objects are surrounded by extensive background clutter and have interior contours. All categories have significant intra-class variations, scale changes, and illumination changes. The advantage of our method is to be able to identify good part-configurations among clutter. The ability to handle deformations depends on the capacity of global shape matching module or in other words the hypothesis evaluation method.

4.1. Experiments using shape information

Similar to the experimental setup in [38], we use only the single hand-drawn models provided for each class. For each model we introduce breakpoints at high curvature contour points. The part bundles created from the contour models can be seen in Fig. 1. In addition to longer contour segments, we need to select shorter ones, since contour parts may be missing in edge images.

Since we compare our results to the state-of-the-art [38], we use Precision vs. Recall (P/R) curves for quantitative evaluation. To compare with the results of [7] that are evaluated by detection rate (DR) vs. false positive per image (FPPI), we translate their results into P/R values as in [38]. Fig. 6 shows P/R curves for the three methods: Contour Selection method of [38] in black, that of [7] in green, and our method in red. Our approach performs better than method in [7] on three

categories, outperforms the Contour Selection method [38] on three categories, and outperforms both methods on two categories ("applelogos" and "swans").

We also compare the precision to [38] and [7] at the same recall values in Table 1. The precision / recall pairs of [38] and [7] are quoted from [38]. We use the same criterion for correct detection as in [7]. Sample successful detections, failure cases and false-positives are shown in Fig. 7.

On a PC with 2.3GHz Pentium (D) CPU and 3GB RAM, the average processing time of our method in MATLAB implementation is around 25 seconds per image.

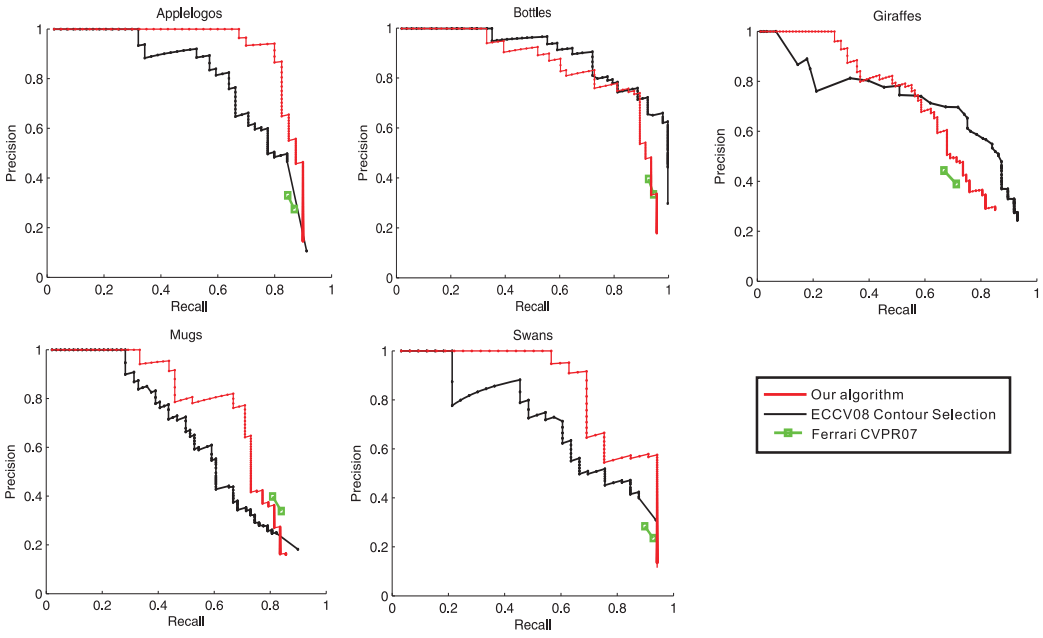


Figure 6: Precision/Recall curves of the Contour Selection method by Zhu et al. [38] (black) and the method by Ferrari et al. [7] (green) on ETHZ shape classes. Our method (red) performs significantly better on two categories viz. Applelogos and Swans.

4.2. Experiments using shape and appearance information

We now present results of using appearance information on the class of giraffes that have distinctive texture because of their skin. We pick the first 16 images from the class to obtain the code book $\mathcal{D}$ of 100 visual words. We only selected the SIFT features inside the ground truth bounding boxes. The detection is based on a nearest neighbor retrieval according the distance d_{mix}, with $\lambda = 8$.

