A Boundary-Fragment-Model for Object Detection

Andreas Opelt[1], Axel Pinz[1], and Andrew Zisserman[2]

[1] Vision-based Measurement Group, Inst. of El. Measurement and Meas. Sign. Proc. Graz, University of Technology, Austria
{opelt, axel.pinz}@tugraz.at
[2] Visual Geometry Group, Department of Engineering Science, University of Oxford
az@robots.ox.ac.uk

Abstract. The objective of this work is the detection of object classes, such as airplanes or horses. Instead of using a model based on salient image fragments, we show that object class detection is also possible using only the object's boundary. To this end, we develop a novel learning technique to extract class-discriminative boundary fragments. In addition to their shape, these "codebook" entries also determine the object's centroid (in the manner of Leibe *et al.* [19]). Boosting is used to select discriminative combinations of boundary fragments (weak detectors) to form a strong "Boundary-Fragment-Model" (BFM) detector. The generative aspect of the model is used to determine an approximate segmentation.

We demonstrate the following results: (i) the BFM detector is able to represent and detect object classes principally defined by their shape, rather than their appearance; and (ii) in comparison with other published results on several object classes (airplanes, cars-rear, cows) the BFM detector is able to exceed previous performances, and to achieve this with less supervision (such as the number of training images).

1 Introduction and Objective

Several recent papers on object categorization and detection have explored the idea of learning a codebook of appearance parts or fragments from a corpus of images. A particular instantiation of an object class in an image is then composed from codebook entries, possibly arising from different source images. Examples include Agarwal & Roth [1], Vidal-Naquet & Ullman [27], Leibe *et al.* [19], Fergus *et al.* [12, 14], Crandall *et al.* [9], Bar-Hillel *et al.* [3]. The methods differ on the details of the codebook, but more fundamentally they differ in how strictly the geometry of the configuration of parts constituting an object class is constrained. For example, Csurka *et al.* [10], Bar-Hillel *et al.* [3] and Opelt *et al.* [22] simply use a "bag of visual words" model (with no geometrical relations between the parts at all), Agarwal & Roth [1], Amores *et al.* [2], and Vidal-Naquet and Ullman [27] use quite loose pairwise relations, whilst Fergus *et al.* [12] have a strongly parametrized geometric model consisting of a joint Gaussian over the

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part II, LNCS 3952, pp. 575–588, 2006.

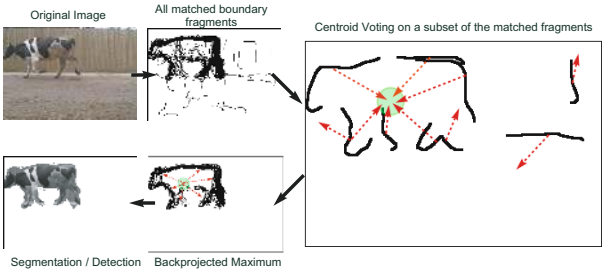

Fig. 1. An overview of applying the BF model detector

centroid position of *all* the parts. The approaches using no geometric relations are able to categorize images (as containing the object class), but generally do not provide location information (no detection). Whereas the methods with even loose geometry are able to detect the object's location.

The method of Leibe *et al.* ([19], [20]) has achieved the best detection performance to date on various object classes (e.g. cows, cars-rear (Caltech)). Their representation of the geometry is algorithmic – all parts vote on the object centroid as in a Generalized Hough transform. In this paper we explore a similar geometric representation to that of Leibe *et al.* [19] but use only the *boundaries* of the object, both internal and external (silhouette). In our case the codebook consists of *boundary-fragments*, with an additional entry recording the location of the object's centroid. Figure 1 overviews the idea. The boundary represents the shape of the object class quite naturally without requiring the appearance (e.g. texture) to be learnt. For certain categories (bottles, cups) where the surface markings are very variable, approaches relying on consistency of these appearances may fail or need considerable training data to succeed. Our method, with its stress on boundary representation, is highly suitable for such objects. The intention is not to replace appearance fragments but to develop complementary features. As will be seen, in many cases the boundary alone performs as well as or better than the appearance and segmentation masks (mattes) used by other authors (e.g. [19, 27]) – the boundary is responsible for much of the success.

