Accepted Manuscript

Constellational Contour Parsing for Deformable Object Detection

Chen Huang, Tony X. Han, Wenming Cao, Zhihai He

PII: S1047-3203(16)30032-3
DOI: http://dx.doi.org/10.1016/j.jvcir.2016.03.028
Reference: YJVCI 1725

To appear in: *J. Vis. Commun. Image R.*

Received Date: 18 July 2015
Revised Date: 6 February 2016
Accepted Date: 30 March 2016

Please cite this article as: C. Huang, T.X. Han, W. Cao, Z. He, Constellational Contour Parsing for Deformable Object Detection, *J. Vis. Commun. Image R.* (2016), doi: http://dx.doi.org/10.1016/j.jvcir.2016.03.028

Constellational Contour Parsing for Deformable Object Detection

Chen Huang[a], Tony X. Han[a], Wenming Cao[b], Zhihai He[a,*]

[a]*Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO, USA, 65211*
[b]*Department of Computer Science, Shenzhen University, Shenzhen, Guangdong Providence, China, 518060*

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 object, it is first decomposed into fragments, then in the test image we simultaneously perform selection of relevant contour fragments in edge images, grouping of the selected contour fragments, and finding best geometry-preserving matching to model contours. Finding the best matching is inherently a computationally expensive problem. To address this challenge, we developed local shape descriptors and an additive similarity metric function which can be computed locally while preserving the capability of matching deformable shapes globally. This allows us to establish a constellational shape parsing framework using low-complexity dynamic programming to find optimal configuration of contour segments in test images to match the model contour. To effectively detect objects with large deformation, we augmented the metric function with a local motion search, modeled the

*Corresponding author
Email addresses: chenhuang@mail.missouri.edu (Chen Huang), hantx@missouri.edu (Tony X. Han), caowe@missouri.edu (Wenming Cao), HeZhi@missouri.edu (Zhihai He)

relationship between different shape parts using multiple concurrent dynamic programming shape parsers. Our experimental results show that the proposed method outperforms the state-of-the-art contour-based object detection algorithms on two benchmark datasets in terms of average precision.

Keywords: object detection, shape matching, optimization, additive similarity measure, dynamic programming, constellational contour parsing

1. Introduction

Object detection is an important and challenging task in computer vision. It allows localization of previous unseen objects in new images. In general, two main paradigms can be distinguished: *appearance-based* and *contour-based*. Appearance-based approaches [1, 2, 3, 4] form the dominant paradigm using local image patch features and constructing rules and models with powerful classification and learning methods for object detection. These types of methods often require a large image dataset for training. Recently, several contour-based methods have demostrated proficiency at the task of object detection, such as [5, 6, 7] and [8]. Compared to other image cues, the outline contour provides a set of powerful and robust features for object characterization which are largely invariant to changes in illumination, object colors and texture [9, 10]. More importantly, the outline contour can efficiently represent image structures with large spatial extents [11]. Recent work on contour-based object detection has shown that, using a single hand-drawn example (e.g. a bottle sketch) instead of a large set of training data, enables detection of objects with a wide variety of textures, poses, and sizes from highly cluttered images with significant illumination variations and shape deformations [12, 13, 14, 15, 16].

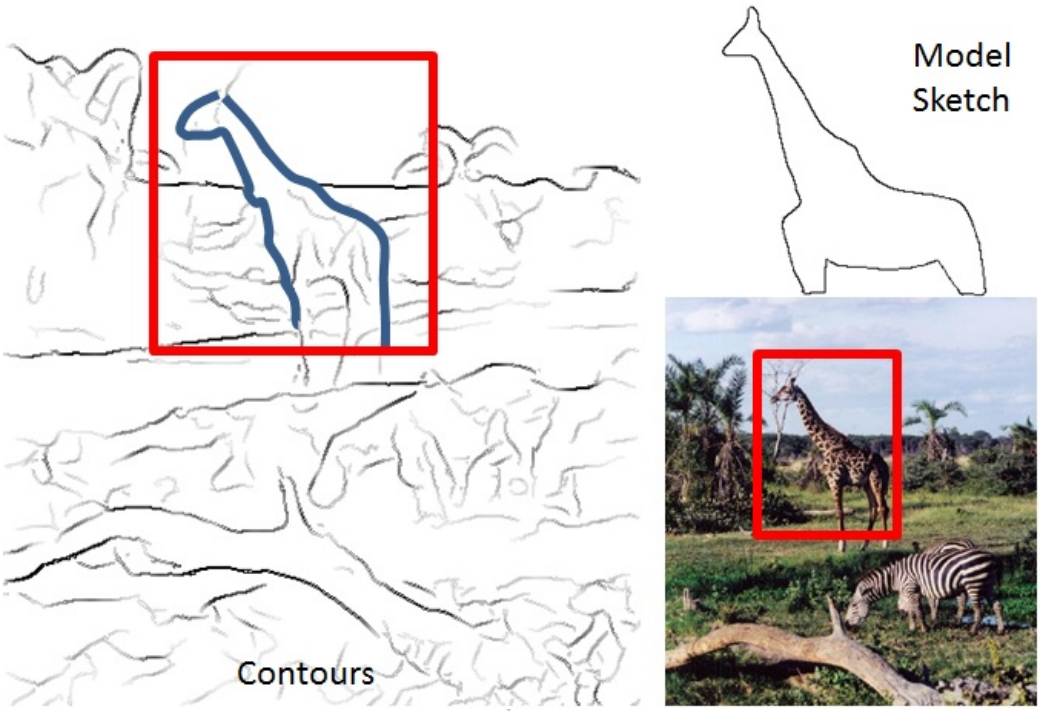

Figure 1: An example of contour-based object detection results generated by the proposed method.

Contour-based object detection is a challenging task. It operates on the edge or contour map extracted from the image obtained by an edge detector, such as Canny [17] or Pb [18]. The edge pixels are grouped into edge fragments in a bottom up process using an edge-linking algorithm, e.g., [19]. Given the contour of target object as a model, the goal of contour-based object detection is to select a small subset of edge fragments that minimize the dissimilarity to the model contour. as shown in Fig. 1.

Shape contours obtained from object segmentation often exhibit a large degree of intra-class of variations and inter-class ambiguities [20]. For example,

animals of similar species are often very similar to each other, except for some small distinguishable features.

During contour-based object detection, we need to identify the subset of contour segments from the cluttered image to best match the example shape model. The shape description, representation, and matching scheme should be able to accommodate large shape deformation of the same object class. Furthermore, the detection process also needs to deal with a so-called hallucination problem, where a subset of edge segments in the background form a shape similar to the target object, resulting in a false positive during detection [14].

This work presents a new method, called *constellation contour parsing* (CCP), for contour-based object detection. Given a shape prototype of the object, such as a hand-drawn sketch or a model image, we aim to search through the clouds of contour segments in the test image to identify the optical configuration of contour segments which best matches the shape prototype, as shown in Fig. 1. We use the term *constellation* because this problem is similar to finding a specific constellation of stars in the sky.

Finding the best constellation is combinatorial optimization problem. After developing local shape descriptors and an additive similarity metric function which can be computed locally while preserving the capability of matching deformable shapes globally, we establish a constellational shape parsing framework using dynamic programming. To effectively detect objects with large deformation, we perform implicit shape parts analysis, model the relationship between different shape parts using multiple concurrent dynamic programming shape parsers, and finalize the detection result using voting scheme. Our experimental results show that the proposed method outperforms the state-of-the-art contour-based object

detection algorithms on two benchmark datasets.

