Object Detection Using Principal Contour Fragments

Changhai Xu
Department of Computer Science
University of Texas at Austin
1 University Station, Austin, TX 78712
changhai@cs.utexas.edu

Benjamin Kuipers
Computer Science and Engineering
University of Michigan
2260 Hayward Street, Ann Arbor, MI 48109
kuipers@umich.edu

***Abstract*—Contour features play an important role in object recognition. Psychological experiments have shown that maximum-curvature points are most distinctive along a contour [6]. This paper presents an object detection method based on *Principal Contour Fragments (PCFs)*, where PCFs are extracted by partitioning connected edge pixels at maximum-curvature points. An object is represented by a set of PCFs and their mutual geometric relations. The mutual geometric relations are described in each PCF's local coordinate system, and they are invariant to translation, rotation, and scale.**

With this representation, given any individual PCF, the system is capable of predicting all other PCFs' geometric properties. Object instances are detected in test images by sequentially locating PCFs whose geometric properties best match their predictions. Detected objects are verified according to their similarity to the model based on both individual PCF descriptors and mutual relation descriptors. Evaluation results show that the system works well in the presence of background clutter, large scale changes, and intra-class shape variations.

***Keywords*-object detection; shape matching; contour matching; edge matching.**

I. Introduction

To identify and localize objects, various visual cues can be exploited such as brightness, color, texture, and contour. Contour features are relatively robust to illumination changes and variations in color and texture. Psychophysical experiments [3] show that humans can identify a simple line drawing as quickly and as accurately as a fully detailed, textured, colored photographic image of the same object.

This paper focuses on localizing object instances in test images based on contour cues, given a single hand-drawn shape exemplar.

Contour fragments are commonly constructed based on edge maps. As is well known, edge detection may give brittle results for cluttered images. Some important edges may be missing, and a single contour can be broken into a few pieces. Previous works [9], [15] use mostly short contour fragments to build models for object detection. In our work, we construct more salient *Principal Contour Fragments (PCFs)*, by first linking edges having position and curvature continuity and then partitioning them at maximum-curvature points. Maximum-curvature points are shown to be most distinctive along a contour [6], hence provide robustness to our PCF-based model.

We represent an object by a set of PCFs and their mutual geometric relations. The model is described by a fully connected graph with the PCFs as nodes and their mutual relations as edges. The mutual geometric relations are described in each PCF's local coordinate system. This local coordinate system is independent of the original image coordinate system, and provides the geometric relations with invariance to translation, rotation, and scale. With this PCF-based object model, the system is capable of predicting a PCF's geometric properties based on any other PCF.

In the object detection stage, corresponding PCFs are discovered sequentially in test images to match the object model. Each time we predict the geometric properties of the next PCF to be detected, based on previously detected PCFs and the relations between the to-be-detected PCF and previously detected PCFs. Then we search for the one that best matches the prediction. The detection order is determined by calculating a detection priority score for each of the remaining PCFs in the object model. After an object instance is detected, a verification step is taken and the object instance is assigned a confidence score representing the similarity between the object instance and the object model.

In the PCF-based model, each individual PCF is assigned a weight indicating their relative importance. Each PCF relation is also assigned a weight indicating how closely the two corresponding PCFs are related. These weights provide a way to set PCFs' priority in the process of object detection. At any time, the PCF with the highest priority is selected as the next to-be-detected PCF. Note that the next to-be-detected PCF is not necessarily a neighbor of the last detected PCF, which is different from general contour tracing such as in [10].

The weight for a PCF relation depends on both the overall distance between the two corresponding PCFs and the local connection between the PCFs' ending points. Thus the PCF-based model encodes information for both global distribution of PCFs and local continuity between adjacent PCFs.

Since we detect object instances by locating corresponding PCFs, the detection results produce the actual object contours, not just bounding boxes.

The rest of the paper is organized as follows. Section II

reviews related work. The object representation is described in Section III. Section IV presents the object detection approach. Evaluation results are demonstrated in Section V. Section VI discusses future work and concludes.

II. Related Work

Several methods such as Shape Contexts [2] and Shock Graphs [13] measure similarity between object shapes, but they assume the objects are pre-segmented. Chamfer matching can be used to detect shapes in cluttered images, but it usually requires multiple templates to handle intra-class shape variations [16].

Extracting contour features typically requires a pre-processing step for edge detection. The Canny edge detector [4] and the Berkeley boundary detector [12] are two popular methods. It has been shown that the Berkeley boundary detector performs better in contour-based object detection compared with the Canny edge detector [15], thus we choose the Berkeley boundary detector for contour detection in our system.