The areas of novelty in the paper include: (i) the manner in which the boundary-fragment codebook is learnt – fragments (from the boundaries of the training objects) are selected to be highly class-distinctive, *and* are stable in their prediction of the object centroid; and (ii) the construction of a strong *detector* (rather than a classifier) by Boosting [15] over a set of weak detectors built on boundary fragments. This detector means that it is not necessary to scan the image with a sliding window in order to localize the object.

Boundaries have been used in object recognition to a certain extent: Kumar *et al.* [17] used part outlines in their application of pictorial structures [11]; Fergus *et al.* [13] used boundary curves between bitangent points in their extension of the constellation model; and, Jurie and Schmid [16] detected circular arc features from boundary curves. However, in all these cases the boundary features are segmented independently in individual images. They are not flexibly selected

Fig. 2. Example training images for the cows category

Fig. 3. Examples of detecting multiple objects in one test image

to be discriminative over a training set, as they are here. Bernstein and Amit [4] do use discriminative edge maps. However, theirs is only a very local representation of the boundary; in contrast we capture the global geometry of the object category. Recently, and independently, Shotton *et al.* [24] presented a method quite related to the Boundary-Fragment-Model presented here. The principal differences are: the level of segmentation required in training ([24] requires more); the number of boundary fragments employed in each weak detector (a single fragment in [24], and a variable number here); and the method of localizing the detected centroid (grid in [24], mean shift here).

We will illustrate BFM classification and detection for a running example, namely the object class cows. For this we selected cow images as [7, 19] which originate from the videos of Magee and Boyle [21]. The cows appear at various positions in the image with just moderate scale changes. Figure 2 shows some example images. Figure 3 shows detections using the BFM detector on additional, more complex, cow images obtained from Google image search.

2 Learning Boundary Fragments

In a similar manner to [19], we require the following data to train the model:

- A training image set with the object delineated by a bounding box.
- A validation image set labelled with whether the object is absent or present, and the object's centroid (but the bounding box is not necessary).

The training images provide the candidate boundary fragments, and these candidates are optimized over the validation set as described below. For the results of this section the training set contains 20 images of cows, and the validation set contains 25 cow images (the positive set) and 25 images of other objects (motorbikes and cars – the negative set).

Given the outlines of the training images we want to identify boundary fragments that:

(i) discriminate objects of the target category from other objects, and
(ii) give a precise estimate of the object centroid.

A candidate boundary fragment is required to (i) match edge chains often in the positive images but not in the negative, *and* (ii) have a good localization of the centroid in the positive images. These requirements are illustrated in figure 4. The idea of using validation images for discriminative learning is motivated by

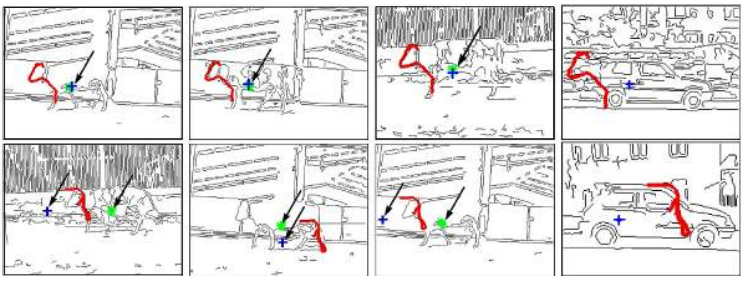

Fig. 4. Scoring boundary fragments. The first row shows an example of a boundary fragment that matches often on the positive images of the validation set, and less often on the negative images. *Additionally* it gives a good estimate of the centroid position on the positive images. In contrast, the second row shows an example of an unsuitable boundary fragment. The cross denotes the estimate of the centroid and the asterisk the correct object centroid.

Sali and Ullman [23]. However, in their work they only consider requirement (i), the learning of class-discriminate parts, but not the second requirement which is a geometric relation. In the following we first explain how to score a boundary fragment according to how well it satisfies these two requirements, and then how this score is used to select candidate fragments from the training images.