Grouping edge pixels into contours can increase the discriminative power compared to considering an unstructured set of edge pixels. However edge and contour detection are sensitive to local image changes and noise. The contour of one object may be broken into several fragments. Some fragments of the target object could be missing. The object contour could wrongly connect with contours in the background clutter [21]. These types of errors can significantly limit the applications of the image contour-based object detection method. CPP does not require a complete edge contour of objects, because what CPP are trying to search is only a set of contour segments rather than complete edge contour. So the proposed detection method is robust to broken edges and can obtain very impressive detection results even when the edge quality is bad. This fact is illustrated in Fig. 2.

In this paper, we choose to explore an object detection system that exploits only contour-based information, because we believe shape contour provides a powerful and generic feature since it is invariant to extreme lighting conditions and large variations in texture or color. Clearly the eventual goal of any detection algorithm is to combine sensibly many different useful types of features, but for the purpose of this paper, we deliberately ignore these to show just how powerful the shape contour is.

The major contribution of this work lies in the following three aspects. (1) We introduced the scheme of constellational contour parsing (CCP) with anchor points, observation rays, and additive shape metric function to convert a combinatorial contour grouping problem into a dynamic programming optimization problem with low complexity. (2) We developed an implicit shape parts analy-

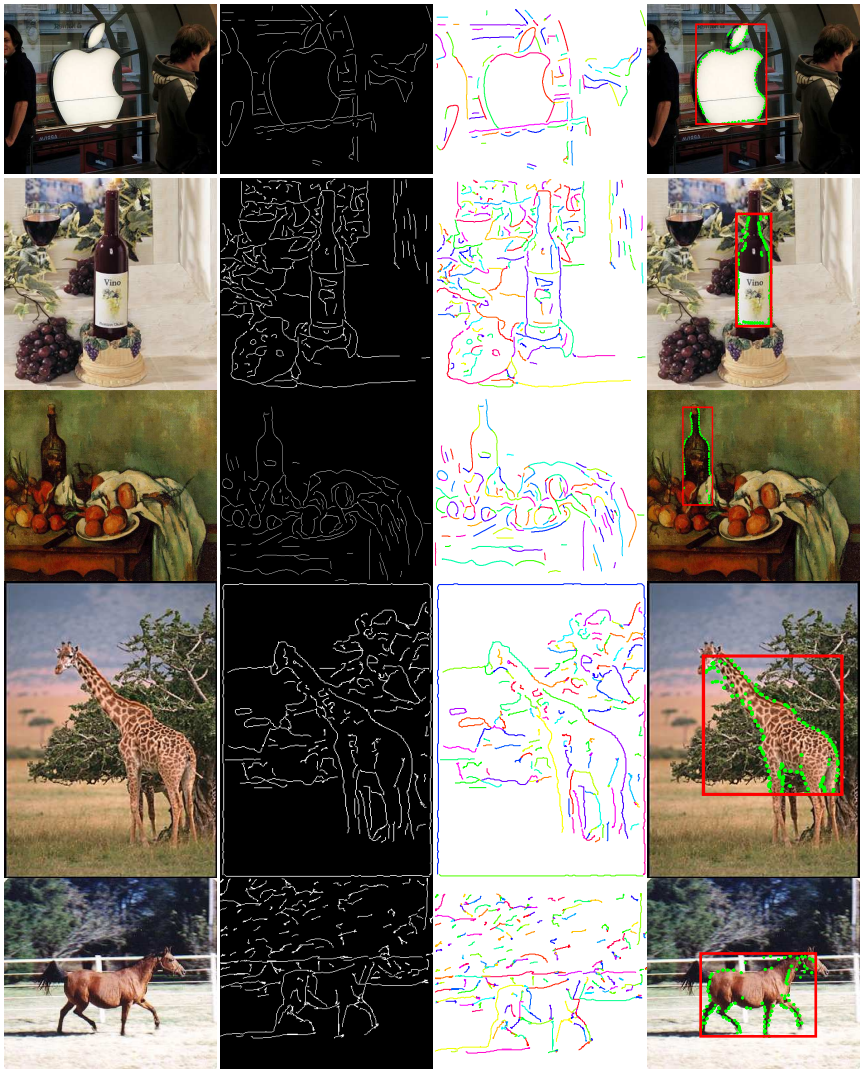

Figure 2: A few detection results when edges are broken. The first column shows original test image; the second column shows the edge maps computed using Pb edge detector; the third column shows edge-linking results by [19]; the last column shows our detection results.

sis scheme using multiple observation points to handle object deformation. (3) The proposed method is able to detect objects with different scales, rotations, per-

spective changes, and deformations at low computational complexity and achieves very encouraging and highly competitive results.

The remainder of the paper is organized as follows. Section 2 reviews the related work on contour-based object detection. Section 3 provides an overview of the proposed system. The constellational contour parsing is explained in Section 4.Experimental results are presented in Section 5.

Concluding remarks and discussions are given in Section 6.

2. Related Work

Shape is widely considered as one of the best features in computer vision [22]. In contrast to gradient and texture-based representations, shape is more descriptive at a larger scale, ideally capturing the object of interest as a whole [23] [24] [25]. A successful shape matching scheme consists of two components: a compact and accurate shape representation, and an efficient as well as robust distance measurement. There are many ways to represent shapes.Examples include axial representation [26, 27], primitive-based representation [28], reference points and projection based representation [29], histograms of oriented gradients [2], hierarchical Fourier descriptor [30] etc.

Measuring the similarity between two shapes often can be done in two ways: 1) by computing the direct difference in features extracted from shape contours, which are invariant to the choice of starting points and robust to a certain degree of deformation; and 2) by performing matching to find the detailed point-wise correspondences to compute the differences, such as the famous Shape Context [31] and Self-Similarity [32]. The latter has recently become dominant due to its ability to capture intrinsic properties, thus leading to more accurate similarity

measures. Inner Distance Shape Context (IDSC) improves the distance of the Shape Context for articulated shapes. In Hu *et al.*'s recent article [33], an adaption of the HOG descriptor called the Gradient Field HOG (GF-HOG) was used for sketch based image retrieval.

However, methods above are not directly applicable for detecting and matching objects in clutter images. In this paper, we focus on contour-based detection. Early work for object detection through shape matching can be dated to the late 90s, e.g., the chamfer matching algorithm by Borgefors *et al.* [34]. Recent year, a large range of contour-based object detection methods[21] [13] [16] [7] have been proposed, and many of them have achieved state-of-the-art performance only by utilizing edge information. A common approach built into many of these methods is to define a distance measures between a given shape to the objects in the image. Usually, the matching and detecting steps are performed at the same time.

Ferrari *et al.* [12, 5] built a network of nearly straight adjacent segments and detected objects in images by deformable template matching. Zhu *et al.* [14] formulated the shape matching of a contour in clutter as a set-to-set matching problem and presented an approximate solution to the hard combinatorial problem by using a voting scheme. They used a context selection scheme by algebraically encoding shape context into linear programming. Ravishankar *et al.* [11] used short segments to approximate the outer contour of objects. They decomposed the model shapes into segments at high curvature points. Dynamic programming is used to group the matched segments in a multi-stage process which begins with triples of segments. Lu *et al.* [15] first decomposed the shape model into several part bundles, and used particle filters to simultaneously select and group relevant contour segments in the test images and then matched them against the model

shape; however [15] also expected image contours to have consistent fragmentation across images, which is not always the case. To address the non-rigid object deformation, Bai *et al.* [21] used the skeleton information to capture the main structure of an object applying the Oriented Chamfer Matching to match the model parts to test images. Ma *et al.* [16] proposed a partial shape matching through local deformation to capture the global shape similarity for object detection.