Shotton *et al.* [14], [15] built a class-specific codebook of uncluttered contour fragments from noisy data where local contour fragments are spatially arranged in a star configuration. This method detects objects across various locations and scales, but it is not clear how well it can identify the actual object boundaries. In addition, a star-based spatial configuration may not be suitable for object shapes that have large imbalance. Ferrari *et al.* [10], [9] constructed a codebook of pairs of adjacent contour fragments and learned class-specific shape models. Fergus *et al.* [8] incorporated high-level meaningful contour fragments and learned a constellation part-based model for object detection. While many training images are required in [14], [15], [8], [9], our system needs only a shape exemplar as input (similar to [10]). Compared to [10], [9], [14], [15] which use mostly short contour fragments to represent objects, we extract more meaningful and salient Principal Contour Fragments to achieve robust detection results (similar to [8]).

In [17], objects are detected based on a single shape exemplar where the shape exemplar is manually partitioned into a few semantic parts. Our system also takes a hand-drawn exemplar as input, but it can automatically partitions the exemplar into meaningful PCFs. In [1] shape variations are handled as a Shape Band model, which exploits the boundary edges only. Our approach can work on either the boundary edges, or internal edges, or both.

Among all the above works, our work is most similar to [10] which locates contour fragments one by one along the object outline. Our work differs from [10] in that it can take advantage of patterns consisting of unconnected contour fragments, because geometric relations are considered between every pair of PCFs regardless of their adjacency. Also, in [10] only roughly straight contour fragments are used where the breaking points on a contour fragment may lie on indistinctive positions. We instead use more general fragments which can be straight fragments or curved ones, where the breaking points lie robustly on positions with maximum curvature.

III. Object Representation

An object is modeled by a set of PCFs, $\{f_i\}$ $(i = 1, ..., N)$, and their mutual geometric relations, $\{r_{ij}\}$ $(i, j = 1, ..., N, and\ i \neq j)$. Each f_i has a weight w_i, and each r_{ij} has a weight w_{ij}, indicating their respective importance. These weights determine each PCF's priority and importance in object detection and verification.

The object model can be depicted in a fully connected graph, with the PCFs as nodes and their relations as edges (Fig. 1).

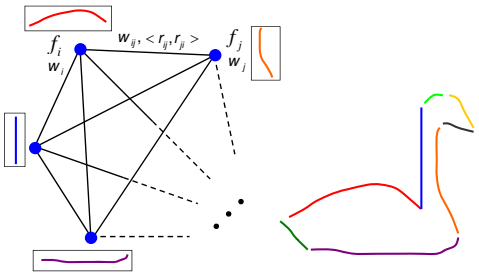

Figure 1. Object representation. The right figure shows extracted PCFs in different colors for a swan. The left figure is an illustration of the PCF-based object model where all the PCFs are related to each other. An object model includes two classes of descriptors: descriptors for individual PCFs and descriptors for PCF relations.

The PCFs are extracted from the input hand-drawn shape exemplar by dividing connected edge pixels at maximum-curvature points. This PCF extraction process in the shape exemplar is the same as in test images, and is done without user intervention (See next section for details).

A PCF is described by

$$f_i = \{P_i^c, P_i^s, P_i^e, \theta_i, l_i, \gamma_i, \alpha_i\} \tag{1}$$

where $P_i^c = (x_i^c, y_i^c)$ is the PCF's centroid, $P_i^s = (x_i^s, y_i^s)$ is its start point, $P_i^e = (x_i^e, y_i^e)$ is its end point, θ_i is its orientation from P_i^s to P_i^e, l_i is its integral length, γ_i is its bending ratio defined as $|P_i^e - P_i^s|$ divided by l_i, and α_i measures its bending strength/direction and is defined as the distance from P_i^c to the line $\overrightarrow{P_i^s P_i^e}$ divided by l_i, with a positive sign if P_i^c is on the left side of $\overrightarrow{P_i^s P_i^e}$ and a negative sign otherwise.

The importance weight for a PCF f_i is proportional to its length and is defined as

$$w_i = l_i / \sum_k l_k \tag{2}$$

where longer PCFs are given more weights than shorter ones.

A relation r_{ij} models f_j's geometric attributes (orientation θ_{ij}, length l_{ij}, centroid P^c_{ij}, start point P^s_{ij}, and end point P^e_{ij}) in f_i's *local coordinate system* (Fig. 2) and is described by

$$r_{ij} = \{\bar{\theta}_{ij}, \sigma^{\theta}_{ij}, \bar{l}_{ij}, \sigma^{l}_{ij}, \bar{P}^c_{ij}, \sigma^c_{ij}, \bar{P}^s_{ij}, \sigma^s_{ij}, \bar{P}^e_{ij}, \sigma^e_{ij}\} \quad (3)$$

where $\bar{\theta}_{ij} \in [-\pi, \pi)$, $\bar{P}^h_{ij} = (\bar{x}^h_{ij}, \bar{y}^h_{ij})$ $(h \in \{c, s, e\})$.