2.1 Scoring a Boundary Fragment

Linked edges are obtained in the training and validation set using a Canny edge detector with hysteresis. We do not obtain perfect segmentations – there may be gaps and false edges. A linked edge in the training image is then considered as a candidate boundary fragment γ_i, and scoring cost $C(\gamma_i)$ is a product of two factors:

1. $c_{match}(\gamma_i)$: the matching cost of the fragment to the edge chains in the validation images using a Chamfer distance [5, 6], see (1). This is described in more detail below.
2. $c_{loc}(\gamma_i)$: the distance (in pixels) between the true object centroid and the centroid predicted by the boundary fragment γ_i averaged over all the positive validation images.

with $C(\gamma_i) = c_{match}(\gamma_i)c_{loc}(\gamma_i)$. The matching cost is computed as

$$c_{match}(\gamma_i) = \frac{\sum_{i=1}^{L^+} distance(\gamma_i, P_{v_i})/L^+}{\sum_{i=1}^{L^-} distance(\gamma_i, N_{v_i})/L^-} \tag{1}$$

where L^- denotes the number of negative validation images N_{v_i} and L^+ the number of positive validation images P_{v_i}, and $distance(\gamma_i, I_{v_i})$ is the distance to the best matching edge chain in image I_{v_i}:

$$distance(\gamma_i, I_{v_i}) = \frac{1}{|\gamma_i|} \min_{\gamma_i \subset I_{v_i}} \sum_{t \in \gamma_i} DT_{I_{v_i}}(t) \tag{2}$$

where $DT_{I_{v_i}}$ is the distance transform. The Chamfer distance [5] is implemented using 8 orientation planes with an overlap of 5 degrees. The orientation of the edges is averaged over a length of 7 pixels by orthogonal regression. Because of background clutter the best match is often located on highly textured background clutter, i.e. it is not correct. To solve this problem we use the $N = 10$ best matches (with respect to (2)), and from these we take the one with the best centroid prediction. Note, images are scale normalized for training.

2.2 Selecting Boundary Fragments

Having defined the cost, we now turn to selecting candidate fragments. This is accomplished by optimization. For this purpose seeds are randomly distributed on the boundary of each training image. Then at each seed we extract boundary fragments. We let the size of each fragment grow and at every step we calculate the cost $C(\gamma_i)$ on the validation set. Figure 5(a) shows three examples of this growing of boundary fragments (the length varies from 20 pixels in steps of 30 pixels in both directions up to a length of 520 pixels). The cost is minimized over the varying length of the boundary fragment to choose the best fragment. If no length variation meets some threshold of the cost we reject this fragment and proceed with the next one. Using this procedure we obtain a codebook of boundary fragments each having the geometric information to vote for an object centroid.

To reduce redundancy in the codebook the resulting boundary fragment set is merged using agglomerative clustering on medoids. The distance function is $distance(\gamma_i, \gamma_j)$ (where I_{v_i} in (2) is replaced by the binary image of fragment γ_j) and we cluster with a threshold of $th_{cl} = 0.2$. Figure 5(b) shows some examples of resulting clusters. This optimized codebook forms the basis for the next stage in learning the BFM.

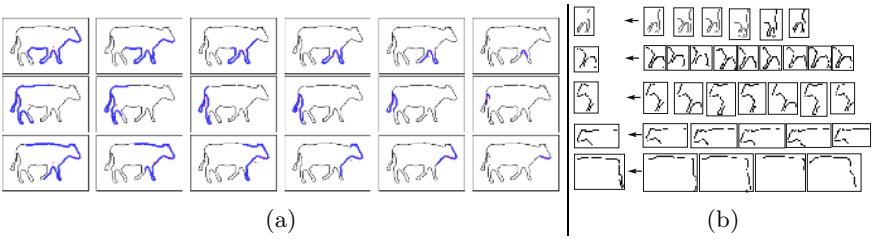

Fig. 5. Learning boundary fragments. (a) Each row shows the growing of a different random seed on a training image. (b) Clusters from the optimized boundary fragments. The first column shows the chosen codebook entries. The remaining columns show the boundary fragments that also lie in that cluster.

3 Training an Object Detector Using Boosting

At this stage we have a codebook of optimized boundary fragments each carrying additional geometric information on the object centroid. We now want to combine these fragments so that their aggregated estimates determine the centroid and increase the matching precision. In the case of image fragments, a single region can be used to determine a unique correspondence (e.g. see [19]). In contrast, boundary fragments are not so discriminating, but a combination of several such fragments (for example distributed around the actual object boundary) is characteristic for an object class.