As stated before, learning is often involved to train a shape model for detection. Shotton *et al.* [9] developed boosted contour-based shape features for object detection. Opelt *et al* [10] learned codebook of contour fragments first, then use Chamfer distance to match learned fragments to edges in the test images. In both approaches the major effort is to learn discriminative combinations of a boundary part as a weak classifier using boosting to build a strong detector. Srinivasan *et al.* [35] addressed the contour grouping problem as many-to-one matching, and used this scheme in both training and testing phases. For the purpose of improving detection and score ranking, a sophisticated training process was designed in which latent SVM was used to guarantee that the many-to-one score is turned discriminatively. Most Recently, Wang *et al.* [7] proposed a Fan Shape Model (FSM) which can preserve discriminative power while allowing for substantial shape deformation, however in FSM all rays do the selection simultaneously, and have no interaction with choices of the other rays; constraints of supported contours and distance consistency are the only evaluation after the detection has been completed. Our work is inspired by this paper but we included local and global constraints in the detection phase.

3. Overview of the Proposed Approach

Given a prototype contour model, such as a hand-drawn sketch, our task was to find the optimal configuration of contour segments from the test image to match the prototype, as shown in Fig. 1. This is a combinatorial optimization problem, which is very computationally expensive, because the number of contour segments in a cluttered image was very large, the computational complexity was prohibitive. To address this challenging issue, we proposed the following idea. Our method is illustrated in Fig. 3. First we densely sampled a grid of anchor points O_n, $1 \leq n \leq N$ within the test image, where N is the total number of anchor points. Each anchor point O_n generated an observation of the contour map at multiple observation angles $\theta_m = m \times 2\pi/M$, $1 \leq m \leq M$. At each observation angle θ_m, a ray originating from the anchor point O_n, denoted by r_m, intersected with a set of contour segments in the image, denoted by $\{x_{m,k}, 1 \leq k \leq K_m\}$. Note K_m is a scalar denoting the total number of intersecting contour segments for one particular ray r_m, and K_m varies between observation angles. K_m could also be 0 where no contour segments are intersecting with this ray. In our experimental, we first revise the longest side of input image to be 600, while keep the original aspect ratio; then sample anchor points O_n with a step size of 8.

For the prototype contour model, we also densely sampled multiple anchor points P_l, $1 \leq l \leq L$, inside the shape. Following the same procedure, we generated the observation rays and intersection contour segments T_m, as illustrated in Fig. 4(b).

With the observation rays from the anchor point $\{r_m\}$ and intersecting contour segments $\{x_{m,k}\}$ obtained from the test image, we were able to formulate the contour-based object detection problem into a multi-stage decision problem. As

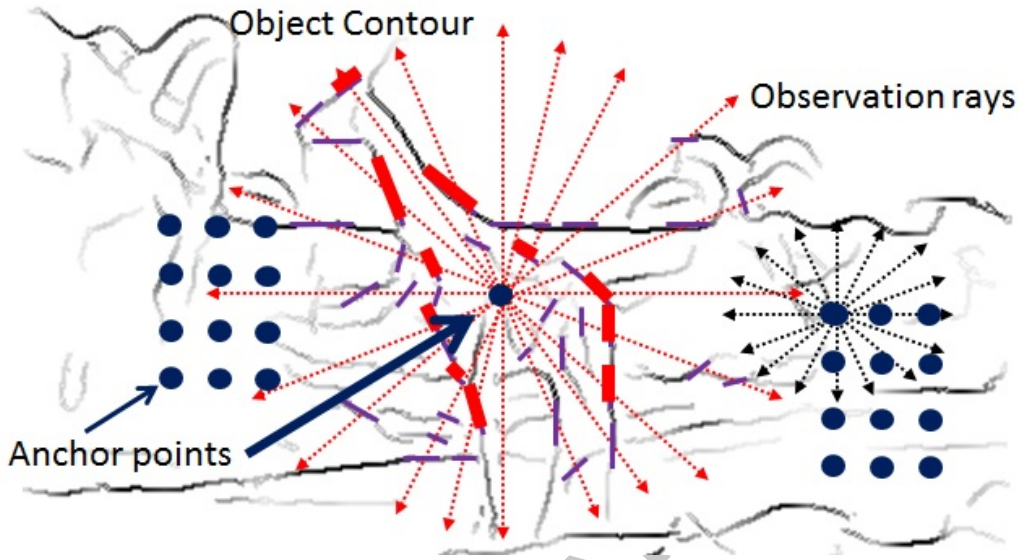

Figure 3: Scanning the contour map of the test image with a grid of anchor points and observation rays. For each ray, it will intersect with a bunch of segments in contour map. Intersection contour segments are highlights with red and purple color in this figure. Select the best anchor point and its corresponding segments sets, which is marked as red and bold, is the purpose of our algorithm.

illustrated in Fig. 4(c), our task was to find the best path of contour segments $x_{m,k}$ across these M observation rays in test image that best matches the model contour segments T_m. To this end, Section 4 developed local shape descriptors and an additive similarity metric function to measure the similarity between a continuous shape and a constellation of contour segments. With this additive metric function, we were able to solve the matching and detecting problem using dynamic programming. We refer to the above procedure as constellational contour parsing (CCP). Using CCP, we can find the optimal configuration of contour segments, denoted by $\mathcal{C}(O_n, P_l)$, for anchor point O_n in the target image to match the

