Contour-based Object Detection

Joseph Schlecht
schlecht@uni-heidelberg.de
Björn Ommer
ommer@uni-heidelberg.de

Interdisciplinary Center for
Scientific Computing
University of Heidelberg
Germany

Abstract

The arrival of appearance-based image features has dramatically influenced the field of visual object recognition. Previous work has shown, however, that contour curvature and junctions are important for shape representation and detection. We investigate a local representation of contours for object detection that complements appearance-based information, such as texture. We present a non-parametric representation of contours, curvature, and junctions which enables their accurate localization. We combine contour and appearance information into a general, voting-based detection algorithm. Besides detecting objects, we demonstrate that this approach reveals the most relevant contours and junctions supporting each object hypothesis. The experiments confirm that our contour-based representation compliments appearance information and the performance of baseline voting methods is significantly improved.

1 Introduction

Contour-based representations have a long history in object recognition and computer vision. Considerable effort was spent in the past matching geometric shape models of objects to image contours [5, 6, 7, 8, 15, 22]. Although these approaches enjoyed some success, it is clear that finding contours exactly belonging to the shape of an object is a hard problem. This insight has given rise to an emphasis on local texture descriptors [13], the dominant approach today. These appearance-based descriptors summarize local texture information in the form of histograms of gradients [9], shape context [3], geometric blur [4], and many others [2, 16]. While prominent edges are roughly encoded, exact shape location has been replaced by a representation of texture. It is well known, however, that curvature of contours and junctions provide crucial shape information [1, 5]. Thus we believe it is time to investigate a contour representation alongside appearance-based descriptors.

In this paper we show that local contour descriptors produce significant gains in object detection and complement texture descriptors. Object contours are a strong representation of shape, whereas texture-based representations summarize contours to avoid matching them to an object exactly. We demonstrate in this work the value of bridging the assets of both. We propose a local contour representation that complements texture features by flexibly encoding junction information and curvature. The representation discretizes contour orientation at an interest point and records the intensity of the contour at each angle as feature elements.

Our local contour representation is a non-parametric distribution over oriented line segments, or bars, sampled densely along contours. Recent work has modeled a number of

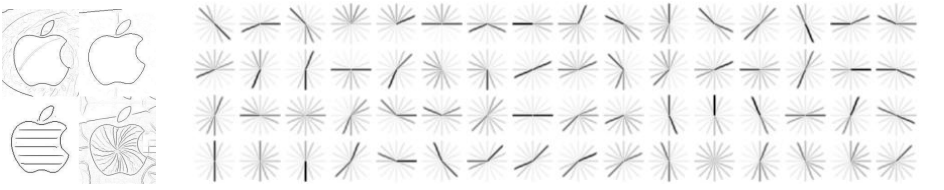


Figure 1: Oriented bar codebook learned from the Applelogos category of the ETHZ dataset. The representation summarizes local contour strength in a discrete set of 16 orientations. The oriented bar features were sampled uniformly along detected image contours (left) and clustered using kmeans. The example illustrates the repeatability of local contour orientation.

adjacent contour segments to represent shape [11]. The descriptor in this case has a fixed junction degree, utilizes long segments and requires them to be ordered. Other recent contour representations typically break contours into view-dependent segments and assemble them by voting or with constrained splines [12, 21, 24]. The length of the segments in these approaches pushes the descriptor outside the bounds of local information, precluding a simple association with local texture descriptors. We also make a distinction with appearance-based descriptors, such as HoG [4], that integrate over edge orientations in image regions and produce a texture summary that marginalizes contour detail. In contrast, our oriented bar descriptor defines a distribution over local curvature and junctions at points along contours.

The contour representation we introduce is highly repeatable throughout an object category and encodes local curvature information. Object shape contours contain local structure that repeats throughout standard views of a category. The corners of a mug, for example, and the junctions where a handle attaches, appear in most exemplars with some angular variation. In Figure 1 we visualize this notion of repeatability, which was inspired by [23]. For this example we computed oriented bar features over Applelogos of the ETHZ shape dataset and clustered them into 64 groups using kmeans. We observe that much of the object shape can be explained by a combination of these local contours.