We combine boundary fragments to form a weak detector by learning combinations which fit well on all the positive validation images. We then learn a strong detector from these weak detectors using a standard Boosting framework which is adapted to learn detection rather than classification. This learning of a strong detector chooses boundary fragments which model the whole distribution of the training data (whereas the method of the previous section can score fragments highly if they have low costs on only a subset of the validation images).

3.1 Weak Detectors

A weak detector is composed of k (typically 2 or 3) boundary fragments. We want a detector to fire ($h_i(I) = 1$) if (i) the k boundary fragments match image edge chains, (ii) the centroid estimates concur, and, in the case of positive images, (iii) the centroid estimate agrees with the true object centroid. Figure 6(a) illustrates a positive detection of an image (with $k = 2$ and the boundary fragments named γ_a and γ_b). The classification output $h_i(I)$ of detector h_i on an image I is defined as:

$$h_i(I) = \begin{cases} 1 & \text{if } D(h_i, I) < th_{h_i} \\ 0 & \text{otherwise} \end{cases}$$

with th_{h_i} the learnt threshold of each detector (see section 3.2), and where the distance $D(h_i, I)$ of h_i (consisting of k boundary fragments γ_j) to an image I is defined as:

$$D(h_i, I) = \frac{1}{m_s^2} \cdot \sum_{j=1}^{k} distance(\gamma_j, I) \tag{3}$$

The $distance(\gamma_j, I)$ is defined in (2) and m_s is explained below. Any weak detector where the centroid estimate misses the true object centroid by more than d_c (in our case 15 pixels), is rejected.

Figure 6(b) shows examples of matches of weak detectors on positive and negative validation images. At these positions as shown in column 2 of figure 6(a) each fragment also estimates a centroid by a circular uncertainty window. Here the radius of the window is $r = 10$. The compactness of the centroid estimate is measured by m_s (shown in the third column of figure 6(a)). $m_s = k$ if the circular uncertainty regions overlap, and otherwise a penalty of $m_s = 0.5$ is allocated. Note, to keep the search for weak detectors tractable, the number

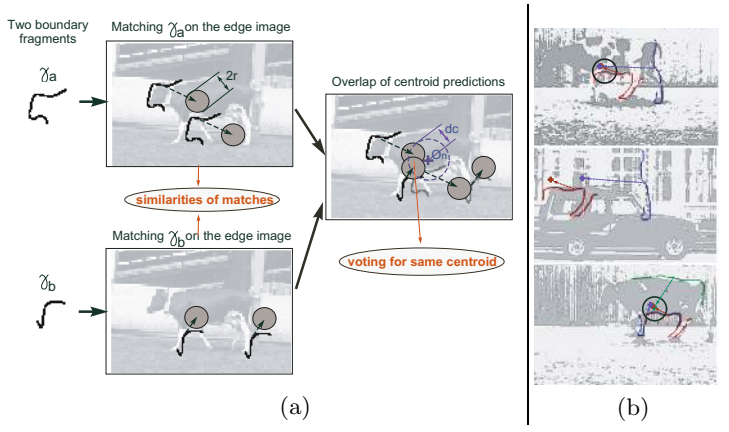

Fig. 6. Learning a weak detector. (a) The combination of boundary fragments to form a weak detector. Details in the text. (b) Examples of matching the weak detector to the validation set. Top: a weak detector with $k = 2$, that fires on a positive validation image because of highly compact centre votes close enough to the true object centre (black circle). Middle: a negative validation image where the same weak detector does not fire (votings do not concur). Bottom: the same as the top with $k = 3$. In the implementation $r = 10$ and $d_c = 15$.

of used codebook entries (before clustering, to reduce the effort already in the clustering procedure) is restricted to the top 500 for $k = 2$ and 200 for $k = 3$ (determined by the ranked costs $C(\gamma_i)$). Also, each boundary fragment is matched separately and only those for which $distance(\gamma_j, I) < 0.2$ are used.