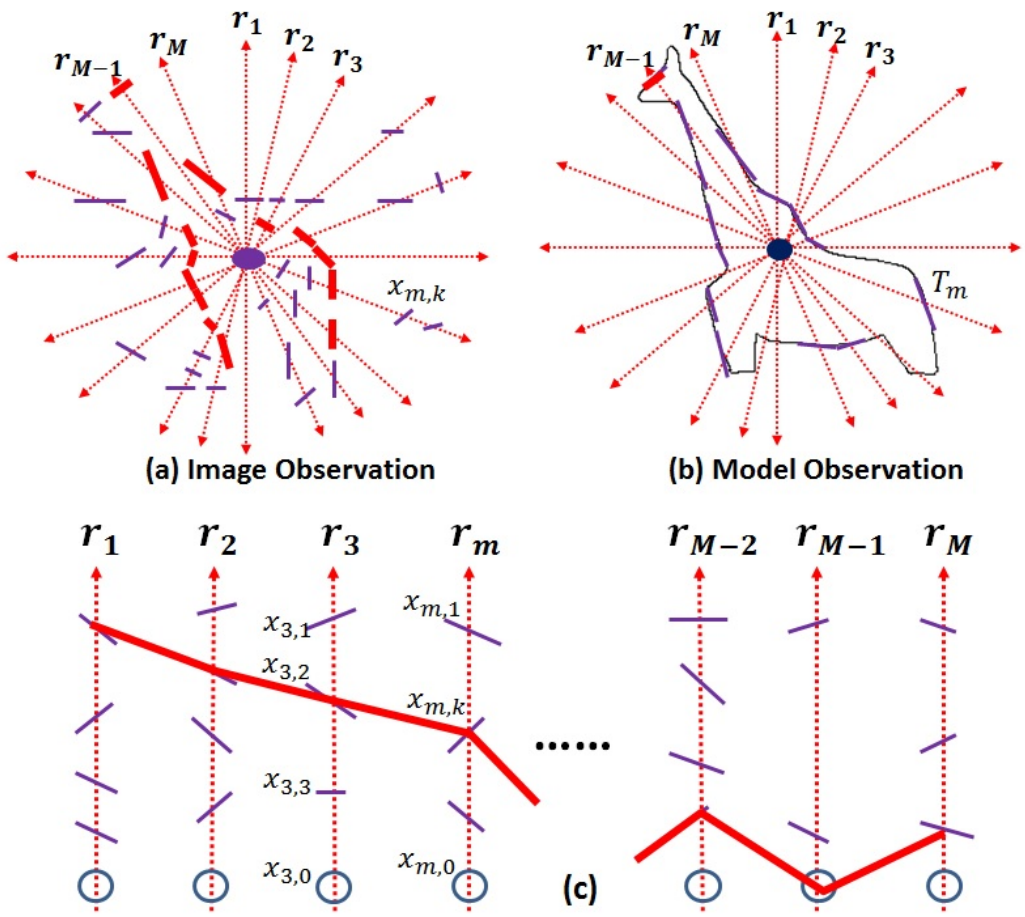

Figure 4: Overview of the proposed constellational contour parsing using dynamic programming. (a) is an example of an anchor point O_n in test image, its corresponding observation rays and intersection contour segments. (b) is an anchor point P_l in the model shape, its corresponding observation rays and intersection contour segments. (c) is the illustration of our dynamic programming approach used to select best contour segments set in (a) that have maximum similarity to segments set in (b)

model shape with anchor point P_l. The corresponding matching score is denoted by $\alpha(O_n, P_l)$.

Our CCP-based object detection operates as follows. We selected multiple anchor points $\{P_l\}$ in the model shape to capture the characteristics of different parts of the object. For each model anchor point P_l, we scaned every observation point O_n in the test image and found the best constellation of contour segments $\mathcal{C}(O_n, P_l)$ around O_n to match the model shape. The best matching score is denoted by $\alpha(O_n, P_l)$. The anchor point with the maximum matching score indicates the object detection result, and the associated constellation of contour segments delineates the object boundary:

$$\mathcal{C}^*(P_l) = \arg \max_{O_n, 1 \leq n \leq N} \alpha(O_n, P_l). \tag{1}$$

In the following section, we explain the major components of the proposed method in more detail.

4. Constellational Contour Parsing Using Dynamic Programming

We observed that, using the procedure outlined in the above section, we were able to successfully convert the original combinatorial optimization problem of contour-based object detection into a dynamic programming problem. This is because, within the proposed framework, to find the best constellation of contour segments $\mathcal{C}(O_n, P_l)$, we only needed to determine which contour segment $x_{m,k}$ should be selected at the stage of r_m. In addition, to make this decision, we only needed information from neighboring rays r_{m-1} or r_{m+1}. This satisfies the Markov decision process property in dynamic programming [36]. We noticed that, during edge and contour detection, some edges or contour segments were missing. To address this issue, we introduced an *empty segment*, denoted by $x_{m,0}$ for each observation ray.

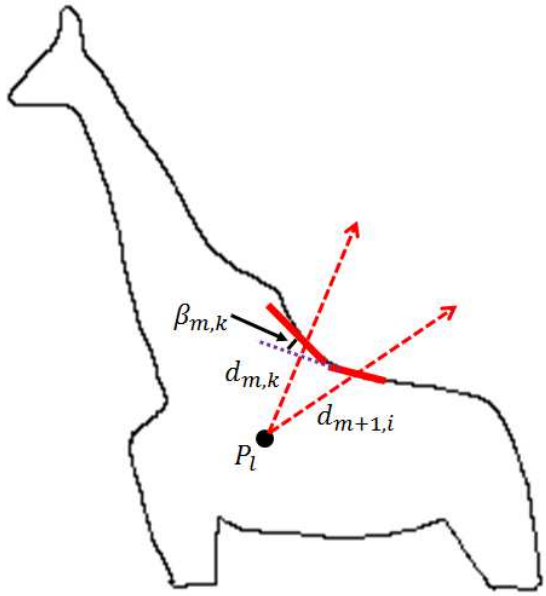

Figure 5: Changes of orientation angles of contour segments and their distance to the anchor point.

The key component in dynamic programming formulation is to construct an additive cost function to measure the fitness of the selected path. Many existing shape descriptors or matching cost metrics developed in the literature, such as the shape-tree descriptor [13], the And-Or graph model [8], and the fan shape model [7], are either global or non-additive. Our proposed shape description scheme analyzes local attributes of contours. As illustrated in Fig. 5, lets $\beta_{m,k}$ be the change of contour segment orientation at $x_{m,k}$, which is the difference between the orientation angles of the contour segments before and after the intersection point. Let $d_{m,k}$ be the distance between the anchor point and $x_{m,k}$. Similarly, we can define these two variables for the model shape, denoted by β_m^* and d_m^*,

respectively. During dynamic programming, as illustrated in Fig. 4(c), the cost for the link between nodes $x_{m,k}$ and $x_{m+1,i}$ is defined as

$$\begin{aligned} C(m+1,i,k) &= w_a \cdot |\beta_{m,k} - \beta_m^*| \\ &+ w_b \cdot \left| \frac{d_{m+1,i}}{d_{m,k}} - \frac{d_{m+1}^*}{d_m^*} \right|, \end{aligned} \tag{2}$$

where the first entry aims to capture the similarity between the test contour and the model shape on local orientation changes, while the second entry on distance changes. w_a and w_b are two normalization coefficients. Let $F(m,k)$ be the accumulated path cost from stage r_1 to stage r_m. At stage r_{m+1}, node $x_{m+1,i}$ is connected to node $x_{m+1,k}$ if

$$k = \arg \min_{1 \le j \le K_{m+1}} F(m,j) + C(m+1,i,j). \tag{3}$$

4.1. Handling Viewing Perspective Changes

We can see that the proposed method, in its original design, is able to handle scale changes, detecting objects at different scales without any additional computational complexity, since it searches all contour segments in the image around the anchor point. As we know, scale estimation has been one of the key and challenging issues in contour-based object detection [12, 37, 7], since it has direct impact on the overall computational complexity and detection performance.

We recognize that the original CCP design presented in the above section is not able to handle object rotations since it is based on a fixed angle partition and ray structure. There are two options to extend the proposed method for handling object rotations. The first option was to rotate the model shape to multiple angles,

perform the matching for each angle, and find the angle with the best match. Certainly, this will increase the overall computational complexity (linearly).

The second option was to relax the observation angle constraint. The cost function in (2) was computed at the same observation angles θ_m for both the test images and model shapes. To relax this constraint, we searched within neighboring angles or even the whole angle range. More specifically, the new cost function becomes

$$\begin{aligned} C_r(m+1,i,k) &= \min_{1\le h\le M}(w_a\cdot|\beta_{m,k}-\beta_h^*| \\ &+ w_b\cdot\left|\frac{d_{m+1,i}}{d_{m,k}}-\frac{d_{h+1}^*}{d_h^*}\right|), \end{aligned} \tag{4}$$

This is essentially a rotation motion estimation procedure. Since every contour segment is rotated by the same angle, we can use the rotation motion estimation at previous segments to predict the motion of the current and future segments. In this way, we can accurately estimate the rotation with a few additional search operations.