In addition to being repeatable, our contour representation describes local curvature. Understanding angle information in contours is important because shape cues concentrate there; segments with high curvature encode more information about shape than straight lines [1, 5]. This is illustrated in Figure 2, where we sampled interest points uniformly over detected edges and computed oriented bar features. We then rendered the features in order of their contour strength and curvature. The latter is measured by the outer product of an oriented bar feature with itself, weighted by angle difference, *e.g.*, $\sum f_{ij} e^{-|\theta_i - \theta_j|}$. We observe that the object is identifiable after rendering a relatively small number of oriented bar features. Curvature clearly concentrates shape information, and the oriented bar descriptor captures this at contour interest points.

The advantages of a local contour representation complement those of texture descriptors and improve object detection. We demonstrate this by joining contour and texture information in a widely used, voting-based detection algorithm. We evaluate our approach on the ETHZ shape dataset and INRIA horses. Our results demonstrate that explicitly representing contour orientations alongside texture-based features, specifically geometric blur [4], significantly improves detection. Another contribution is the evaluation process itself. We average our performance over many random trials and report standard error statistics over mean performance, yielding a more meaningful comparison within this work and between others. We

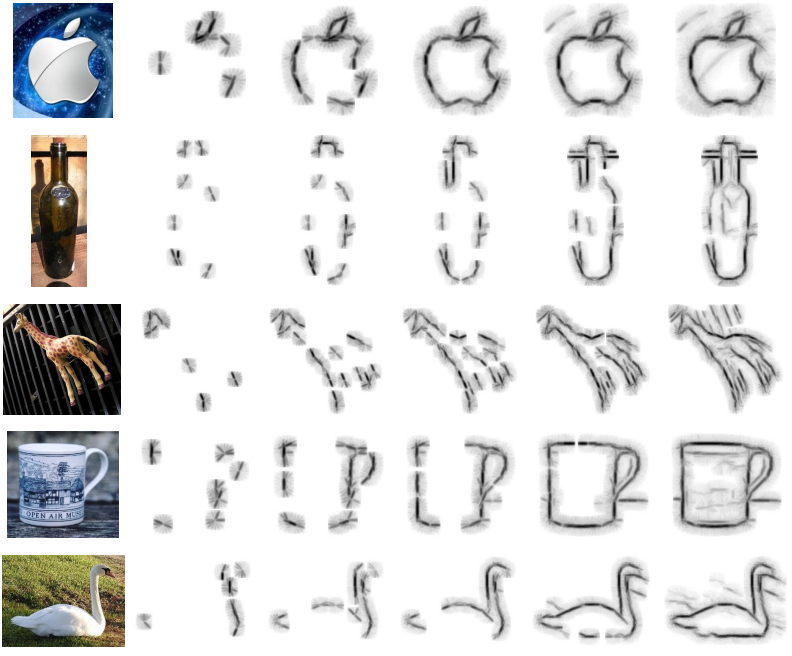


Figure 2: Shape information is concentrated at points of high curvature. The oriented bar features are ordered and rendered by contour strength and amount of curvature. The first column shows images from the ETHZ shape dataset. The second column shows 10 oriented bar features and continues to the right with 30, 50, 100 and 200 features. Notice that with just 10 oriented bar features most of the categories are discernible. This illustrates that our contour representation encodes curvature in a way that is highly informative of shape.

first describe our voting-based detection strategy and then summarize the contour and texture features we rely on. We finish with an empirical evaluation and discussion.

2 Detection

We first briefly outline the detection strategy for evaluating the combination of contour and texture information. We evaluate the proposed representation based on the widely used Hough transform for object detection. The approach is kept general to allow for wide applicability rather than specializing on a specific object category. The emphasis is on evaluating local contour properties, not the voting or verification stages. Nonetheless, we found that this simple detector, based on our object representation, is surprisingly effective; it exceeds many state-of-the-art voting-based detection systems. Figure 3 outlines the detection process.

To make a detection we first extract edge contours E from an image using the Berkeley edge detector [17, 19]. We sample N interest points locations $x_1, \ldots, x_N$ uniformly along contours. At each of the locations we compute our oriented bar contour representation $b_1, \ldots, b_N$ and the texture-based geometric blur [4] features $g_1, \ldots, g_N$. The details of these local image descriptors are in Sect 3. We then concatenate the descriptors at the i-th interest point into a feature vector $f_i = (b_i, g_i)$.