3.2 Strong Detector

Having defined a weak detector consisting of k boundary fragments and a threshold th_{h_i}, we now explain how we learn this threshold and form a strong detector H out of T weak detectors h_i using AdaBoost. First we calculate the distances $D(h_i, I_j)$ of all combinations of our boundary fragments (using k elements for one combination) on all (positive and negative) images of our validation set $I_1 \dots I_v$. Then in each iteration $1 \dots T$ we search for the weak detector that obtains the best detection result on the current image weighting (for details see AdaBoost [15]). This selects weak detectors which generally (depending on the weighting) "fire" often on positive validation images (classify them as correct and estimate a centroid closer than d_c to the true object centroid) and not on the negative ones. Figure 7 shows examples of learnt weak detectors that contribute to the strong detector. Each of these weak detectors also has a weight w_{h_i}. The output of a strong detector on a whole test image is then:

$$H(I) = sign(\sum_{i=1}^{T} h_i(I) \cdot w_{h_i}). \quad (4)$$

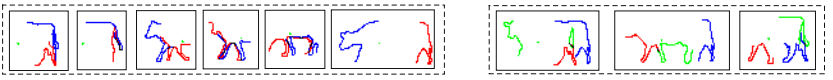

Fig. 7. Examples of weak detectors, left for $k = 2$, and right for $k = 3$

The sign function is replaced in the detection procedure by a threshold t_{det}, where an object is detected in the image I if $H(I) > t_{det}$ and no evidence for the occurrence of an object if $H(I) \leq t_{det}$ (the standard formulation uses $t_{det} = 0$).

4 Object Detection

Detection algorithm and segmentation: The steps of the detection algorithm are now described and qualitatively illustrated in figure 8. First the edges are detected (step 1) then the boundary fragments of the weak detectors, that form the strong detector, are matched to this edge image (step 2). In order to detect (one or more) instances of the object (instead of classifying the whole image) each weak detector h_i votes with a weight w_{h_i} in a Hough voting space (step 3). Votes are then accumulated in a circular search window ($W(x_n)$) with radius d_c around candidate points x_n (represented by a Mean-Shift-Mode estimation [8]). The Mean-Shift modes that are above a threshold t_{det} are taken as detections of object instances (candidate points). The confidence in detections at these candidate points x_n is calculated using probabilistic scoring (see below). The segmentation is obtained by backprojection of the boundary fragments (step 3) of weak detectors which contributed to that centre to a binary pixel map. Typically, the contour of the object is over-represented by these fragments. We obtain a closed contour of the object, and additional, spurious contours (seen in figure 8, step 3). Short segments (< 30 pixels) are deleted, the contour is filled (using Matlab's 'filled area' in regionprops), and the final segmentation matte is obtained by a morphological opening, which removes thin structures (votes from

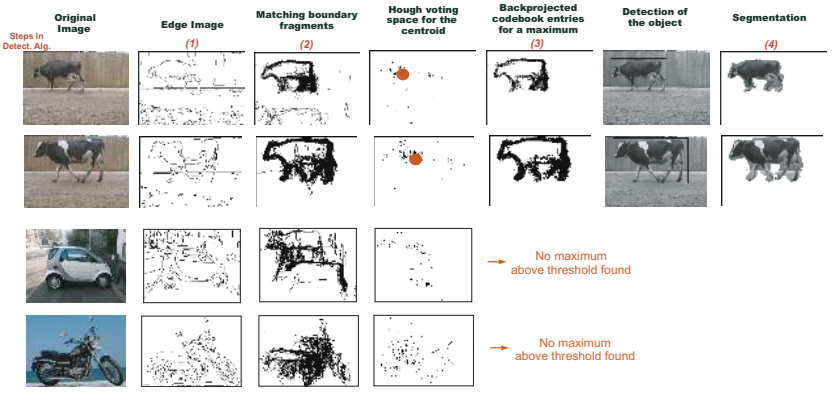

Fig. 8. Examples of processing test images with the BFM detector

outliers that are connected to the object). Finally, each of the objects obtained by this procedure is represented by its bounding box.

Probabilistic scoring: At candidate points x_n for instances of an object category c, found by the strong detector in the test image I_T we sum up the (probabilistic) votings of the weak detectors h_i in a 2D Hough voting space which gives us the probabilistic confidence:

$$conf(x_n) = \sum_{i}^{T} p(c, h_i) = \sum_{i}^{T} p(h_i)p(c|h_i) \tag{5}$$

where $p(h_i) = \frac{1}{\sum_{q=1}^{M} score(h_q, I_T)} \cdot score(h_i, I_T)$ describes the pdf of the effective matching of the weak detector with $score(h_i, I_T) = 1/D(h_i, I_T)$ (see (3)). The second term of this vote is the confidence we have in each specific weak detector and is computed as:

$$p(c|h_i) = \frac{\#fires_{correct}}{\#fires_{total}} \tag{6}$$

where $\#fires_{correct}$ is the number of positive and $\#fires_{total}$ is the number of positive and negative validation images the weak detector fires on. Finally our confidence of an object appearing at position x_n is computed by using a Mean-Shift algorithm [8] (circular window $W(x_n)$) in the Hough voting space defined as: $conf(x_n|W(x_n)) = \sum_{X_j \in W(x_n)} conf(X_j)$.