This descriptor contains both expected values and deviations for geometric attributes. The confidence for each geometric attribute is evaluated by

$$\begin{aligned} g(\theta_{ij}) &= \exp(-\frac{(T(\theta_{ij} - \bar{\theta}_{ij}))^2}{(\sigma^{\theta}_{ij})^2}), \\ g(l_{ij}) &= \exp(-\frac{(l_{ij} - \bar{l}_{ij})^2}{(\sigma^{l}_{ij})^2}), \\ g(P^h_{ij}) &= \exp(-\frac{(x^h_{ij} - \bar{x}^h_{ij})^2 + (y^h_{ij} - \bar{y}^h_{ij})^2}{(\sigma^h_{ij})^2}) \end{aligned} \quad (4)$$

where $g(\cdot) \in (0, 1]$ and $T(.)$ is a function that normalizes its argument to $[-\pi, \pi)$.

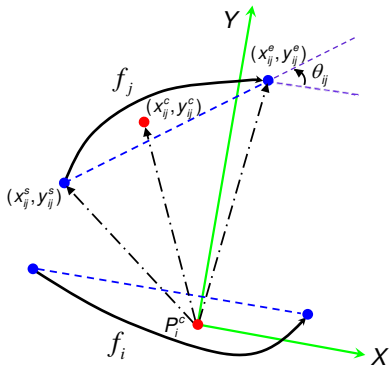

Figure 2. Illustration of the PCF relation r_{ij}. The relation is represented in f_i's local coordinate system. The origin is at f_i's centroid P^c_i, the x-axis is aligned with the direction from f_i's start point to its end point, and f_i's length is normalized to 1.

This representation provides the system the flexibility of PCFs' geometric distribution, therefore allows the system to be capable of accommodating shape deformations. The parameters in Eq. 3 are defined as

$$\begin{aligned} \bar{\theta}_{ij} &= \theta_j - \theta_i, \\ \bar{l}_{ij} &= l_j / l_i, \\ \bar{x}^h_{ij} &= \sqrt{(x^h_j - x^c_i)^2 + (y^h_j - y^c_i)^2}\cos(\theta_j - \theta_i)/l_i, \\ \bar{y}^h_{ij} &= \sqrt{(x^h_j - x^c_i)^2 + (y^h_j - y^c_i)^2}\sin(\theta_j - \theta_i)/l_i \end{aligned} \quad (5)$$

and

$$\begin{aligned} \sigma^{\theta}_{ij} &= \pi/6, \\ \sigma^{l}_{ij} &= 0.5 l_{ij}, \\ \sigma^h_{ij} &= \min\{|P^h_j - P^c_i|, |P^h_j - P^s_i|, |P^h_j - P^e_i|\}/l_i \end{aligned} \quad (6)$$

where $h \in \{c, s, e\}$. The parameters σ^{θ}_{ij}, σ^{l}_{ij}, and σ^h_{ij} are empirically selected by allowing the object shape to deform in a certain range (ideally they should be learned from real images), and kept fixed in this work. Note that $\bar{l}_{ij}$, $\bar{x}^h_{ij}$, $\bar{y}^h_{ij}$, σ^{l}_{ij}, and σ^h_{ij} have been scaled according to f_i's length.

Since we define the relation r_{ij} in f_i's local coordinate system, r_{ij} is completely independent of the original image coordinate system.

The importance weight for r_{ij} is defined as

$$\begin{aligned} w_{ij} &= 0.5\exp(-\frac{|P^c_j - P^c_i|}{L}) \\ &+ 0.5\exp(-\frac{\min_{h^1,h^2}|P^{h^1}_j - P^{h^2}_i|}{0.05L}) \end{aligned} \quad (7)$$

where $h^1, h^2 \in \{c, s, e\}$. The first part in the right-hand side measures the global connection strength between the two PCFs, and the second part measures the local connection strength. Intuitively closer PCFs are assigned higher weights. Here $L = (l_i + l_j)/2$. Note that in our definitions, w_{ij} and w_{ji} are always the same, but r_{ij} may be different from r_{ji}.

In this PCF-based object representation, each PCF is related to all others in its local coordinate system, which provides our system the ability to predict all other PCFs' geometric properties from any individual PCF.

IV. Object Detection

The goal of the system is to locate potential object instances and return corresponding confidence scores, given a hand-drawn object shape exemplar. We first extract PCFs in the input shape exemplar as well as in test images. Then the object model is created based on the extracted PCFs in the shape exemplar. Then we search for PCF subsets as object instances in test images to match the object model. Each detected object instance is then verified based on the object model.

