Using Partial Edge Contour Matches for Efficient Object Category Localization

Hayko Riemenschneider, Michael Donoser, and Horst Bischof*

Institute for Computer Graphics and Vision,
Graz University of Technology, Austria
{hayko,donoser,bischof}@icg.tugraz.at

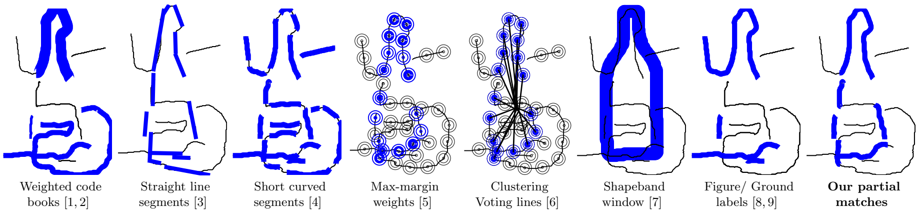
Abstract. We propose a method for object category localization by partially matching edge contours to a single shape prototype of the category. Previous work in this area either relies on piecewise contour approximations, requires meaningful supervised decompositions, or matches coarse shape-based descriptions at local interest points. Our method avoids error-prone pre-processing steps by using all obtained edges in a partial contour matching setting. The matched fragments are efficiently summarized and aggregated to form location hypotheses. The efficiency and accuracy of our edge fragment based voting step yields high quality hypotheses in low computation time. The experimental evaluation achieves excellent performance in the hypotheses voting stage and yields competitive results on challenging datasets like ETHZ and INRIA horses.

1 Introduction

Object detection is a challenging problem in computer vision. It allows localization of previously unseen objects in images. In general, two main paradigms can be distinguished: appearance and contour. Appearance-based approaches form the dominant paradigm using the bag-of-words model [10], which analyzes an orderless distribution of local image features and achieves impressive results mainly because of powerful local image description [11].

Recently, the contour-based paradigm has become popular, because shape provides a powerful and often more generic feature [12] since an object contour is invariant to extreme lighting conditions and large variations in texture or color. Many different contour-based approaches exist and the research falls mainly into four categories. The works of [1, 2] focus on the aspect of learning edge codebooks, where chamfer matching is used to evaluate local shape similarity. Other research uses piecewise approximations of the edges by short segments [13, 4] or supervised decompositions [8]. In [5, 6, 14] the problem is cast as a matching between shape-based descriptors on local interest points.

* This work was supported by the Austrian Research Promotion Agency (FFG) project FIT-IT CityFit (815971/14472-GLE/ROD).



Weighted code books [1, 2] | Straight line segments [3] | Short curved segments [4] | Max-margin weights [5] | Clustering Voting lines [6] | Shapeband window [7] | Figure/ Ground labels [8, 9] | **Our partial matches**

Fig. 1. Overview of related work: Our approach relaxes the piecewise approximations and local neighborhoods. We use partial matching to find contour fragments belonging to the foreground rather than discarding entire edges. See Section 2 for details.

The main motivation for our work is that *"connectedness is a fundamental powerful driving force underexploited in object detection"* [3]. Viewing edge contours as connected sequences of any length instead of short segment approximations or local patches on interest points provides more discrimination against background cutter. In our contributions we focus on the partial matching of noisy edges to relax the constraints on local neighborhoods or on assigning entire edges as background disregarding local similarities. We formulate a category localization method which efficiently retrieves partial edge fragments that are similar to a single contour prototype. We introduce a self-containing descriptor for edges which enables partial matching and an efficient selection and aggregation of partial matches to identify and merge similar overlapping contours up to any length. A key benefit is that the longer the matches are, the more they are able to discriminate between background clutter and the object instance. In this way we lift standard figure / ground assignment to another level by providing local similarities for all edges in an image. We retrieve these partial contours and combine them directly in a similarity tensor and together with a clustering-based center voting step we hypothesize object locations. This greatly reduces the search space to a handful of hypotheses and shows excellent performance compared to state of the art in the voting stage. For a full system evaluation, the hypotheses are further verified by a standard multi-scale histogram of gradients (HOG) classifier.

2 Related Work

There exists a range of work in the contour-based paradigm which achieve state-of-the-art performance for several object categories using contour information, for an overview see Figure 1. The research falls into four main categories, namely (i) learning codebooks of contour fragments, (ii) approximating contours by piecewise segments, (iii) using local description of the contour at selected interest points, or (iv) assigning entire edges to either foreground or background. Additional techniques are used in each work, for example learning deformation models, sophisticated cost functions or probabilistic grouping.