With this new cost function, the proposed method was able to effectively handle scale changes and rotations, which are the two dominating view changes in object detection. Algorithm 1 describes how CCP can find the best constellation of fragments in a clutter contour map. As we can see from the experimental results, other types of deformations, such as skewness caused by affine or perspective transforms, can be well accommodated by our method.

4.2. Analysis of Computational Complexity

To find best match of one pair $\alpha(O_n, P_l)$, the computational complexity is $O(K_m^2 M)$, because it is a Viterbi algorithm. Here M is the number of obser-

Algorithm 1: CCP finds best constellation of fragments

Input: model contour segments T, image edge map E

Output: best configuration of fragments in E fitting T

$O_n \leftarrow$ densely sample anchor points in image

for $n \leftarrow 1$ ***to*** N **do**

  $r_m \leftarrow$ generates ray originating from the O_n

  $x_{m,k} \leftarrow$ ray r_m intersects with E

  $F(1,k) \leftarrow initialization$

  for $m \leftarrow 2$ ***to*** M **do**

    $F(m,k) \leftarrow \arg\max_{1 \leq i \leq K_{m-1}} F(m-1,i) + C_r(m,i,k)$

  $\mathcal{C}(O_n, P_l) \leftarrow \arg\max_{1 \leq k \leq K_M} F(M,k)$

$\mathcal{C}^*(P_l) \leftarrow \arg\max_{O_n, 1 \leq n \leq N} \alpha(O_n, P_l)$

return $\mathcal{C}^*(P_l)$

vation angles for anchor point O_n, and K_m denotes the number of intersecting contour segments for one particular ray r_m, as we mentioned before. So the total complexity to compute $\mathcal{C}^*(P_l)$ is $O(K_m^2 MN)$, where N is the number of anchor points densely sampled in the test image. We carried out the experiments on a PC with an Intel Core i7-2640M 2.8GHz CPU and 8GM memory. On average, the time cost for a detection on an image is around 2$\sim$3 minutes.

4.3. Selection of Observation Points

Currently, we examined a grid of observation points O_n in the test image. To further reduce the computational complexity, we developd a simple yet effective scheme for early detection of observation points with no possibility of being inside an object. More specifically, an observation point will be skipped if its percentage of empty rays is larger than a certain threshold (e.g., 60%), or the portion of consecutive empty rays is larger than a certain threshold (e.g. 30%). A ray is empty if it does not intersect with any contours. According to our experiments, this scheme is able to reduce the overall complexity by 50-70%. Future work is planned to investigate more sophisticated and effective approaches for selecting observation points to minimize the complexity.

5. Experimental Results

We tested our method on the two challenging datasets: ETHZ shape dataset [12] and INRIA-Horse dataset [38]. The ETHZ shape dataset has five categories and contains a total of 255 images, and the INRIA-Horse dataset consists of 170 images containing one or more side-viewed horses and 170 images without horse. Both datasets provided ground truth bounding boxes for object instances.

For edge detection, we used the results from [12] which is based on the Berkeley boundary detector [18]. To convert the grayscale edge map to a binary edge map, we set all pixels with their values larger than 0.02×255 as edge pixels. This means that we do not adjust the threshold to achieve better edges and detection performance. To handle non-monotonic object shapes with multiple contours, we cascaded these contours together and treated them as a single curve. Accordingly, the observation rays and dynamic programming stages for each contour were then cascaded together.

For performance evaluations and comparisons, we followed the PASCAL criteria, i.e., a detection result is deemed as correct if the intersection of the detected bounding box and the ground truth over the union (IoU) of the two bounding boxes is larger than 50% [39, 7], otherwise detections are counted as false positive.

5.1. Experiment on ETHZ shape dataset

We used all five classes of shapes from ETHZ dataset(i.e., applelogos, bottles, giraffes, mugs and swans). There are 32$\sim$87 images in each class, and the images contain at least one and sometimes multiple instances of a class and have a large amount of background clutter. All object classes have significant intra-class variations and scale changes. The system is only given a single hand-drawn shape exemplar for each class, and we tested on all 255 images.

We focused on performance comparisons using the state-of-the-art contour-based object detection methods [40, 5, 35, 8, 41, 15], and the texture based discriminative part model [3]. We plotted the precision/recall (PR) curves in Fig. 6. We used the toolbox in [39] to calculate average precision (AP). Table 1 shows the average precision (AP) results for six methods. Our AP is only slightly lower (about 0.001) than the best result in [16]. Among the five categories, we achieved

best AP for Applelogos. For the categories of Giraffes and Bottles and Swans, we are very close to the best performance.

We also show the false positive per image (FPPI) vs. detection rate (DR) results in Fig. 7. Table 2 compares the detection rates of all methods at 0.3/0.4 FPPI. Our method achieves the best performance. We have achieved the best performance for Mugs, while our score is comparable to the best ones in the other four classes. The curve of our methods increased shapely at the beginning and reaches the peak of detection rate before 0.3/0.4 FPPI, so there is no difference in detection rates at 0.3 FPPI and 0.4 FPPI. Fig. 8 shows some samples of our detection results. Our system can accurately locate actual object contours. In addition, it can effectively handles the scenarios with multiple object instances. Fig. 9 shows some false positive detection examples.

Table 1: Comparison of average precision (AP) on the ETHZ dataset with other methods.

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean AP
Our method	**0.891**	0.900	0.812	0.852	0.925	0.876
Wang *et al.* [7]	0.866	**0.975**	**0.832**	0.843	0.828	0.869
Ma *et al.* [16]	0.881	0.920	0.756	0.868	**0.959**	**0.877**
Srinivasan *et al.* [35]	0.845	0.916	0.787	**0.888**	0.922	0.872
Maji *et al.* [41]	0.869	0.724	0.742	0.806	0.716	0.771
Felz *et al.* [3]	0.891	0.95	0.608	0.721	0.391	0.712
Lu *et al.*[15]	0.844	0.641	0.617	0.643	0.798	0.709

Table 2: Comparison of detection rates for 0.3/0.4 FPPI on the ETHZ dataset.