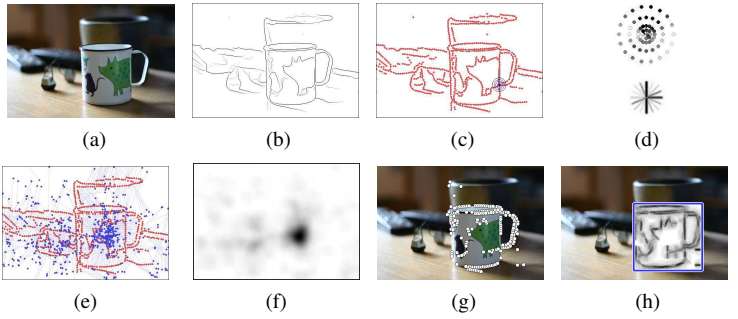


Figure 3: Outline of our approach to detect a Mug in image (a). We identify edge contours (b), uniformly sample interest points (c), and compute features (d). Example geometric blur features (top) and oriented bar representation (bottom) were computed where the handle attaches. Features are matched to training examples and shift vectors are extended through scale space (one scale shown) (e). A density over object hypotheses is estimated (f). Interest points belonging to a strong hypothesis are identified (g) as well as a scaled bounding box with rendered oriented bars (h).

For each of the features f_i in the test image, we find a nearest neighbor $\hat{f}_i$ in a training set. We use euclidean distance for matching and a kd-tree to make the retrieval efficient. We could quantize the features into a codebook, as in Figure 1, if matching performance becomes an issue. As part of the training process, we store a shift vector v_i that points to the object center for each of the training instances $\hat{f}_i$. Then for matches $f_1,\ldots,f_N$, we have a corresponding set of a shift vectors $v_1,\ldots,v_N$. We combine the v_i with interest point locations x_i to cast votes for the position of an object.

In addition to voting for object position, we cast each of the votes through a set of discrete scales. Formally, we vote for object hypotheses over position and scale space in a Hough accumulator

$$H(x,\sigma) = \sum_{i=1}^{N} w_i\, \delta(\|x_i + \sigma\, v_i - x\|)\,, \tag{1}$$

where δ is the Dirac delta function. This equation has been used numerous times and the weights have been estimated in many ways, including probability models [14], max-margin classifiers [18], and constrained co-activations [20]. Surprisingly, we found in our experiments that uniform weights exceeds the performance of most of these methods (excluding post-processing verification stages).

We estimate a discrete density over image position and scale space using a 3-D Gaussian

$$h(x,\sigma) = c \sum_{\mathsf{x},\mathsf{s}} H(x-\mathsf{x},\sigma-\mathsf{s})\,\eta(\mathsf{x},\mathsf{s};\omega_x,\omega_\sigma)\,. \tag{2}$$

This is a convolution of the Hough accumulator with a kernel function η that has ω_x^2 and ω_σ^2 on the diagonal of its covariance matrix. The constant c normalizes the discrete density. We can efficiently compute h using the fast Fourier transform. We identify object hypotheses as local maxima according to their 26-way neighborhood. A watershed segmentation over position in each scale of h gives the features casting votes for a hypothesis.

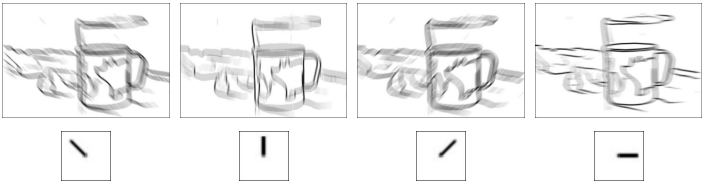


Figure 4: Image edge map convolved with four oriented bar filters. The oriented bar feature is computed by sampling from each channel at interest point locations x_i.

3 Object representation

In this section we present a representation for the complimentary characteristics of object, their contours and appearance. We first summarize our oriented bar descriptor for contours, followed by the geometric blur descriptor for texture.

3.1 Contours as oriented bars

To represent object contours, it is important to capture local curvature as well as junctions of various degrees and angles. Thus, we seek a non-parametric representation that describes contours at an interest point as a distribution over oriented line segments, or bars. The oriented bars are a set of filters $F_1, \ldots, F_D$ that have a line segment with one endpoint in the filter center and orientation angle $\theta \in [0, 2\pi]$. We create the filters by rendering a line of length L and blurring it with a Gaussian. This works to offset the discretization effects of orientation angles; a smoothed line will give a partial response to a contour that has almost the same angle.