Learning codebooks: Shotton et al. [1] and Opelt et al. [2] concurrently proposed to construct shape fragments tailored to specific object classes. Both find matches to a pre-defined fragment codebook by chamfer matching to the query image and then find detections by a star-shaped voting model. Their methods rely on chamfer matching which is sensitive to clutter and rotation. In both approaches the major aspect is to learn discriminative combinations of boundary parts as weak classifiers using boosting to build a strong detector.

Piecewise approximation: Ferrari et al. [15, 3] build groups of approximately straight adjacent segments (kAS) to work together in a team to match the model parts. The segments are matched within a contour segmentation network which provides the combinations of multiple simple segments using the power of connectedness. In later work they also show how to automatically learn codebooks [3], or how to learn category shape models from cropped training images [16]. In a verification step they use a thin-plate-spline (TPS)-based matching to accurately localize the object boundary. Similar to this, Ravishankar et al. [4] use short segments to approximate the outer contour of objects. In contrast to straight segments, they prefer slightly curved segments to have better discriminative power between the segments. They further use a sophisticated scoring function which takes local deformations in scale and orientations into account. However, they break the reference template at high curvature points to be able to match parts, again resulting in disjoint approximations of the actual contour. In their verification stage, the gradient maps are used as underlying basis for object detection avoiding the error-prone detection of edges.

Shape-based interest points: This category uses descriptors to capture and match coarse descriptions of the local shape around interest points. Leordeanu et al. [17] use simple features based on normal orientations and pairwise interactions between them to learn and detect object models in images. Their simple features are represented in pairwise relations in category specific models that can learn hundreds of parts. Berg et al. [14] formulate the object detection problem as a deformable shape matching problem. However, they require hand-segmented training images and do not learn deformation models in training. Further in the line are the works of Maji and Malik [5] and Ommer and Malik [6] which match geometric blur features to training images. The former use a max-margin framework to learn discriminative weights for each feature type to ensure maximal discrimination during the voting stage. The latter provide an interesting adaptation of the usual Hough-style center voting. Ommer and Malik transform the discrete scale voting to a continuous domain where the scale is another unknown in the voting space. Instead of multiple discrete center vectors, they formulate the votes as lines and cluster these to find scale-coherent hypotheses. The verification is done using a HOG-based fast SVM kernel (IKSVM).

Figure / ground assignment: Similar in concept but not in practice are the works of Zhu et al. [8] and Lu et al. [9]. They cast the problem as figure / ground labeling of edges and decide for a rather small set of edges which belong to the foreground and which are background clutter. By this labeling they reduce the clutter and focus on salient edges in their verification step. Lu et al. use particle filters under static observation to simultaneously group and label the edge contours. They use a new shape descriptor based on angles to decide edge contour similarity. Zhu et al. use control points along the reference contour to find possible edge contour combinations and then solve cost functions efficiently using linear programming. They find a maximal matching between a set of query image contours and a set of salient contour parts from the reference template, which was manually split into a set of reference segments. Both assume to match entire edge contours to the reference sets and require long salient contours. Recent work by Bai et al. [7] is also based on a background clutter removal stage called shapeband. Shapeband is a new type of sliding window adapted to the shape of objects. It is used to provide location hypotheses and to select edge contour candidates. However, in their runtime intensive verification step they iteratively compute shape context descriptors [18] to select similar edge contours. Another recent approach by Gu et al. [19] proposes to use regions instead of local interest points or contours to better estimate the location and scale of objects.

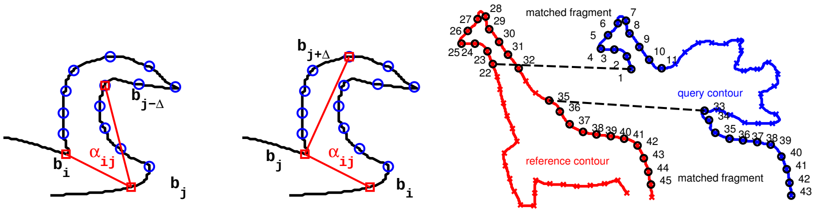
We place our method in between the aforementioned approaches. We use edge contours in the query image and match them at any length from short contour segments up to full regions boundaries using partial shape matching. In such a setting the similarity to the prototype shape decides the complexity and length of the considered contours.