A. Extract Principal Contour Fragments

To extract PCFs, we first trace connected edge pixels to form initial fragments, then link initial fragments into longer ones according to their position and curvature continuity, and then partition the resulting fragments at maximum-curvature points.

1) Extract initial contour fragments: Initial contour fragments can be formed by tracing neighboring edge pixels in the extracted edge map of an input image. In our experiments, we used the ETHZ Shape Classes dataset [10], [9] and copied the edge detection results from the dataset which uses the Berkeley boundary detector [12]. Very small fragments are removed since they are usually not salient features.

2) Link contour fragments: Edge detection results can be very brittle, and an ideal contour may be broken into a few pieces. Among the initial contour fragments, we repeatedly link fragment pairs that have position and curvature continuity.

Let f_a and f_b be two fragments where f_b's start point is close to f_a's end point. We sample three points $\{a_0, a_1, a_2\}$ with equal space from f_a's end part, and another three points $\{b_0, b_1, b_2\}$ from f_b's start part.

We first fit a circle for $\{a_0, a_1, a_2\}$ and measure the distance from each point in $\{b_0, b_1, b_2\}$ to the circle. Then we fit another circle for $\{b_0, b_1, b_2\}$ and measure the distance from each point in $\{a_0, a_1, a_2\}$ to the new circle. If these distances are lower than some threshold, the two fragments are linked to form a new one. This process repeats until no two fragments can be linked any further.

3) Partition contour fragments: For each resulting fragment obtained from the above step, we calculate the curvature for every point on the fragment. Local maximum-curvature points are then identified [11]. Among these maximum-curvature points, those whose curvatures exceed a threshold are used to partition this fragment into a list of PCFs. In test images, besides each originally extracted PCF, we also store a reversed copy where the original PCF and the reversed one have the same set of points but opposite directions.

B. Build the Object Model

The extracted PCFs from the shape exemplar contribute the set of PCFs $\{f_i\}$ $(i = 1, ..., N)$ in the object model. For each f_i, we calculate its orientation, length, centroid, start point, end point, bending ratio, and bending strength/direction. Then for each pair of PCFs, their geometric relation is identified according to Eq. 5. The weights for the PCFs and their relations are assigned according to Eq. 2 and Eq. 7.

C. Detect Object Instances

The task is to find PCF subsets from the set of PCFs $\{f_j^I\}$ $(j = 1, ..., M)$ in the test image, where each PCF subset has a good match with the PCFs $\{f_i\}$ $(i = 1, ..., N)$ in the object model. We will refer to each matching PCF subset as a candidate contour. The process of forming candidate contours is as follows.

1) Let $K = \{1, ..., N\}$ be the index set for $\{f_i\}(i = 1, ..., N)$. For each pair of f_i and f_j^I, we create a new candidate contour and f_j^I is added into the contour as $\tilde{f}_i$ (corresponding to f_i). An associated index set $\tilde{K}$ is also created to indicate which PCFs in the model have been detected, and i is added as the first element.
2) For each candidate contour, the index of the next to-be-detected PCF in the model is determined by

$$k = \arg\max_m \sum_i w_i w_{im}, \quad i \in \tilde{K},\ m \in K \backslash \tilde{K} \tag{8}$$

where all previously detected PCFs vote to decide which is the next PCF to be discovered. Intuitively the next to-be-detected PCF is the one that has the strongest connection with the previously detected PCFs.
3) The geometric attributes of the next to-be-detected PCF $\tilde{f}_k$ are predicted based on the object model and previously detected PCFs in the candidate contour. The final prediction is an weighted average of the predictions from all individual detected PCFs, where the weights are $w_i w_{ik}$ $(i \in \tilde{K})$.
4) Find the best PCF candidate in the test image, and add this PCF into the candidate contour. If the matching confidence is too low, then the corresponding PCF in the model is tagged as missing.
5) Repeat the above procedures until all candidate contours are formed.

D. Measure Detection Confidence

For each detected object instance, we measure its similarity confidence compared with the object model. We scale each object instance such that its total length is the same as the total length of corresponding PCFs in the object model. The similarity confidence is then evaluated as

$$\lambda = w^\lambda \lambda^f \lambda^r \tag{9}$$

where w^λ is a weight factor, λ^f is the similarity for individual PCFs, and λ^r is the similarity for PCF relations.

The weight factor w^λ is defined as the total weight of the successfully detected PCFs divided by the total weight of all PCFs in the model including the missing ones. The individual PCF similarity λ^f is calculated based on the length, bending ratio, and bending strength/direction for each PCF, which is measured in a similar way as in Eq. 4, where the deviations are a third of the corresponding length, bending ratio, and bending strength/direction respectively in the model. The PCF relation similarity λ^r is calculated according to Eq. 4.