5 Detection Results

In this section we compare the performance of the BFM detector to published state-of-the-art results, and also give results on new data sets. Throughout we use fixed parameters ($T = 200$, $k = 2$, $t_{det} = 8$) for our training and testing procedure unless stated otherwise. An object is deemed correctly detected if the overlap of the bounding boxes (detection vs ground truth) is greater than 50%.

Cows: First we give quantitative results on the cow dataset. We used 20 training images (validation set 25 positive/25 negative) and tested on 80 unseen images, half belonging to the category cows and half to counter examples (cars and motorbikes). In table 2 we compare our results to those reported by Leibe *et al.* [19] and Caputo *et al.* [7] (Images are from the same test set – though the authors do not specify which ones they used). We perform as well as the result in [19], clearly demonstrating that in some cases the contour alone is sufficient for excellent detection performance. Kumar *et al.* [17] also give an RPC curve for cow detection with an ROC-equal-error rate of 10% (though they use different test images). Note, that the detector can identify multiple instances in an image, as shown in figure 3.

Variation in performance with number of training images: The results on the cow dataset reported above have been achieved using 20 training images. Figure 9 shows how the number of training images influences the performance of

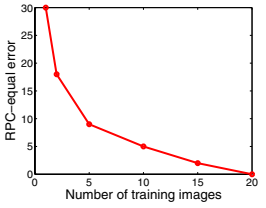

Fig. 9. Error depending on the number of training images for the cow dataset

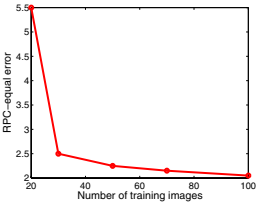

Fig. 10. Error depending on the number of training images for Cars-Rear

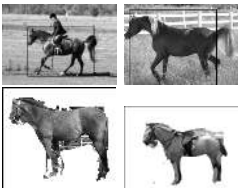

Fig. 11. Example of BFM detections for horses showing computed bounding boxes and segmentations

the BFM detector. Even with five images our model achieves detection results of better than 10% RPC-equal-error rate. The performance saturates at twenty in this case, but this number is dependent on the degree of within class variation (e.g. see fig.10).

Caltech datasets: From the widely used Caltech datasets we performed experiments on the category Cars-Rear and Airplanes. Table 1 shows our results compared with other state of the art approaches on the same test images as reported in [12]. First we give the detection results (BFM-D) and compare them to the best (as far as we know) results on detection by Leibe *et. al* [18, 19, 20] (scale changes are handled as described in section 6). We achieve superior results – even though we only require the bounding boxes in the training images (and not foreground segmentation as in [24], for example). For the classification results an image is classified, in the manner of [12], if it contains the object, but localization by a bounding box is not considered. Compared to recently published results on this data we again achieve the best results. Note that the amount of supervision varies over the methods where e.g. [26] use labels and bounding boxes (as we do); [2, 3, 12, 22] use just the object labels; and Sivic *et al.* [25] use no supervision. It should be pointed out, that we use just 50 training images and 50 validation images for each category, which is less than the other approaches use. Figure 10 shows the error rate depending on the number of training images (again, the same number of positive and negative validation images are used). However, it is known that the Caltech images are now not sufficiently demanding, so we next consider further harder situations.

Table 1. Comparison of the BFM detector to other published results on the Caltech dataset (Cars-Rear and Airplanes). The first two columns give the actual object detection error (BFM-D) and the remaining columns the categorization of the images (BFM-C) given by the ROC-equal error rates.