The edge map E is convolved with each of the oriented bar filters, creating D channels of contour orientation responses

$$B_d(x) = \sum_{\mathrm{x}} E(x - \mathrm{x}) F_d(\mathrm{x}) \,. \tag{3}$$

Figure 4 shows four examples of the oriented bar filters and their responses to an edge map. Note that this is different from the oriented edge descriptor [19], since the filters are elongated bars, not gradients, and the signal is an edge map. Further, we are not integrating over gradient orientations in an image region, as is done with HoG [9].

We sample each channel at the interest points to create a feature vector

$$\hat{b}_i = B_1(x_i), \ldots, B_D(x_i) \,. \tag{4}$$

The feature is then normalized by its magnitude, $b_i = \hat{b}_i / \|\hat{b}_i\|$. We want the distance between local contour orientations to be similar regardless of the relative contour intensities.

We choose the number of orientations as $D = 16$ and a bar length of $L = 20$. The oriented bars in F_d are smoothed with a Gaussian that has a standard deviation of 2 pixels.

3.2 Texture as geometric blur

Besides contours, appearance information such as texture is key to representing objects. To describe local texture in the vicinity of interest points, we compute the geometric blur

descriptor [4]. This descriptor summarizes the response of a signal under all affine transformations at a point. It gives best results when the signal is sparse, such as an edge map. For a discrete signal E, the descriptor centered at location x is a convolution with a spatially varying Gaussian kernel,

$$G_x(y) = \sum_{\times} E(x+y-\times)\,\eta(\times;\alpha\|x-y\|+\beta)\,. \tag{5}$$

That is, the signal is convolved with a Gaussian kernel η whose standard deviation is a linear function of distance from the descriptor's center, x. In this work, we use the parameters $\alpha = 0.5$ and $\beta = 1$.

We sample G_x at locations $y_1,\ldots,y_C$ arranged in concentric circles about x. Each location is defined in polar coordinates by a radius r and angle θ. We use 7 radii ranging from 0 to 50 pixels. At each radius the points are sampled with equally spaced angles $\theta \in [0, 2\pi]$. This ranges in frequency from 1 point at $r = 0$, to 18 at the largest radius. For each signal, we sample a total of $C = 71$ points from the descriptor. An example can be seen in Figure 3(d).

Each of the point locations from the set of radii and angles are computed under the geometric blur descriptor and concatenated into a feature vector. The feature at interest point x_i is then given by

$$\hat{g}_i = G_i(y_1),\ldots,G_i(y_C)\,. \tag{6}$$

As with the oriented bar descriptor, we normalize the feature by its magnitude, $g_i = \hat{g}_i/\|\hat{g}_i\|$. This masks differences in pixel intensities due to lighting changes and other affects.

We compute the geometric blur feature over multiple sparse signals to increase its descriptiveness. We use 8 oriented edge responses [19] given by the Berkeley edge detector. We simply concatenate the features over each signal to produce an extended descriptor. The dimensionality is then reduced by half using PCA. Doing so maintains over 90% of the signal variation and yields a texture feature with 284 dimensions.

4 Experiments and results

We evaluate the proposed object representation in the context of multi-scale object detection in cluttered scenes. We perform a rigorous evaluation using the ETHZ shape dataset and INRIA horses. We create 10 random training and testing trials for each category within these datasets. Each of the ETHZ trials is constructed by randomly selecting half the target category from the data and an equal number of non-target images distributed evenly among the other categories. For INRIA each trial contains 50 positive and 50 negative examples, as is common. We separately evaluate our approach on the ETHZ and INRIA datasets against several comparable state-of-the-art approaches. We further show a comparison of our combined representation (OB+GB) against two baselines of only oriented bars (OB) or geometric blur (GB). This isolates any performance gains to the effects of our approach.

We configure the detector with the following options. The maximum number of interest points sampled in a test image N is 1000, with at least a 5 pixel spacing between each. We ignore contours in the edge map that have an intensity less than 25% of the maximum. In the Hough accumulator we choose a bin size of 5 pixels for position and divide the scales into 20 bins. The kernel bandwidth ω_x for position is 12 pixels, and the scale bandwidth ω_σ is set to 0.25. We found that weighting the contour and texture components of the combined feature equally gave the best results. Finally, as previously mentioned, we empirically determined

Figure 5: Example detections on ETHZ demonstrating the accuracy and localization ability of our representation. Ground-truth bounding boxes are in red and our detections are in blue.

that setting all per vote weights w_i to a uniform value yields good performance. We use exactly this configuration for experiments on both the ETHZ and INRIA datasets.