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Our method	**0.95/0.95**	0.964/0.964	0.915/0.915	**0.968/0.968**	**1/1**	**0.959/0.959**
Wang *et al.* [7]	0.90/0.90	**1/1**	**0.92/0.92**	0.94/0.94	0.94/0.94	0.940/0.940
Srinivasan *et al.* [35]	0.95/0.95	**1/1**	0.872/0.896	0.936/0.936	**1/1**	0.952/0.956
Maji *et al.* [41]	0.95/0.95	0.929/0.964	0.896/0.896	0.936/0.967	0.882/0.882	0.919/0.932
Felz *et al.* [3]	0.95/0.95	**1/1**	0.729/0.729	0.839/0.839	0.588/0.647	0.821/0.833
Lu *et al.* [15]	0.9/0.9	0.792/0.792	0.734/0.77	0.813/0.833	0.938/0.938	0.836/0.851
Riemenschneider *et al.* [6]	0.9333/0.9333	0.970/0.970	0.792/0.819	0.846/0.863	0.926/0.926	0.893/0.905
Ferrari *et al.* [5]	0.777/0.832	0.798/0.816	0.399/0.445	0.751/0.8	0.632/0.705	0.671/0.72
Zhu *et al.* [14]	0.800/0.800	0.929/0.929	0.681/0.681	0.645/0.742	0.824/0.823	0.776/0.795

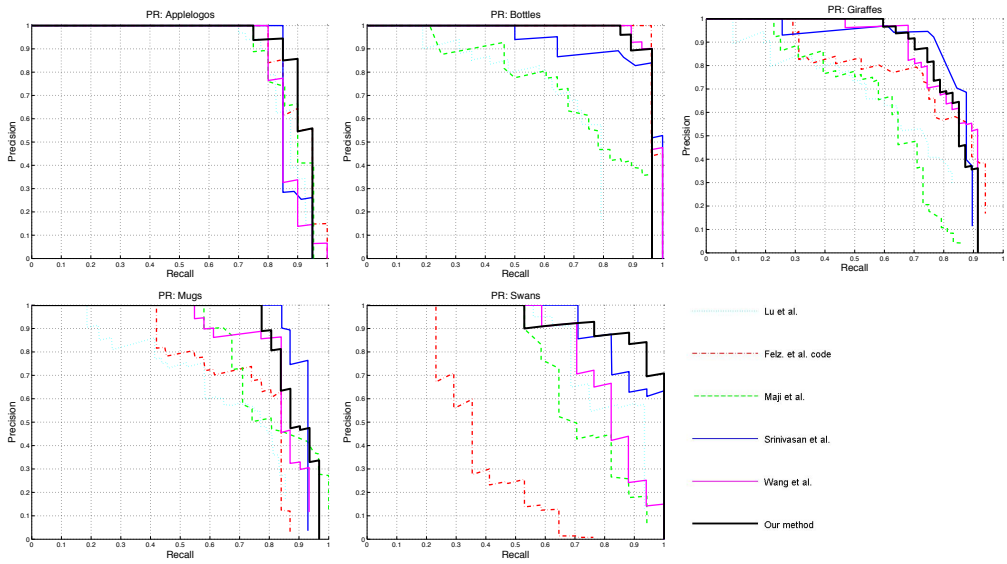

Figure 6: Precision/Recall curves of our method in comparison with existing methods on the ETHZ shape classes.

5.2. *Experiment on INRIA-Horse dataset*

The INRIA-Horse dataset consist of 170 images with one or more horses and 170 images without horses, which is more challenging than the ETHZ shape

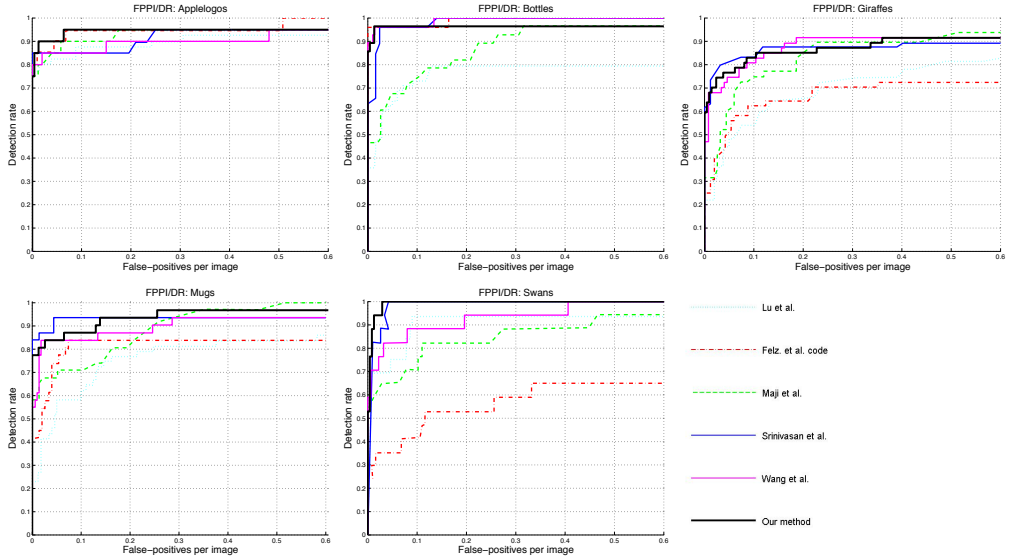

Figure 7: DR/FPPI curves of our method in comparison with existing methods on the ETHZ shape classes.

dataset. Horse appear in images at several scales, and against occlusion and cluttered background. Fig. 10 reports the plots of detection rate (DR) vs. false positive per image (FPPI). It shows that our algorithm slightly outperformed the recent methods: We achieved a detection rate of 88.6% at 1.0 FPPI; the reported results of competing algorithms are: 87.3% in [42], 85.27% in [41], 80.77% in [43], and 73.75% in [5]. Some of our detection results are exhibited in Fig. 11, where false positive samples are highlighted with an orange frame.

6. Further Discussions and Conclusion

We investigated the problem of detecting deformable objects from cluttered images given a single sketch as model. We developed constellational contour parsing for contour-based object detection, which found the the optimal configu-

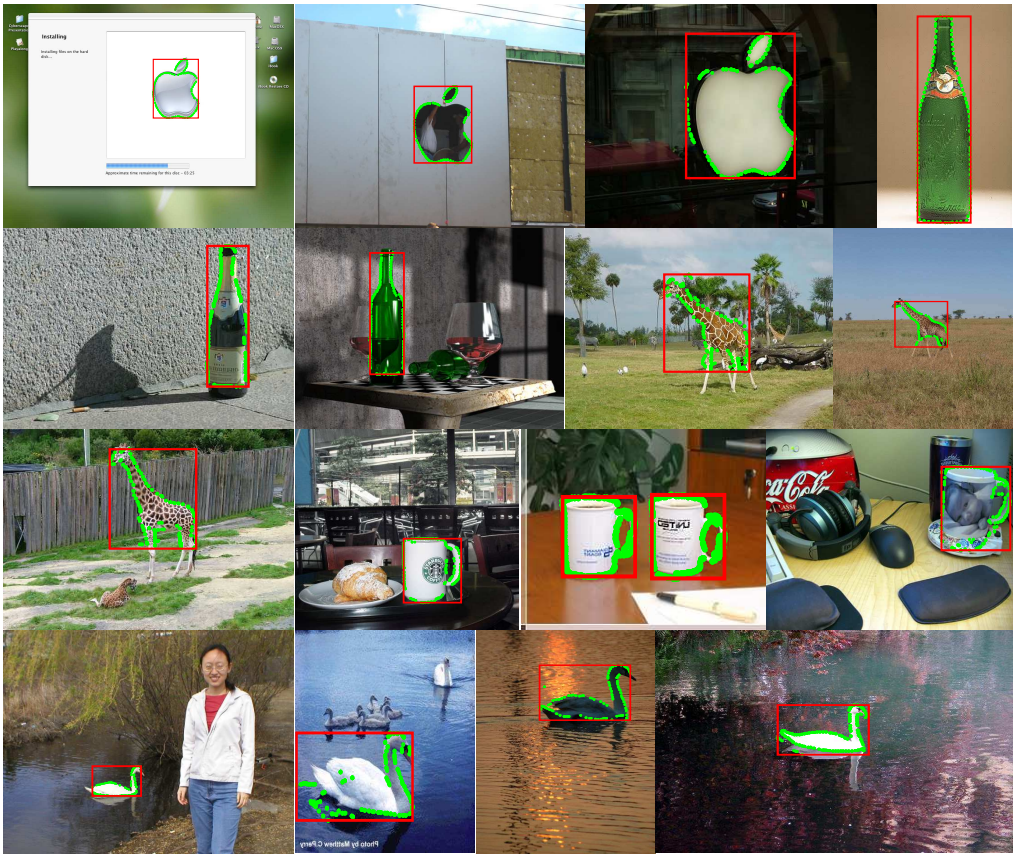

Figure 8: Samples of object detection results on ETHZ dataset

ration of contour segments in the clutter image to match the model contour. We introduced the scheme of anchor points and observation rays and developed local shape descriptors and an additive similarity metric function, so as to transform the original combinatorial optimization problem of contour grouping and matching into a dynamic programming problem.