E. Handle the Scale Problem

Our PCF-based model describes geometric relations in a PCF's local coordinate system, where the PCF's length is normalized to 1, thus the scale problem is intrinsically handled for similarity evaluation between an object instance and the object model. But the scale issue still exists in PCF extraction.

In PCF extraction, the key step is to identify the maximum-curvature points. The curvature at the kth point is calculated based on the $(k - \delta)$th and $(k + \delta)$th points. We choose δ at three different scales, $\delta = \{l/16, l/32, l/64\}$ where l is the contour fragment's length. Each input shape exemplar produces three object models that contain different PCFs. Each test image also produces three sets of different PCFs that correspond to different scales.

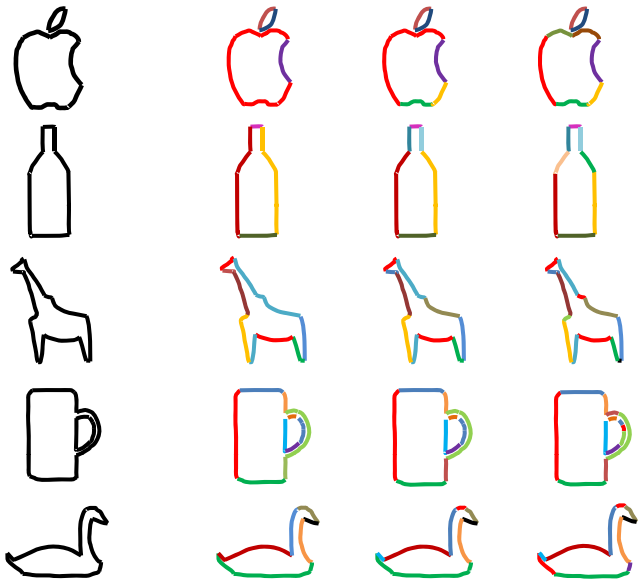

Figure 3. Input shape exemplars and constructed PCF-based object models. The left column shows the five input shape exemplars used in our experiments. The right three columns show the extracted PCFs in different colors for each exemplar. These PCFs are automatically extracted.

V. Evaluation

We evaluate the performance of the PCF-based object model on the ETHZ Shape Classes dataset [10], [9]. This dataset contains 5 object classes (apple logos, bottles, giraffes, mugs, and swans) over 255 images. It provides ground truth bounding boxes for object instances. The dataset is challenging due to background clutter, large scale changes, and intra-class shape variations.

The system is only given a single hand-drawn shape exemplar for each class. For each shape exemplar we automatically extract three object models (Fig. 3), which correspond to different scales in PCF extraction (please see Section IV-E for details). For edge detection, we use the results from the work in [10] which is based on the Berkeley boundary detector [12].

The detection results are shown in Fig. 4. Our system can locate actual object contours as well as their bounding boxes. In addition, it handles the scenario where there exist multiple object instances (only the best detection is shown in Fig. 4).

Corresponding to the detection results in Fig. 4, we also show the extracted PCFs for these images in Fig. 5. These PCFs are obtained by grouping neighboring edge elements into contour fragments and then partitioning them at maximum-curvature points.

We assess the system performance as the detection rate (DR) versus the number of false positives per image ($FPPI$). Following the criteria in [10], a detection is considered correct if the detected bounding box and the ground truth bounding box have an overlap rate over 20%. The overlap rate is defined as ratio of the intersection area over the union area of the two bounding boxes (between ground truth and actual detection).

We quantitatively evaluate our system performance by comparing the detection results with one of the state of the art works in [10]. We choose this work as our comparison baseline because both the work in [10] and ours use hand-drawn shape exemplars as input and do not require a set of training images. In addition, neither work needs to involve a post-processing step for shape refinement (such as in [9]).

The comparison results are shown in Fig. 6. Our method gives comparative overall performance. For apple logos, giraffes, mugs, and swans, our method gives comparative or better DR/FPPI rates. In particular, our method tends to give higher detection rate at low FPPIs. For bottles, our detection results are worse, mainly because there are many bottles in the test images that are significantly different from the hand-drawn shape exemplar.

VI. Conclusion and Future Work

We have presented a novel contour-based model for object detection. The model is represented by a set of Principal Contour Fragments (PCFs) and their mutual geometric relations. Objects are detected in test images by sequentially searching for the best matching PCFs based on the object model.