3 Partial Shape Matching for Object Detection

In the following sections we describe our proposed approach to detect objects by computing partial similarities between edge contours in a query image and a reference template. For the sake of clarity, we will now define some terms used throughout the paper, see Figure 2 for a visual illustration. We use the term fragment to denote a part of an edge contour. Edge contours can be arbitrarily long, contain irrelevant parts or may also be incomplete due to missing edge detector responses or occlusions which make parts of the object invisible. The query contours are the connected edge contours found by the detector and subsequent 8-neighborhood linking. The reference contour is a single hand-drawn model of the object's outer boundary. A valid matched fragment is defined as a part of an edge contour that is similar to a part of the reference contour.

Our goal is to identify matches from fragments of arbitrary length (contained within the query edges) to the reference contour, by analyzing a self-contained representation and description of the shape of the detected edge contours. We want to build a representation that contains the whole as well as any part of a contour, which enables matching independently from the remaining parts.

Our detection method consists of three parts: First, edges are extracted from an image and represented as lists of coordinates. This representation is the basis for the self-containing contour descriptor. Second, the matching is a vital component which allows the efficient retrieval of contour fragments similar to a given



(a) $i < j$ (b) $i > j$ (c) Partial matches

Fig. 2. Illustration of 2D angle description and matching. a-b) An angle is measured between any two sampled points b_i and b_j, which define a fragment inside an edge contours, c) shows partial matches in an occluded edge to a reference contour.

reference prototype. Third, for each matched fragment we calculate a center vote to estimate the location of the searched object and aggregate coherent fragments based on their voting, scale and correspondence to the reference.

3.1 Fragment Description

Our goal is to exploit the connectedness of an edge contour implicitly yet allowing to retrieve parts of an edge as fragments. Many different methods have been proposed for partial contour matching. Angular representations are a natural choice due their direct encoding of geometric layout. For example, Turney et al. [20] use the slope θ and arc length s as local representation for boundaries, however only on a small set of images. In a more recent work Brendel and Todorovic [21] find matching fragments in complex images using circular dynamic time warping with a runtime of 200ms per match. Chen et al. [22] proposed an efficient matching, however their descriptor only measures local shape and ignores global contour similarity. Felzenszwalb et al. [23] proposed a hierarchy of deformable shapes where only a single contour can be matched in subtrees and matching two contours requires 500ms. A recent hierarchical approach by Kokkinos and Yuille [24] formulates the task as image parsing and provide fast coarse to fine matches. Lu et al. [9] developed a shape descriptor based on a 3D histogram of angles and distances for triangles connecting points sampled along the contours. They do not allow partial matching and the descriptor requires high computational costs. Donoser et al. [25] developed a descriptor which can be seen as a subset of [9], where angles between any two points and a fixed third point on a closed contour are analyzed. They demonstrate efficient matching between two closed shapes within a few milliseconds.

Inspired by the high quality of hierarchical approaches, we adopt the descriptor from Donoser et al. [25], which was designed for matching whole object silhouettes, to handle the requirements of object detection in cluttered images. First, partial matching of the cluttered edges must be possible. Since there are no closed contours around a cluttered object after standard edge detection, we design a novel self-containing descriptor which enables efficient partial matching. Second, similar to hierarchies the descriptor encodes coarse and fine contour information. Different sampling of the descriptor enables direct access to different levels of detail for the contour, whereas the full descriptor implicitly contains all global and local contour information.

The method in [25] proposed an efficient matching step to describe and then retrieve all redundant and overlapping matching combinations. Their brute force algorithm delivers good results on clean silhouette datasets. However, for an object detection task this is not feasible due to the prohibitive combinators (multiple scales, multiple occlusions and hundreds of edges per image). Additionally, slightly shifted matches at neighboring locations contradict each other and do not provide coherent object location hypotheses. Therefore, in contrast to [25], we propose an efficient summarization scheme directly in an obtained 3D similarity tensor. Such an approach has several strong benefits like selection and aggregation of only coherent center vote matches, longer merged matches out of indiscriminate shorter segments and further an immense speedup due to the reduction of the number of returned matches. The main motivation is to exploit the connectedness of edge contours instead of using individual interest points or short piecewise approximations of edge contours.

As a first step we sample a fixed number N of points from the closed reference contour that can be ordered as $R = \{r_1, r_2, \ldots, r_N\}$. As next step we have to extract connected and labeled edge contours from the query image. Edge detection and linking in general is a quite challenging task [26]. We apply the Pb edge detector [27] and link the results to a set of coordinate lists. For the obtained query contours, points are sampled at equal distance, resulting in a sequence of points $B = \{b_1, b_2, \ldots, b_M\}$ per contour. The sampling distance d between the points allows to handle different scales. Sampling with a larger distance equals to a larger scale factor, and vice versa. For detecting objects in query images we perform an exhaustive search over a range of scales, which is efficiently possible due to the properties of our descriptor and matching method.