Quantitative performance is reported in terms of precision-recall and false-positives per image (FPPI). We average performance results over 10 random trials and estimate a standard error. The latter has not been reported before; averaged results have been given in the past, but no indication of accuracy in the mean was provided. Further, when comparing performance, it is often the case one algorithm claims a performance improvement that could be within this standard error. We provide this information to lay a solid foundation for comparison with other work in the past and future. The computation time of our approach is low (about 15s per image), so these experiments are feasible in a reasonable amount of time.

In Figure 5 we see the detection results qualitatively. For a few examples in each of the ETHZ shape classes we show a detection bounding box and ground-truth box, followed by the interest points that voted for the detection and the rendered oriented bars. Our representation directly explains which contour pixels in the image support the object hypothesis. Compared to KAS [□], significantly more local details are captured. In Figure 6 we show more examples of true and false-positives from the ETHZ dataset. The quality of the contour representation is quite high in most detections. In the right-most column of each row we show a false-positive detection with a high score. In the case of Applelogos, a circular pattern on one of the mugs is the cause, while a bottle is detected in the pattern of a mug. The latter is due to a bottle label matching well to the clutter on a mug face. The legs of a Giraffe are seen in the upper contours of a bottle's neck and an occluded mug is detected in a bottle image. Finally we see a swan hallucinated in the swirl pattern of an apple logo. In all, we observe that the objects are generally detected well.

In Figure 7 we show the precision-recall curves for each of the classes in ETHZ and INRIA horses. We also report the mean average precision in Table 1. In all categories but