Partial occlusion is a challenging problem in object detection, which is not significantly demonstrated in the ETHZ dataset. The proposed PCF-based object model contains

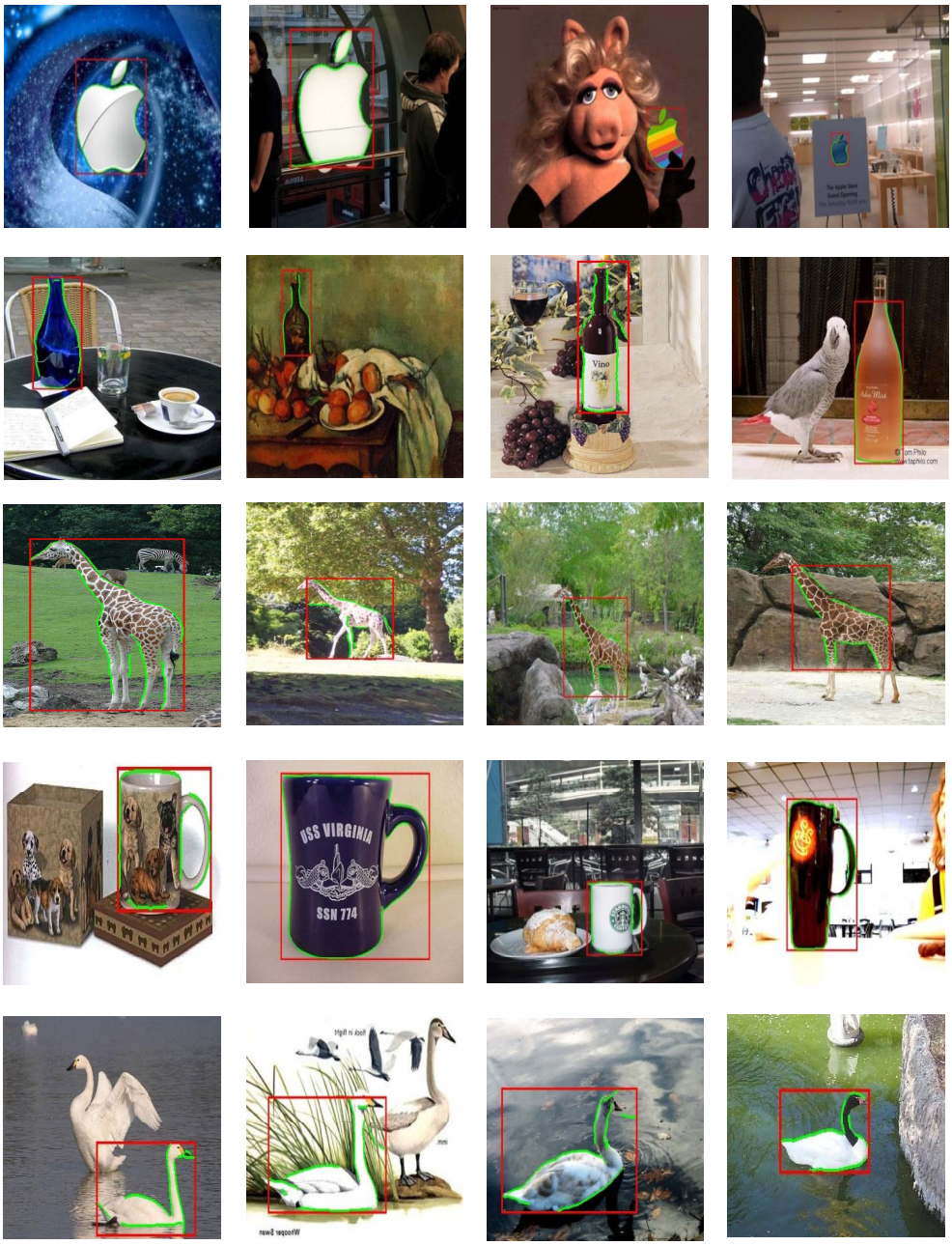

Figure 4. Detection results (best viewed in color). Objects are detected by locating a set of PCFs in the test image such that they match the PCFs in the object model. The bounding boxes of the detected objects are shown in red rectangles. Besides bounding boxes, the proposed method is also capable of locating the actual object contours which are shown in green color.

information of the geometric relation for every PCF pair, therefore we expect it to be capable of handling partial occlusion. We will test the model using datasets with partial occlusion in future work.

The object instances in the ETHZ dataset do not have significant orientation changes from the shape exemplars. The PCF-based model is expected to handle large orientation changes since it includes relative rather than absolute geometric attributes between pairs of PCFs. We will test the model using datasets with significant orientation changes.