We use a matrix of angles which encode the geometry of the sampled points leading to a translation and rotation invariant description for a query contour. The descriptor is calculated from the relative spatial orientations between lines connecting the sampled points. In contrast to other work [9, 25], we calculate angles α_{ij} between a line connecting the points b_i and b_j and a line to a third point relative to the position of the previous two points. This angle is defined

$$\alpha_{ij} = \begin{cases} \sphericalangle(\overline{b_i b_j}, \overline{b_j b_{j-\Delta}}) & \text{if } i < j \\ \sphericalangle(\overline{b_i b_j}, \overline{b_j b_{j+\Delta}}) & \text{if } i > j \\ 0 & \text{if } abs(i-j) \leq \Delta \end{cases} , \qquad (1)$$

where b_i and b_j are the i^{th} and j^{th} points in the sequence of sampled points of the contour and Δ is an offset parameter of the descriptor (5 for all experiments). See Figure 2 for an illustration of the choice of points along the contour. The third point is chosen depending on the position of the other two points to ensure that the selected point is always inside the contour. This allows us to formulate the descriptor as a self-containing descriptor of any of its parts.

The angles α_{ij} are calculated between every pair of points along a contour. In such a way a contour defined by a sequence of M points is described by an $M \times M$ matrix where an entry in row i and column j yields the angle α_{ij}. Figure 3 illustrates the descriptors for different shape primitives. The proposed descriptor has four important properties. First, its angular description makes it translation and rotation invariant. Second, a shift along the diagonal of the descriptor handles the uncertainty of the starting point in edge detection. Third, it represents the connectedness of contours by using the sequence information providing a local (close to matrix diagonal) and global (far from matrix diagonal) description. And most importantly, the definition as a self-containing descriptor allows to implicitly retrieve partial matches which is a key requirement for cluttered and broken edge results.

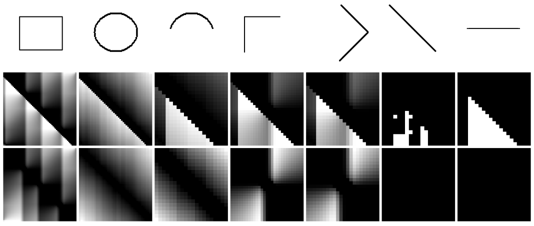


Fig. 3. Visualization of descriptors for selected contour primitives. Middle row shows descriptors from [25] and bottom row shows our descriptors. Note how each fragment is included in its respective closed contours (square and circle) in our version, which is not fulfilled for [25] since it was designed for closed contour matching.

3.2 Fragment Matching and Merging

Matching and merging partial contours is an important part of our approach and is based on the 2D edge contour descriptors introduced in the previous section. For any two descriptors representing two contours, the aim of matching is to identify parts of the two contours which are similar to each other. In terms of comparing corresponding descriptor matrices, one has to compare all sub-blocks of the descriptor matrices to find all matching possibilities and lengths. For efficient calculation of all similarity scores, we apply the algorithmic optimization using integral images as proposed in [25] to access the partial descriptor differences in constant time, which returns the similarities (differences between our angle descriptors) for all matching triplets $\{r, q, l\}$ stored in a 3D similarity and correspondence tensor $\Gamma_{(r,q,l)}$. The first two dimensions identify the starting points of the match in the reference (r) and query edge contour (q) and the third dimension defines the length (l) of the match. Note that this tensor fully defines all possible correspondences between the reference and the edge contour. Figure 4 shows the similarity tensor for the partial matching example in Figure 2. The two matched fragments correspond to the peaks (a) in the tensor, here a single slice at a fixed length $l = 11$ is shown.

A main issue is redundancy within the tensor as there may be many overlapping and repetitive matches. This poses a problem for object detection in cluttered images. Our goal is to find the longest and most similar fragments and merge repetitive matches instead of retrieving all individual matches. This is an important part of this work. First, it is necessary to outline some of the properties of our 3D similarity and correspondence tensor $\mathit{\Gamma}_{(r,q,l)}$.