Cat.	BFM-D	[18]	BFM-C	[12]	[22]	[25]	[2]	[3]	[14]	[26]	[28]
Cars-Rear	**2.25**	6.1	**0.05**	8.8	8.9	21.4	3.1	2.3	1.8	9.8	-
Airplanes	**7.4**	-	**2.6**	6.3	11.1	3.4	4.5	10.3	-	17.1	5.6

Table 2. Comparison of the BFM detector to other published results on the cows

Method	RPC-err.
Caputo *et al.* [7]	2.9%
Leibe *et al.* [19]	0.0%
Our approach	0.0%

Table 3. The first 3 rows show the failures made by the three different models (FP=false positive, FN=false negative, M=multiple detection). The last row shows the RPC-equal-error rate for each model.

-	cow	horse1	horse2
FP	0	3	0
FN	0	13	12
M	0	1	2
RPC-err	0%	23%	19%

Horses and Cow/horse discrimination: To address the topic of how well our method performs on categories that consist of objects that have a similar boundary shape we attempt to detect and discriminate horses and cows. We use the horse data from [16] (no quantitative comparison as the authors could not report their exact test set because of lost data). In the following we compare three models. In each case they are learnt on 20 training images of the category and a validation set of 25 positive and 25 negative images that is different for each model. The first model for cows (cow-BFM) is learnt using no horses in the negative validation set (13 cars, 12 motorbikes). The second model for horses (horse1-BFM) is learnt using also cows in the negative validation set (8 cars, 10 cows, 7 motorbikes). Finally we train a model (horse2-BFM) which uses just cow images as negative validation images (25 cows). We now apply all three models on the same test set, containing 40 images of cows and 40 images of horses (figure 11 shows example detection results). Table 3 shows the failures and the RPC-equal error rate of each of these three models on this test set. The cow model is very strong (no failures) because it needs no knowledge of another object class even if its boundary shape is similar. Horse1-BFM is a weaker model (this is a consequence of greater variations of the horses in the training and test images). The model horse2-BFM obviously gains from the cows in the negative validation images, as it does not have any false positive detections. Overall this means our models are good at discriminating classes of similar boundary shapes.

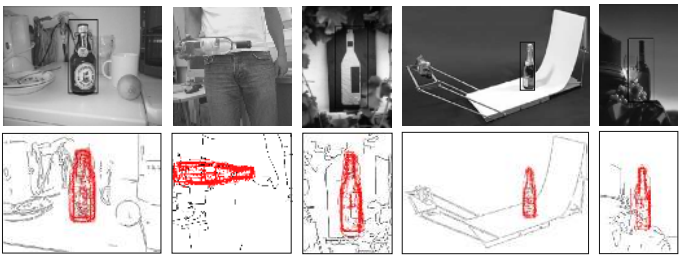

Fig. 12. Example of BFM detections for bottles. The first row shows the bounding box of the detection and the second row shows the backprojected boundary fragments for these detections.

Still, categories with higher intra-class variability (like horses compared to cows) are harder to learn and might need more training data to generalize over the whole distribution.

Bottles: To show the advantage of an approach relying on the shape of an object category we set up a new dataset of bottle images. This consists of 118 images collected using Google Image Search. Negative images are provided by the Caltech background image set. We separated the images in test/training/validation-set (64/24/30) and added the same amount of negative images in each case. We achieve an RPC-equal error rate of 9%. Figure 12 shows some detection examples.

6 Invariance to Scale, Rotation and Viewpoint

This section briefly discusses the topic of invariance of the BFM with respect to scale, rotation and changes in viewpoint.

Search over scale: A scaled codebook representation is used. Additionally we normalize the parameters in the detection procedure with respect to scale, for example the radius for centroid estimation, in the obvious way. The Mean-Shift modes are then aggregated over the set of scales, and the maxima explored as in the single scale case. Results on Cars-rear, airplanes and bottles of section 5 were obtained by this method.

Rotation: To achieve in-plane rotation invariance we use rotated versions of the codebook (see figure 12 second column for an example). The BFM is invariant to small rotations in plane due to the orientation planes used in the Chamfer-matching. This is a consequence of the nature of our matching procedure. For many categories the rotation invariance up to this degree may be sufficient (e.g. cars, cows) because they have a favoured orientation where other occurrences are quite unnatural.