Class (Recall%)	[7]	[38]	Our method
Applelogos (86.4%)	32.6%	49.3%	**56.0%**
Bottles (92.7%)	33.3%	**65.4%**	48.0%
Mugs (83.4%)	**40.9%**	25.7%	27.8%
Giraffes (70.3%)	43.9%	**69.3%**	50.0%
Swans (93.9%)	23.3%	31.3%	**59.0%**

Table 1: Precision / recall pairs of the method of [7] and the Contour Selection method of [38] are quoted from [38]. Our precision is higher than [7] on four classes (except for "mugs") and higher than [38] on three classes (except for "bottles" and "giraffes").

Fig. 8 shows the P/R curves using only shape and combining shape and appearance. Fig. 9 shows the improvement of using appearance information on two sample images. As can be seen using appearance information can significantly improve the detection performance.

5. Conclusions

In this paper we study the problem of contour based object detection. A new contour model, part bundles, is proposed that decomposes contour fragments into subsets. We show that this decomposition efficiently groups competing fragments and therefore enables brute force search hypothesis voting, which generates high quality and concise candidate detection sets. With these candidate sets, we use global shape similarity for false positive pruning and achieved excellent performance on ETHZ dataset. In addition, we show that appearance information, which can be easily integrated into our approach, can help to achieve further improvement for objects with distinctive texture when contour model and shape matching do not sufficiently capture the deformability of objects. Integrating such additional cues is straightforward in our framework.

Acknowledgment

This work was supported in part by the NSF Grants IIS-0812118, BCS-0924164, IIS-0916624, AFOSR Grant FA9550-09-1-0207, National Natural Science Foundation of China 60903096, 60496315, 60802009 and National 863 plans projects 2008AA01Z204, 2009AA01Z205. N. Adluru is supported by Computation and Informatics in Biology and Medicine (CIBM) program and Morgridge Institute for Research at the UW-Madison.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:705–522, 2002. 2, 3, 10

[2] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):849–865, 1988. 2

[3] Y. Chen, L. Zhu, C. Lin, A. Yuille, and H. Zhang. Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 289–296. MIT Press, Cambridge, MA, 2008. 3

[4] P. Felzenszwalb and D. McAllester. A min-cover approach for finding salient curves. In *CVPRW: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC, USA, 2006. IEEE Computer Society. 2

[5] P. F. Felzenszwalb and J. Schwartz. Hierarchical matching of deformable shapes. In *CVPR*, 2007. 2

[6] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):36–51, 2008. 2

[7] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, pages 1–8, 2007. 2, 11, 12, 13

[8] V. Ferrari, T. Tuytelaars, and L. J. V. Gool. Object detection by contour segment networks. In *ECCV*, pages 14–28, 2006. 5, 11

[9] M. Galun, R. Basri, and A. Brandt. Multiscale edge detection and fiber enhancement using differences of oriented means. In *ICCV '07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*. IEEE Computer Society, 2007. 2

[10] K. Grauman and T. Darrell. "The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features". *IEEE Int'l Conf. on Computer Vision*, II:1458-1465, 2005. 4

[11] W. E. L. Grimson. The combinatorics of object recognition in cluttered environments using constrained search. *Artif. Intell.*, 44(1-2):121–165, 1990. 2

[12] W. E. L. Grimson. *Object recognition by computer: the role of geometric constraints*. MIT Press, Cambridge, MA, USA, 1990. 2

[13] W. E. L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(4):469–482, 1987. 2

[14] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1778–1785, Washington, DC, USA, 2005. IEEE Computer Society. 3

[15] D. Hoiem, A. Stein, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *International Conference on Computer Vision (ICCV)*, October 2007. 2

[16] P. D. Kovesi. MATLAB and Octave functions for computer vision

and image processing. School of Computer Science & Software Engineering, The University of Western Australia, 2008. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>. 4, 7, 8

[17] L. J. Latecki and R. Lakamper. Convexity rule for shape decomposition based on discrete contour evolution. *CVIU*, 73:441–454, 1999. 5

[18] L. J. Latecki, C. Lu, M. Sobel, and X. Bai. Multiscale random fields with application to contour grouping. In *Neural Information Processing Systems Conf. (NIPS), Vancouver*, December 2008. 3