I. A fragment (r, q, l) is assigned a similarity (Euclidean distance between angular descriptors) by $\mathit{\Gamma}_{(r,q,l)}$.
II. Length variations (r, q, l_2) with $l_2 < l$ define the same correspondence, yet shorter in length.
III. Diagonal shifts in the indices $(r + 1, q + 1, l)$ also represent the same match, yet one starting point *later*.
IV. Unequal shifts $(r + 1, q, l)$ define a different correspondence, however very similar and close.
V. Due to occlusions or noise, multiple matches per edge contour may exist. The example in Figure 2 is a shifted match *much later* $(r + 13, q + 32, l)$ defining the same correspondence, yet skipping (32-13=19) points of noise.
VI. Matches near to the end of each contour (if not closed) have a maximal length given by the remaining points in each contour sequence.

Perfect matches would result in singular *peaks* in a slice. However due to these small shifts along the same correspondence or with an unequal offset, matches result in a *hill*-like appearance of the similarity, see Figure 4a. Given these properties we now define a matching criterion to deliver the longest and most similar matches, i.e. finding the peaks not once per slice but for the entire 3D tensor. This summarization is made of three steps: (a) finding valid correspondences satisfying the constraints on length and similarity, (b) merging all valid correspondences to obtain the longest combination of the included matches (property II) and (c) selecting the maximal similarity of matches in close proximity (property IV). The steps are in detail as following:

First, we define a function $\mathcal{L}(r, q, l)$ which gives the lengths at any given valid correspondence tuple r, q as

$$\mathcal{L}(r,q,l) = \begin{cases} l & \text{if } \mathit{\Gamma}_{(r,q,l)} \leq s_{lim} \text{ and } l \geq l_{lim} \\ 0 & \text{else} \end{cases}, \tag{2}$$

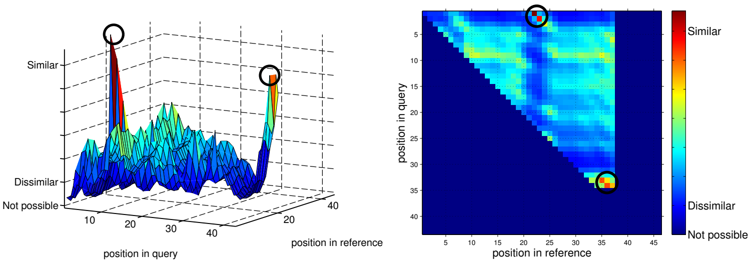
where the value at $\mathcal{L}(r, q, l)$ is the length of a valid fragment. A valid fragment has a similarity score below the limit s_{lim} and a minimal length limit of l_{lim}. This function is used to define a subset of longest candidates by

$$\Psi_{(r,q)} = \forall_{r,q} : \underset{l\in\min(N,M)}{\arg\max}\ \mathcal{L}(r,q,l), \tag{3}$$

where $\Psi_{(r,q)}$ is a subset of $\Gamma_{(r,q,l)}$ containing the longest matches at each correspondence tuple (r,q). This set contains matches for every possible correspondence given by the constraints on similarity and matching positions (see property II, VI). However, we further want to reduce this to only the local maxima (conserving property IV). Since the set can now be considered as a 2D function, we find the connected components $\mathcal{C}$ satisfying $\Psi_{(r,q)} > 0$. The final set of candidates are the maxima per connected component and is defined as

$$\Upsilon_{(r,q,l)} = \forall c_i \in C : \underset{\Gamma_{(r,q,l)}}{\arg\max}\,(\Psi_{(r,q)} \in c_i), \tag{4}$$

where $\Upsilon_{(r,q,l)}$ holds the longest possible and most similar matches given the constraints on minimum similarity s_{lim} and minimal length l_{lim}. In the example shown in Figure 2 and 4 the final set contains two matches, which are the longest possible matches. Note that shorter matches in the head and back are possible, but are directly merged to longer and more discriminative matches by analyzing the whole tensor. Furthermore, obtained matches are local maxima concerning similarity scores. This provides an elegant and efficient summarization leading to coherent and discriminative matches and reduced runtime.



(a) Similarity scores in 3D (b) A single slice in 2D

Fig. 4. Illustration of the similarity and correspondence tensor $\Gamma_{(r,q,l)}$ at length $l = 11$ for example shown in Figure 2(c): (a) the two peaks correspond to the matches found. Matching uncertainty results in multiple peaks in a *hill*-like appearance. (b) shows the same similarity in a flat view, where red signals high similarity and dark blue defines invalid matches due to length constraints. Best viewed in color.