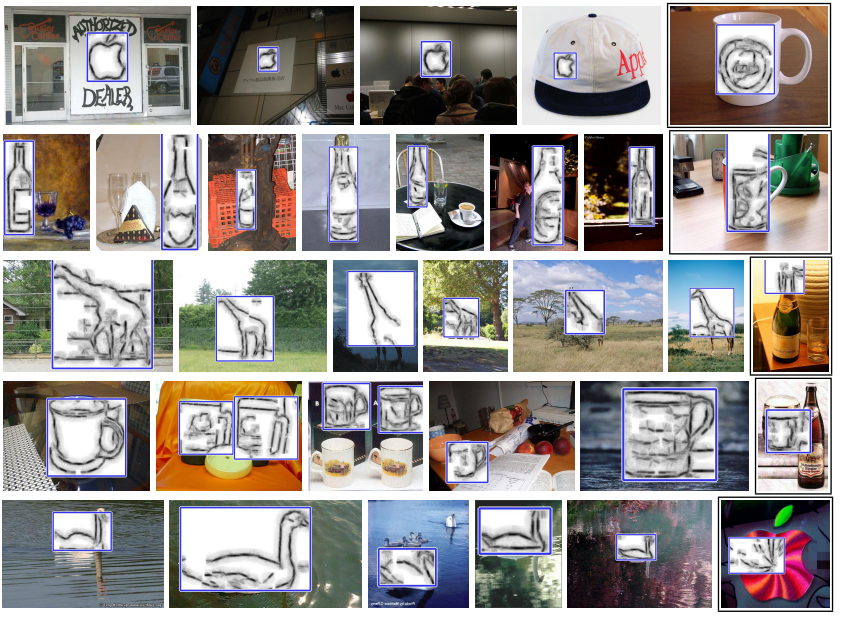


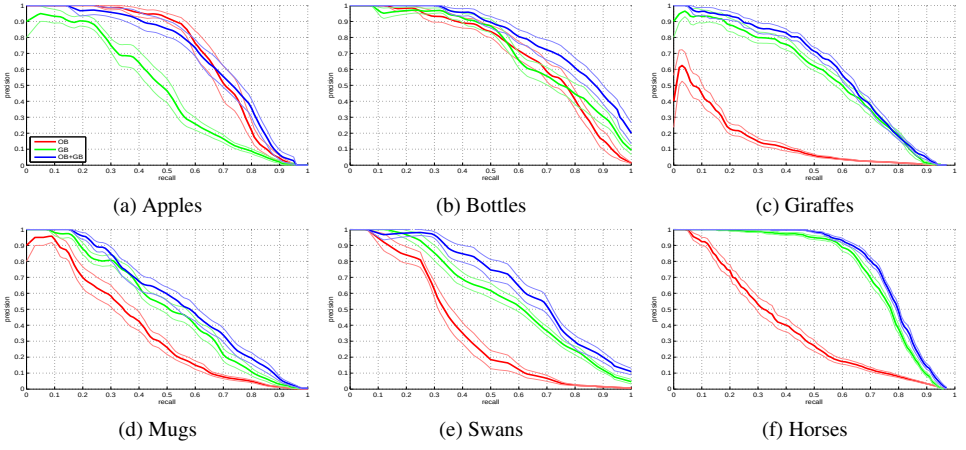
Figure 6: Example detections on ETHZ with rendered oriented bar descriptors. The rightmost image in each row (category) shows a false detection.

	OB	±	GB	±	OB+GB	±
Apples	67.4	2.6	45.5	2.9	67.0	2.9
Bottles	69.0	2.6	71.1	2.9	79.4	2.6
Giraffes	14.0	1.4	54.9	2.1	59.1	1.4
Mugs	35.7	2.2	50.2	2.0	55.3	2.3
Swans	34.5	2.1	56.6	1.7	65.2	2.4
Horses	37.6	2.0	74.4	0.9	78.1	0.7

Table 1: Mean average precision for the ETHZ and INRIA datasets. We compare our combined representation against the individual components as a baseline. The combined contour and texture representation shows a significant increase in performance.

Applelogos we see a significant gain of our combined representation. We further observe that the Applelogos are the only category that OB outperforms GB, which could be due to its simple shape. Figure 8 gives the average FPPI curves compared against a few comparable voting-based methods. Table 2 summarizes a comparison at 0.4 FPPI against our baselines and several comparable methods. In most of these cases our approach outperforms these.

One of the crucial points of this paper is that current approaches which are mainly based on texture representation are insufficient for representing object contours. Similarly edge representations such as KAS [11] are not capable of capturing the fine scale details of contour curvature and junctions with varying degree. Our evaluation demonstrates that the proposed representation is successful in modeling object appearance and also provides an accurate description of the fine-scale, local characteristics of object contours. Thus, this representation is widely applicable in part-based object models [10, 25], which are one of the leading paradigms for object detection.



(a) Apples (b) Bottles (c) Giraffes

(d) Mugs (e) Swans (f) Horses

Figure 7: Average precision-recall for the ETHZ and INRIA datasets. Red is oriented bars only, green is geometric blur only, blue is the two together.

	OB	±	GB	±	OB+GB	±	[11]	PAS[12]	TPS[12]	[18]*	[20]
Apples	76.1	2.5	65.5	2.7	81.4	2.5	60.0	28.9	83.2	85.0	75.0
Bottles	83.2	2.2	89.3	2.1	93.4	2.8	92.9	56.4	81.6	67.0	89.3
Giraffes	27.0	2.3	68.4	2.1	70.0	1.8	51.1	34.1	44.5	55.0	62.0
Mugs	52.8	2.5	69.2	2.2	74.6	2.6	77.4	35.5	80.0	55.0	65.0
Swans	54.8	3.5	84.5	2.3	90.2	3.4	52.9	44.9	70.5	42.5	53.0
Horses	40.4	2.1	77.4	2.4	79.2	0.8	76.9	45.0	68.0	52.0	–