Changes in viewpoint: For natural objects (e.g. cows) the perceived boundary is the visual rim. The position of the visual rim on the object will vary with pose but the shape of the associated boundary fragment will be valid over a range of poses. We performed experiments under controlled conditions on the ETH-80 database. With a BFM learnt for a certain aspect we could still detect a prominent mode in the Hough voting space up to 45 degrees rotation in both directions (horizontal and vertical). Thus, to extend the BFM to various aspects this invariance to small viewpoint changes reduces the number of necessary positions on the view-sphere to a handful of aspects that have to be trained separately. Our probabilistic formulation can be straightforwardly extended to multiple aspects.

7 Discussion and Conclusions

We have described a Boundary Fragment Model for detecting instances of object categories. The method is able to deal with the partial boundaries that typically

are recovered by an edge detector. Its performance is similar to or outperforms state-of-the-art methods that include image appearance region fragments. For classes where the texture is very variable (e.g. bottles, mugs) a BFM may be preferable. In other cases a combination of appearance and boundary will have superior performance.

It is worth noting that the BFM once learnt can be implemented very efficiently using the low computational complexity method of Felzenszwalb & Huttenlocher[11].

Currently our research is focusing on extending the BFM to multi-class and multiple aspects of one class.

Acknowledgements

This work was supported by the Austrian Science Foundation FWF, project S9103-N04, ECVision and Pascal Network of Excellence.

References

1. S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE PAMI*, 26(11):1475–1490, Nov. 2004.
2. J. Amores, N. Sebe, and P. Radeva. Fast spatial pattern discovery integrating boosting with constellations of contextual descriptors. In *Proc. CVPR*, volume 2, pages 769–774, CA, USA, June 2005.
3. A. Bar-Hillel, T. Hertz, and D. Weinshall. Object class recognition by boosting a part-based model. In *Proc. CVPR*, volume 2, pages 702–709, June 2005.
4. E. J. Bernstein and Y. Amit. Part-based statistical models for object classification and detection. In *Proc. CVPR*, volume 2, pages 734–740, 2005.
5. G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE PAMI*, 10(6):849–865, 1988.
6. H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time Euclidean distance transform algorithms. *IEEE PAMI*, 17(5):529–533, May. 1995.
7. B. Caputo, C. Wallraven, and M. Nilsback. Object categorization via local kernels. In *Proc. ICPR*, pages 132–135, 2004.
8. D. Comaniciu and P. Meer. Mean shift: A robust approach towards feature space analysis. In *IEEE PAMI*, volume 24(5), pages 603–619, 2002.
9. D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Proc. CVPR*, pages 10–17, 2005.
10. G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *ECCV04. Workshop on Stat. Learning in Computer Vision*, pages 59–74, 2004.
11. P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *Intl. Journal of Computer Vision*, 61(1):55–79, 2004.
12. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, pages 264–271, 2003.
13. R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. ECCV*, pages 242–256, 2004.

14. R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. Proc. CVPR*, 2005.
15. Y. Freund and R. Schapire. A decision theoretic generalisation of online learning. *Computer and System Sciences*, 55(1):119–139, 1997.
16. F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *Proc. of CVPR*, pages 90–96, 2004.
17. M. Kumar, P. Torr, and A. Zisserman. Extending pictural structures for object recognition. In *Proc. BMVC*, 2004.
18. B. Leibe. *Interleaved Object Categorization and Segmentation.* PhD thesis, Swiss Federal Institute of Technology, 2004.
19. B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV04. Workshop on Stat. Learning in Computer Vision*, pages 17–32, May 2004.
20. B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive mean-shift search. In *DAGM'04*, pages 145–153, Aug. 2004.
21. D. Magee and R. Boyle. Detecting lameness using re-sampling condensation and multi-steam cyclic hidden markov models. *Image and Vision Computing*, 20(8):581–594, 2002.
22. A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. ECCV*, pages 71–84, 2004.
23. E. Sali and S. Ullman. Combining class-specific fragments for object classification. In *Proc. BMVC*, pages 203–213, 1999.
24. J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *Proc. ICCV*, pages 503–510, 2005.
25. J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Proc. ICCV*, 2005.
26. J. Thureson and S. Carlsson. Appearance based qualitative image description for object class recognition. In *Proc. ECCV*, pages 518–529, 2004.
27. M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *Proc. ICCV*, volume 1, pages 281–288, 2003.
28. W. Zhang, B. Yu, G. Zelinsky, and D. Samaras. Object class recognition using multiple layer boosting with heterogenous features. In *Proc. CVPR*, pages 323–330, 2005.