3.3 Hypothesis Voting

Matching as described in the previous section provides a set of matched fragments for the query edges, which have to be combined to form object location hypotheses. In the following we describe how matched fragments are grouped for object locations hypotheses and scores are estimated.

Fragment Aggregation Up to this point we have a set $\Upsilon_{(r,q,l)}$ of matched parts of edge contours detected in a query image which are highly similar to the provided prototype contour. Every match has a certain similarity and length. Further, we can map each matched contour to its reference contour and estimate the object centroid from the given correspondence tuple. The aggregation of the individual fragments identifies groups of fragments which compliment each other and form object location hypotheses.

For this step we cluster the matched fragments analyzing their corresponding center votes and their scale by mean-shift mode detection with a scale-dependent bandwidth. The bandwidth resembles an analogy to the classical Hough accumulator bin size, however with the added effect that we combine the hypotheses locations in a continuous domain rather than discrete bins.

Hypothesis Ranking All obtained hypotheses are ranked according to a confidence. For this purpose we investigate two ranking methods. The first is based on the coverage of detected fragments, where ζ_{COV} is a score relative to the amount of the reference contour that is covered by the matched fragments, defined as

$$\zeta_{COV} := \frac{1}{N} \sum_{i=1}^{N} (f_i \times S_i), \tag{5}$$

where f_i is the number of times the i-th contour point has been matched and S_i is the corresponding weight of this point. This is normalized by the number of contour points N in the reference contour. The coverage score ζ_{COV} provides a value describing how many parts are matched to the reference contour for the current hypothesis. We use a uniform weight of $S_i = 1$. However, for example weights given by the contour flexibility [28] would be an interesting aspect.

As a second score, we use a ranking as proposed by Ommer and Malik [6]. They define the ranking score ζ_{PMK} by applying an SVM classifier to the image windows around the location hypotheses. The kernel is the pyramid match kernel (PMK) [29] using histograms of oriented gradients (HOG) as features. Positive samples for each class are taken from the ground truth training set. Negative samples are retrieved by evaluating the hypotheses voting and selecting the false positives. The bounding boxes are resized to a fixed height while keeping median aspect ratio. Since the mean-shift mode detection may not deliver the true object location, we sample locations in a grid of windows around the mean-shift center. At each location we evaluate the aforementioned classifier and retrieve the highest scoring hypothesis as new detection location.

4 Experiments

We demonstrate the performance of our proposed object category localization method on two different reference data sets: ETHZ (Section 4.1) and INRIA

Voting and Ranking Stage (FPPI=1.0)							*Verification Stage (FPPI=0.3/0.4)*				
ETHZ	Hough	M^2HT	w_{ac}	**Our**	PMK	**Our**	M^2HT	PMK	KAS	System	**Our**
Classes	[13]	[5]	[6]	**work**	[6]	**work**	[5]	[6]	[3]	Full [13]	**work**
Apples	43.0	80.0	85.0	90.4	80.0	90.4	95.0/95.0	95.0/95.0	50.0/60.0	77.7/83.2	93.3/93.3
Bottles	64.4	92.4	67.0	84.4	89.3	96.4	92.9/96.4	89.3/89.3	92.9/92.9	79.8/81.6	97.0/97.0
Giraffes	52.2	36.2	55.0	50.0	80.9	78.8	89.6/89.6	70.5/75.4	49.0/51.1	39.9/44.5	79.2/81.9
Mugs	45.1	47.5	55.0	32.3	74.2	61.4	93.6/96.7	87.3/90.3	67.8/77.4	75.1/80.0	84.6/86.3
Swans	62.0	58.8	42.5	90.1	68.6	88.6	88.2/88.2	94.1/94.1	47.1/52.4	63.2/70.5	92.6/92.6
Average	53.3	63.0	60.9	**69.4**	78.6	**83.2**	91.9/93.2	87.2/88.8	61.4/66.9	67.2/72.0	**89.3/90.5**

Table 1. Hypothesis voting, ranking and verification stages show competitive detection rates using PASCAL criterion for the ETHZ shape database [15] compared to related work. For the voting stage our coverage score increases the performance by 6.5% [6], 8.5% [5] and 16.1% [13] leading to state-of-the-art voting results at reduced runtime.

horses (Section 4.2). We significantly outperform related methods in the hypotheses generation stage, while attaining competitive results for the full system. Results demonstrate that exploiting the connectedness of edge contours in a partial contour matching scenario enables to accurately localize category instances in images in efficient manner. Note also that we only use binary edge information for the hypothesis voting and do not include edge magnitude information, which plays important roles in other work [3, 4, 6, 5].