[19] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 878–885, Washington, DC, USA, 2005. IEEE Computer Society. 2

[20] D. G. Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999. 10

[21] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 2

[22] D. Martin, C. Fowlkes, and J. Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. In *IEEE Trans. Pattern Anal. Mach. Intell.*, volume 26, pages 530–549, 2001. 6

[23] G. McNeill and S. Vijayakumar. Part-based probabilistic point matching using equivalence constraints. In *Neural Information Processing Systems*, pages 969–976, 2006. 2

[24] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proceedings of the European Conference on Computer Vision*, 2006. 2

[25] K. Siddiqi and B. B. Kimia. Parts of Visual Form: Computational Aspects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17, pages 239–251, 1995. 10

[26] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 503–510, Washington, DC, USA, 2005. IEEE Computer Society. 2

[27] A. Stein, D. Hoiem, and M. Hebert. Learning to find object boundaries using motion cues. In *IEEE International Conference on Computer Vision (ICCV)*, October 2007. 2

[28] A. Tamrakar and B. B. Kimia. No grouping left behind: From edges to curve fragments. In *ICCV '07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*. IEEE Computer Society, 2007. 2

[29] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, pages 127–133, 2003. 2

[30] N. Trinh and B. B. Kimia. A symmetry-based generative model for shape. In *ICCV*

'07: Proceedings of the Eleventh IEEE International Conference on Computer Vision. IEEE Computer Society, 2007. 2

[31] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995. 10

[32] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1-2):61–81, 2005. 4, 10

[33] L. Wang, J. Shi, G. Song, and I. Shen. Object detection combining recognition and segmentation. In *ACCV07*, pages 189–199, 2007. 2

[34] S. T. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981. 9, 10

[35] J. Willamowski, D. Arregui, G. Csurka, C. Dance, and L. Fan. "Categorizing Nine Visual Classes using Local Appearance Descriptors", *ICPR Workshop Learning for Adaptable Visual Systems*, 2004. 4

[36] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. "Local Features and Kernels for Classifcation of Texture and Object Categories: A Comprehensive Study." *International Journal of Computer Vision*, 73(2):213-238, 2007. 4, 10

[37] Q. Zhu, G. Song, and J. Shi. Untangling cycles for contour grouping. In *ICCV '07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*. IEEE Computer Society, 2007. 2

[38] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV08*, 2008. 2, 5, 8, 11, 12, 13, 18

[39] S.-C. Zhu and D. Mumford. *A Stochastic Grammar oF Images*. Now Publishers, 2007. 3

[40] X. Bai, X. Wang, L.J. Latecki, W. Liu, Z. Tu. Active Skeleton for Non-rigid Object Detection *ICCV*, 2009. 2

[41] S.-C. Zhu and D. Mumford. *A Stochastic Grammar oF Images*. Now Publishers, 2007.

Figure 7: Sample detection results. The edge map is overlaid on the image in white. The detected fragments are shown in black. The corresponding model parts are shown in top left corners. The bottom most row (enclosed in red) shows some failure cases and false positives.

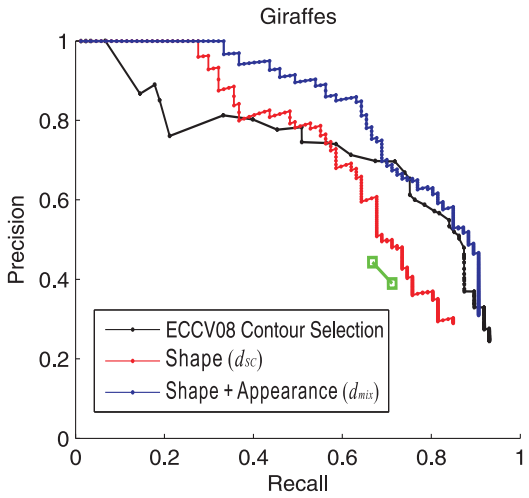


Figure 8: The Precision/Recall curves using shape based evaluation (d_{SC}) and combining shape and appearance (d_{mix}) on the class of giraffes. We also show the results of [38] for comparison. The combination of shape and appearance significantly improves the results.

Figure 9: Two examples to show the improvement obtained by combining appearance information. Top row: Results of using only shape. Bottom row: Results of combining appearance with shape.