To effectively detect objects with large deformation, we augmented the metric function with local motion search, modeled the relationship between different shape parts using multiple concurrent dynamic programming shape parsers. Our

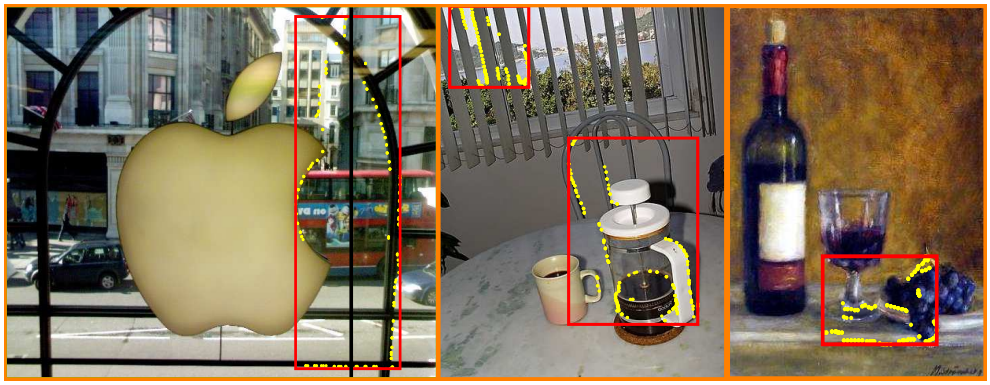

Figure 9: Samples of false positive detections on ETHZ dataset

experimental results showed that the proposed method to be very competitive with existing methods on the benchmark dataset and outperformed them in many cases.

6.1. Further Discussion

Dynamic programming has been used in shape analysis to find the correspondence between two shapes [7]. In this work, we used dynamic programming to find a subset of contour segments, more specifically, a constellation of contour segments, to match the model shape.

Our work is related to the contour segment network method in [12], which finds a path in the contour segment network to match the target shape. This network method connects neighboring contour segments into a graph based on local connections. In our work, we use observation rays to scan the contours and constructed a multi-stage dynamic programming trellis. In the test images, objects often have missing edges and contours. In this case, the network method, which relies on local connections, may fail to find the accurate matches. However, our CCP method can still correctly operate over this gap of missing contours since it examines the overall constellation and does not relies on local connections.

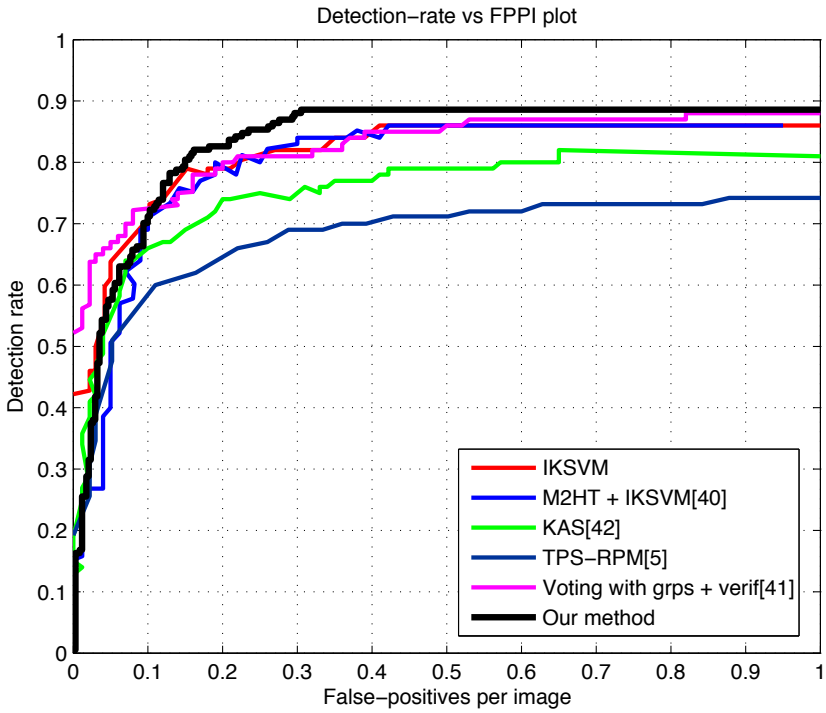

Figure 10: DR/FPPI curves of our method in comparison with existing methods on the INRIA-Horse dataset.

6.2. *Major Contributions*

The major contribution of this work lies in the following three aspects. (1) We introduced the scheme of constellational contour parsing (CCP) with anchor points, observation rays, and additive shape metric function to convert a combinatorial contour grouping problem into a dynamic programming problem with low complexity. (2) We developed an implicit shape parts analysis scheme using multiple observation points to handle object deformation. (3) The proposed method is able to detect objects with different scales, rotations, perspective changes, and de-