Our proposed object localization method is not inherently scale invariant. We analyze 10 scales per image, where scale is defined by the distance between the sampled points. Localization of an object over all scales (!) requires on average only 5.3 seconds per image for ETHZ in a Matlab implementation.

4.1 ETHZ Shape Classes

Results are reported on the challenging ETHZ shape dataset consisting of five object classes and a total of 255 images. All classes contain significant intra-class variations and scale changes. The images sometimes contain multiple instances of a category and have a large amount of background clutter.

Unfortunately direct comparison to related work is quite hard since many different test protocols exist. Foremost, on the ETHZ dataset there exist two main methods for evaluation. First, a class model is learned by training on half of the positive examples from a class, while testing is done on all remaining images (half of positive examples and all other negative classes) averaged over five random splits. Second, the ETHZ dataset additionally provides hand-drawn templates per class to model the categories. This step requires no training and has shown to provide slightly better results in a direct comparison [13]. Further, the detection performance may be evaluated using one of the two measures, namely the stricter PASCAL or the 20%-IoU criterion, which require that the intersection of the bounding box of the predicted hypotheses and the ground truth over the union of the two bounding boxes is larger than 50% or 20% respectively. Additional aspects in the evaluation are the use of 5-fold cross validation, aspect ratio voting and most influential the use of features. Using strong features including color and appearance information naturally has a benefit over gradient information and again over pure binary shape information. This spectrum of features has the benefit to complement each other. Thus in our approach we use the hand-drawn models to match only binary edges in an query image and for a full system we further verify their location using a standard gradient-based classifier trained on half of the positive training samples.

Class-wise results for ETHZ using the strict PASCAL criterion are given in Table 1. The focus of this work lies on hypothesis voting stage, where we can show excellent results of 69.4% and 83.2%, without and with a PMK classifier ranking. The PMK ranking increases the scores for three classes (bottles, giraffes and mugs). The reason is that the classifier is better able to predict the instance of these classes, especially for mugs, where our system produces twice as many hypotheses compared to the other classes (on average 20 for mugs compared to 8 for the other classes). The coverage score performs better on compact object classes (applelogos and swans). Please note, the other methods do not use hand-drawn prototypes. We achieve an overall improvement over related work ranging from 6.5% [6], 8.5% [5] to 16.1% [13] without classifier ranking, and 4.6% over [6] using a classifier ranking. We also achieve competitive results after verification of 90.5% compared to 66.9% [3], 72.0% [13], 88.8% [6] and 93.2% [5] at 0.4 FPPI.

Due to the lack of hypothesis voting results for other approaches, we also provide a range of comparisons with previous work using the full system. We evaluate our method using the 20%-IoU criterion and summarize the results in Table 2. Compared to related work we also achieve excellent results using this criterion. Note again, that direct comparison has to be seen with caution, since methods either use hand-drawn or learned models. See Figure 5 for some exemplary successful detections and some failure cases.

ETHZ shape classes: *Verification Stage (FPPI = 0.3/0.4) using 20%-IoU*							
Method	Supervised	Template	Template	Template	Codebook	Learned	Template+Learned
	Lu [9]	Ravishankar [4]	Ferrari [15]	Ferrari [16]	Ferrari [3]	Ferrari [16]	**Our work**
Average	90.3/91.9	93.0/95.2	70.5/81.5	82.4/85.3	74.4/79.7	71.5/76.8	**94.4/95.2**

Table 2. Average detection rates for related work on hand-drawn and learned models.

4.2 INRIA Horses

As a second dataset we use the INRIA horses [13], which consists of 170 images with one or more horses in side-view at several scales and cluttered background, and 170 images without horses. We use the same training and test split as [13] of 50 positive examples for the training and test on the remaining images (120+170). We again use only a single reference template which was chosen from the pixel-wise segmentation of a random horse from the training set. For this dataset the performance is 83.72% at FPPI=1.0 and thus is better than recent results 73.75% by [16], 80.77% by [3] and almost as good as 85.27% by [5], which