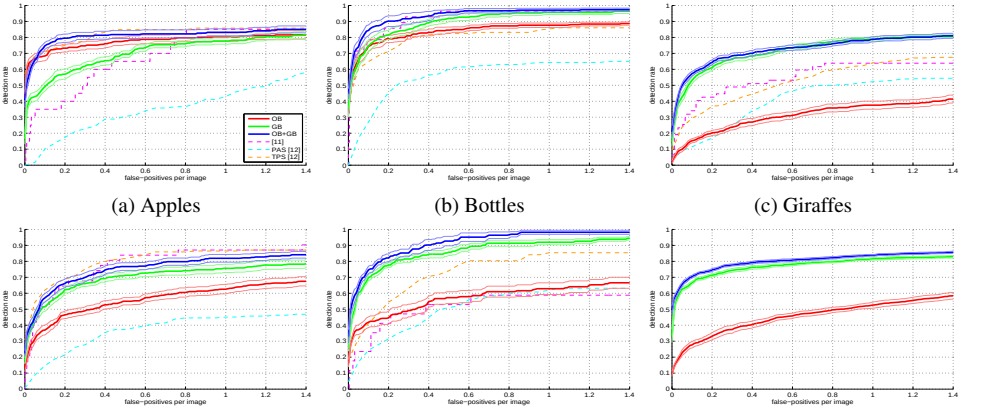
Table 2: Average detection rate at 0.4 false-positives per image on the ETHZ and INRIA datasets. We compare against several voting-based reference methods. The results for [11, 18, 20] are from voting only, no verification. Note that [18] is reported at 1.0 FPPI. PAS[12] is voting only and TPS[12] is their full system. Our combined contour and texture representation provides a clear gain and is competitive with state-of-the-art.

5 Discussion

For object detection to work, a robust and powerful representation is required. Objects are characterized by their appearance, especially their texture, and by the shape of their contours. We have presented an effective and computationally efficient representation of contours and junctions that accurately localizes and describes the local shape of contours. Contour representation and a semi-local appearance descriptor have been combined in a voting-based object detection approach. In addition to detecting objects, we showed that our approach can find the most relevant contours and junctions in each object hypothesis. Experiments have confirmed that contours and appearance are complimentary. Consequently, their combination has significantly improved the performance of baseline voting methods.

6 Acknowledgments

This work was supported by the Excellence Initiative of the German Federal Government and the Frontier fund.



(a) Apples (b) Bottles (c) Giraffes

(d) Mugs (e) Swans (f) Horses

Figure 8: Average false-positives per image for ETHZ categories and INRIA horses. Red is oriented bars only, green is geometric blur only, blue is the two together.

References

[1] Fred Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3), 1954. 1, 2

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features. *Computer Vision and Image Understanding*, 10(3), June 2008. 1

[3] Serge Belongie and Jitendra Malik. Matching shapes. In *International Conference on Computer Vision*, pages 454–461, 2001. 1

[4] Alexander C. Berg and Jitendra Malik. Geometric blur for template matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 607–615, 2001. 1, 2, 3, 6

[5] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, April 1987. 1, 2

[6] Thomas O. Binford. Visual perception by computer. In *IEEE Systems Science and Cybernetics Conference*, 1971. 1

[7] Rodney A. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17:285–348, 1981. 1

[8] M. B. Clowes. On seeing things. *Artificial Intelligence*, 2(1):79–116, 1971. 1

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, June 2005. 1, 2, 5

[10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, September 2010. 8

[11] Vittorio Ferrari, Loic Fevrier, Frederic Jurie, and Cordelia Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):36–51, January 2008. 2, 7, 8, 9

[12] Vittorio Ferrari, Frederic Jurie, and Cordelia Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, pages 1–20, 2009. 2, 9

[13] B. Julesz. Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981. 1

[14] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *European Conference on Computer Vision Workshop on Statistical Learning in Computer Vision*, pages 17–32, 2004. 4

[15] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987. 1

[16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1

[17] Michael Maire, Pablo Arbeláez, Charless Fowlkes, and Jitendra Malik. Using contours to detect and localize junctions in natural images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 3

[18] Subhransu Maji and Jitendra Malik. Object detection using max-margin Hough transform. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1038–1045, 2009. 4, 9

[19] David R. Martin, Charless Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):530–549, May 2004. 3, 5, 6

[20] Björn Ommer and Jitendra Malik. Multi-scale object detection by clustering lines. In *International Conference on Computer Vision*, pages 484–491, 2009. 4, 9

[21] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, 80(1):16–44, 2008. 2

[22] Alex P. Pentland. Recognition by parts. In *International Conference on Computer Vision*, pages 612–620, 1987. 1

[23] Xiaofeng Ren, Charless Fowlkes, and Jitendra Malik. Figure/ground assignment in natural images. In *European Conference on Computer Vision*, volume 2, pages 614–627, 2006. 2

[24] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Contour-based learning for object detection. In *International Conference on Computer Vision*, pages 503–510, 2005. 2

[25] Praveen Srinivasan, Qihui Zhu, and Jianbo Shi. Many-to-one contour matching for describing and discriminating object shape. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1673–1680, 2010. 8