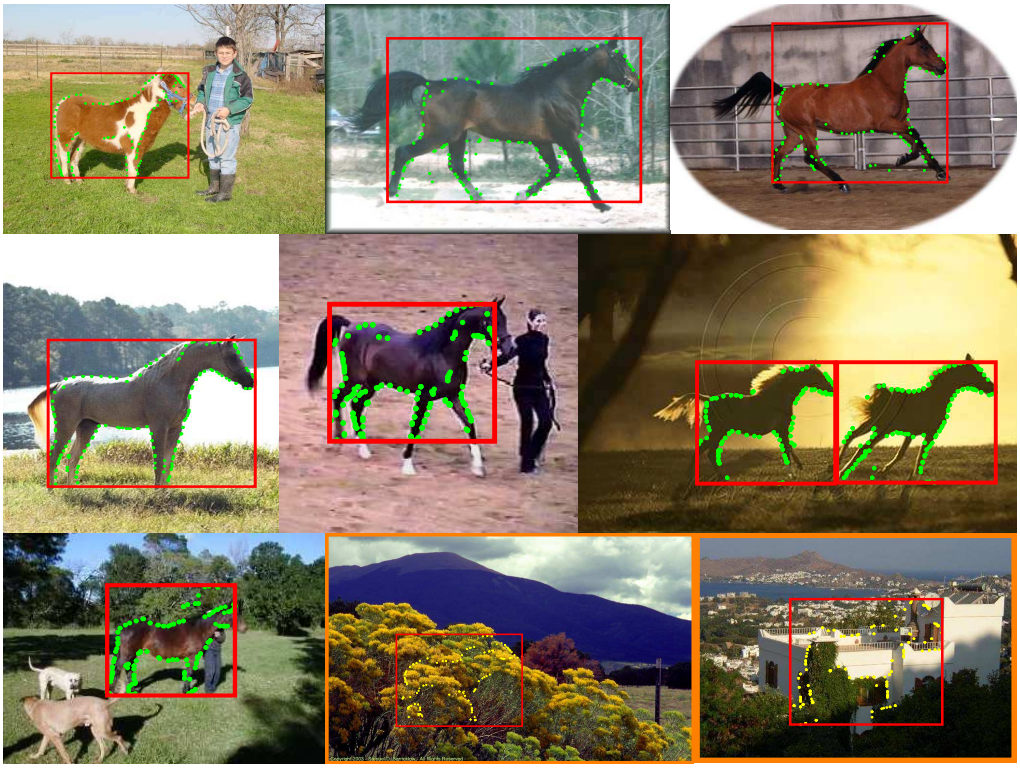

Figure 11: A few representative shape detection results on INRIA-Horse dataset. Two false positives are labeled by bold orange frames.

formations at low computational complexity and achieves very encouraging and highly competitive results.

References

[1] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, IEEE, 2001, pp. I–511.

[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, IEEE, 2005, pp. 886–893.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (2010) 1627–1645.

[4] X. Wang, M. Yang, S. Zhu, Y. Lin, Regionlets for generic object detection, in: ICCV'13: Proc. IEEE 14th International Conf. on Computer Vision, 2013.

[5] V. Ferrari, F. Jurie, C. Schmid, Accurate object detection with deformable shape models learnt from images, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[6] H. Riemenschneider, M. Donoser, H. Bischof, Using partial edge contour matches for efficient object category localization, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 29–42.

[7] X. Wang, X. Bai, T. Ma, W. Liu, L. J. Latecki, Fan shape model for object detection, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 151–158.

[8] X. Wang, L. Lin, Dynamical and-or graph learning for object shape modeling and detection., in: NIPS, 2012, pp. 242–250.

[9] J. Shotton, A. Blake, R. Cipolla, Contour-based learning for object detection, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 1, IEEE, 2005, pp. 503–510.

[10] A. Opelt, A. Pinz, A. Zisserman, A boundary-fragment-model for object detection, in: Computer Vision–ECCV 2006, Springer, 2006, pp. 575–588.

[11] S. Ravishankar, A. Jain, A. Mittal, Multi-stage contour based detection of deformable objects, in: Computer Vision–ECCV 2008, Springer, 2008, pp. 483–496.

[12] V. Ferrari, T. Tuytelaars, L. Van Gool, Object detection by contour segment networks, in: Computer Vision–ECCV 2006, Springer, 2006, pp. 14–28.

[13] P. F. Felzenszwalb, J. D. Schwartz, Hierarchical matching of deformable shapes, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[14] Q. Zhu, L. Wang, Y. Wu, J. Shi, Contour context selection for object detection: A set-to-set contour matching approach, in: Computer Vision–ECCV 2008, Springer, 2008, pp. 774–787.

[15] C. Lu, L. J. Latecki, N. Adluru, X. Yang, H. Ling, Shape guided contour grouping with particle filters, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 2288–2295.

[16] T. Ma, L. J. Latecki, From partial shape matching through local deformation to robust global shape similarity for object detection, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1441–1448.

[17] J. Canny, A computational approach to edge detection, Pattern Analysis and Machine Intelligence, IEEE Transactions on (1986) 679–698.

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

[19] P. D. Kovesi, MATLAB and Octave functions for computer vision and image processing, Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, ????

[20] T. B. Sebastian, P. N. Klein, B. B. Kimia, Recognition of shapes by editing their shock graphs, Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (2004) 550–571.

[21] X. Bai, Q. Li, L. J. Latecki, W. Liu, Z. Tu, Shape band: A deformable object detection approach, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1335–1342.

[22] I. Biederman, G. Ju, Surface versus edge-based determinants of visual recognition, Cognitive psychology 20 (1988) 38–64.

[23] A. Toshev, B. Taskar, K. Daniilidis, Shape-based object detection via boundary structure segmentation, International journal of computer vision 99 (2012) 123–146.

[24] S. Bagon, O. Brostovski, M. Galun, M. Irani, Detecting and sketching the common, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 33–40.

[25] M. Liu, B. C. Vemuri, S.-I. Amari, F. Nielsen, Shape retrieval using hierarchical total bregman soft clustering, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (2012) 2407–2419.

[26] K. Siddiqi, S. M. Pizer, Medial representations: mathematics, algorithms and applications, volume 37, Springer, 2008.

[27] M. Leyton, A process-grammar for shape, Artificial Intelligence 34 (1988) 213–247.

[28] P. J. Flynn, A. K. Jain, Cad-based computer vision: from cad models to relational graphs, in: Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on, IEEE, 1989, pp. 162–167.

[29] J. Damski, J. S. Gero, A logic-based framework for shape representation, Computer-Aided Design 28 (1996) 169–181.

[30] C. Huang, T. X. Han, Z. He, Multi-scale embedded descriptor for shape classification, Journal of Visual Communication and Image Representation (2014).

[31] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, Pattern Analysis and Machine Intelligence, IEEE Transactions on 24 (2002) 509–522.

[32] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[33] R. Hu, J. Collomosse, A performance evaluation of gradient field hog descriptor for sketch based image retrieval, Computer Vision and Image Understanding 117 (2013) 790–806.

[34] G. Borgefors, Hierarchical chamfer matching: A parametric edge matching algorithm, Pattern Analysis and Machine Intelligence, IEEE Transactions on 10 (1988) 849–865.

[35] P. Srinivasan, Q. Zhu, J. Shi, Many-to-one contour matching for describing and discriminating object shape, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1673–1680.

[36] P. F. Felzenszwalb, R. Zabih, Dynamic programming and graph algorithms in computer vision, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (2011) 721–740.

[37] B. Ommer, J. Malik, Multi-scale object detection by clustering lines, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 484–491.

[38] F. Jurie, C. Schmid, Scale-invariant shape features for recognition of object categories, in: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, IEEE, 2004, pp. II–90.

[39] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2010) 303–338.

[40] V. Ferrari, F. Jurie, C. Schmid, From images to shape models for object detection, International Journal of Computer Vision 87 (2010) 284–303.

[41] S. Maji, J. Malik, Object detection using a max-margin hough transform, in:

Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1038–1045.

[42] P. Yarlagadda, A. Monroy, B. Ommer, Voting by grouping dependent parts, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 197–210.

[43] V. Ferrari, L. Fevrier, F. Jurie, C. Schmid, Groups of adjacent contour segments for object detection, Pattern Analysis and Machine Intelligence, IEEE Transactions on 30 (2008) 36–51.

Highlights

- We proposed a novel framework for contour-based object detection from cluttered environment
- We simultaneously perform selecting of relevant contour fragments, grouping of the selected contour fragments, and finding best geometry-preserving matching to model contours
- We developed local shape descriptors and an additive similarity metric function which can be computed locally while preserving the capacity of matching deformable shape globally
- We augmented the metric function with a local motion search, modeled the relationship between different shape parts using multiple concurrent dynamic programming shape parsers
- The proposed method outperforms the state-of-the-art contour-based object detection algorithms on two benchmark datasets in terms of average precision