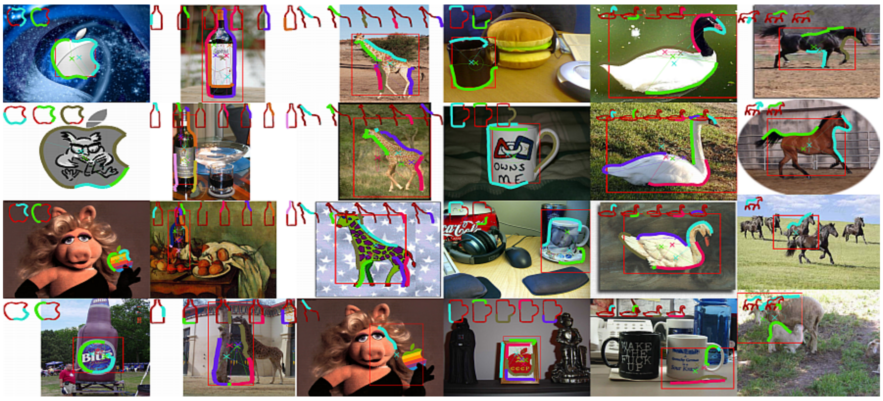


Fig. 5. Results on ETHZ shape classes and INRIA horses (also see additional material).

additionally vote for aspect ratios. Presumably this would also increase our recall for the strongly articulated horses since we detect the partial matches, however a single rigid reference template does not capture the centroid change.

5 Conclusion

We have presented a new approach in the paradigm of contour-based object detection based on partial contour matches to a reference template and show competitive results on state-of-the-art datasets like ETHZ shape and INRIA horses. Complementary to related work, we demonstrated that we can relax the approximations by piecewise segments by providing partial matching of contours instead of selecting or ignoring complete contours as well as extending the search beyond local neighborhoods of interest points. Our system implicitly handles parts of a contour and thus does not require grouping long salient curves or harmful splitting of contours to be able to match parts. Though a verification stage is a vital part for a full object detection system, we believe the focus should lie on better reflecting the hypotheses voting space, since this has a direct effect on the speed and accuracy of the full detector performance. In future work we will investigate learning discriminating weights [1, 5] and interactions between contour fragments [17, 3].

References

1. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. In: ICCV. (2005)
2. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. In: ECCV. (2006)
3. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. PAMI (2008)

4. Ravishankar, S., Jain, A., Mittal, A.: Multi-stage contour based detection of deformable objects. In: ECCV. (2008)
5. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR. (2009)
6. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: ICCV. (2009)
7. Bai, X., Li, Q., Latecki, L., Liu, W., Tu, Z.: Shape band: A deformable object detection approach. In: CVPR. (2009)
8. Zhu, Q., Wang, L., Wu, Y., Shi, J.: Contour context selection for object detection: A set-to-set contour matching approach. In: ECCV. (2008)
9. Lu, C., Latecki, L., Adluru, N., Ling, H., Yang, X.: Shape guided contour fragment grouping with particle filters. In: ICCV. (2009)
10. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV. (2003)
11. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI (2005)
12. Biederman, I.: Human image understanding: Recent research and a theory. In: Computer Vision, Graphics, and Image Processing. Volume 32. (1985)
13. Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. In: IJCV. (2009)
14. Berg, A., Berg, T., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: CVPR. (2005)
15. Ferrari, V., Tuytelaars, T., Gool, L.V.: Object detection by contour segment networks. In: ECCV. (2006)
16. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detections with deformable shape models learnt from images. In: CVPR. (2007)
17. Leordeanu, M., Hebert, M., Sukthankar, R.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In: CVPR. (2007)
18. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. PAMI (2002)
19. Gu, C., Lim, J., Arbelaez, P., Malik, J.: Recognition from regions. In: CVPR. (2009)
20. Turney, J., Mudge, T., Volz, R.: Recognizing Partially Occluded Parts. PAMI (1985)
21. Brendel, W., Todorovic, S.: Video object segmentation by tracking regions. In: ICCV. (2009)
22. Chen, L., Feris, R., Turk, M.: Efficient partial shape matching using smith-waterman algorithm. In: NORDIA. (2008)
23. Felzenszwalb, P., Schwartz, J.: Hierarchical matching of deformable shapes. In: CVPR. (2007)
24. Kokkinos, I., Yuille, A.: Hop: Hierarchical object parsing. In: CVPR. (2009)
25. Donoser, M., Riemenschneider, H., Bischof, H.: Efficient partial shape matching of outer contours. In: ACCV. (2009)
26. Donoser, M., Riemenschneider, H., Bischof, H.: Linked Edges as Stable Region Boundaries. In: CVPR. (2010)
27. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI (2004)
28. Xu, C., Liu, J., Tang, X.: 2D Shape Matching by Contour Flexibility. PAMI (2009)
29. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV. (2005)