Compared to the work in [10] which deals with object shapes consisting of connected contour fragments, our method can in principle handle both connected and unconnected fragments. Thus the PCF-based model can take advantage of internal contour fragments within the boundary (these fragments may be completely disconnected from the

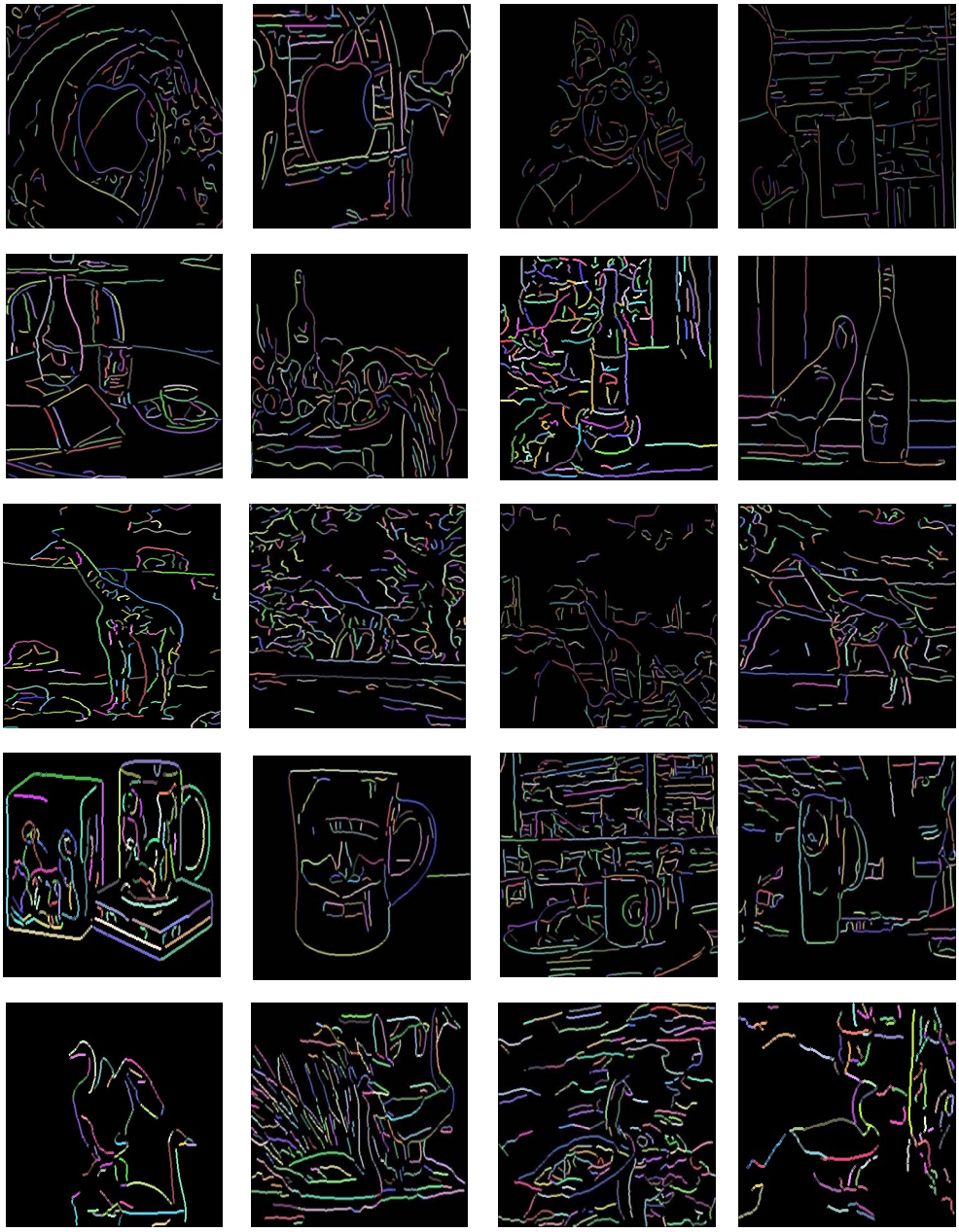

Figure 5. Extracted PCFs in test images (best viewed in color). PCFs are extracted by partitioning connected edge pixels at maximum-curvature points. Curvatures are calculated with dynamic intervals based on the corresponding contour fragment's length (see text for details). PCFs are shown in randomly generated colors (different PCFs may accidentally generate the same color).

boundary) as well as those that lie on the boundary. We will investigate how well it can detect patterns with unconnected fragments.

Currently all the relation parameters between PCFs are manually assigned and kept fixed in all the experiments. For different object categories such as apple logos and swans, their shape may vary to different extent, and fixed parameters can not meet this need. One of our future focuses will be to automatically learn these parameters from a set of training images.

We will also develop richer descriptors for both individual PCFs and PCF relations such as adding in orientations for a PCF at its end points.

Acknowledgment. This work has taken place in the Intelligent Robotics Labs at the University of Texas at Austin and at the University of Michigan. Research in the Intelligent Robotics Labs is supported in part by grants from the National Science Foundation (IIS-0713150 to UT Austin and

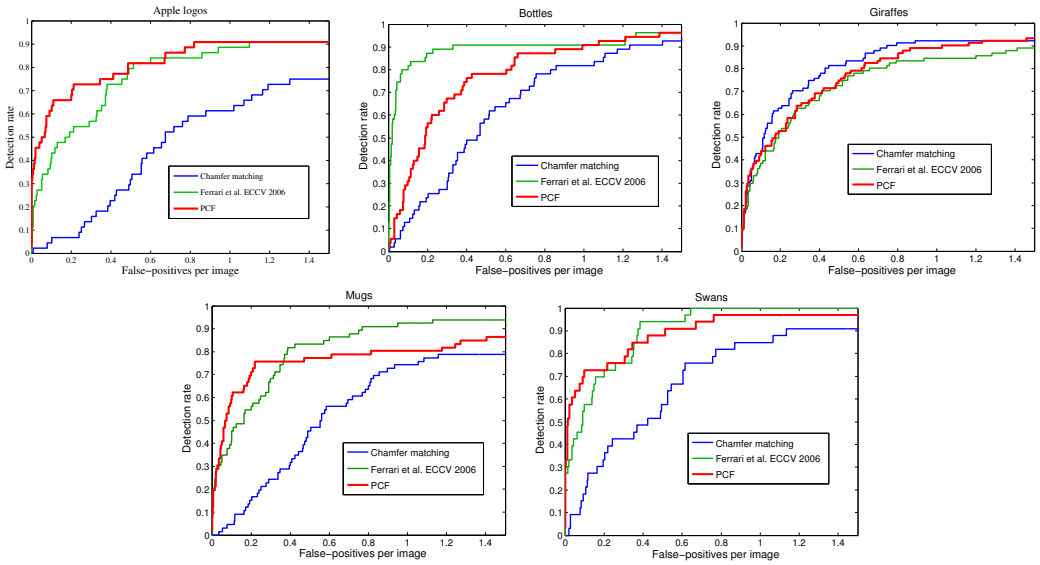

Figure 6. Comparison results (best viewed in color). We compare our work with the results reported in [10] which also includes Chamfer matching results. In overall, our method gives comparative results to Ferrari *et al.* ECCV 2006 work.

CPS-0931474 to UM) and from the TEMA-Toyota Technical Center to UM.

REFERENCES

[1] X. Bai, Q. Li, L. Latecki, W. Liu, and Z. Tu, "Shape band: A deformable object detection approach," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[2] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 509–522, 2002.

[3] I. Biederman and G. Ju, "Surface versus edge-based determinants of visual recognition," *Cognitive Psychology*, vol. 20, no. 1, pp. 38–64, 1988.

[4] J. Canny, "A computational approach to edge detection," *Readings in Computer Vision: Issues, Problems, Principles and Paradigms*, pp. 184–203, 1986.

[5] O. Danielsson, S. Carlsson, and J. Sullivan, "Automatic learning and extraction of multi-local features," *International Conference on Computer Vision*, 2009.

[6] J. De Winter and J. Wagemans, "Perceptual saliency of points along the contour of everyday objects: A large-scale study," *Perception and Psychophysics*, vol. 70, no. 1, p. 50, 2008.

[7] P. Felzenszwalb and J. Schwartz, "Hierarchical matching of deformable shapes," *Computer Vision and Patt. Recog.: CVPR2007*, pp. 1–8, 2007.

[8] R. Fergus, P. Perona, and A. Zisserman, "A visual category filter for google images," *European Conference on Computer Vision*, pp. 242–256, 2004.

[9] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection," *International Journal on Computer Vision*, 2009.

[10] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Object detection by contour segment networks," *Lecture Notes in Computer Science*, vol. 3953, p. 14, 2006.

[11] S. Hermann and R. Klette, "Global curvature estimation for corner detection," *Image and Vision Computing New Zealand*, 2005.

[12] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 530–549, 2004.

[13] T. Sebastian, P. Klein, and B. Kimia, "Recognition of shapes by editing shock graphs," in *IEEE International Conference on Computer Vision*, 2001, pp. 755–762.

[14] J. Shotton, A. Blake, and R. Cipolla, "Contour-based learning for object detection," in *International Conference on Computer Vision*, vol. 1, 2005, pp. 503–510.

[15] ——, "Multiscale categorical object recognition using contour fragments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1270–1281, 2008.

[16] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2003.

[17] Q. Zhu, L. Wang, Y. Wu, and J. Shi, "Contour context selection for object detection: A set-to-set contour matching approach," *European Conference on Computer Vision*, pp. 774–787, 2008.