Fusing shape and appearance information for object category detection

Andreas Opelt, Axel Pinz
Graz University of Technology, Austria

Andrew Zisserman
Dept. of Engineering Science, University of Oxford, UK

Abstract

We present a method which is able to combine various feature types (e.g. image patches and edge boundaries) to learn models for object categories. Our objective is to detect object instances in an image, as opposed to the easier task of image categorization.

We investigate two algorithms for learning and detecting object categories. Both algorithms benefit from combining features. The first uses a naive combination method for detectors each employing only one type of feature, the second learns the best features (from a pool of patches and boundaries).

In experiments we achieve comparable results to the state of the art over a number of datasets, and for some categories we even achieve the lowest errors that have been reported so far. The results also show that certain object categories prefer certain feature types (e.g. boundary fragments for airplanes).

1 Introduction

Much of the recent research into object category recognition has developed models focussed on one type of feature – either appearance patches or edge fragments [1, 4, 5, 7, 8, 12, 14, 17, 20]. This is not ideal as some classes cannot be distinguished by one feature type alone (e.g. discriminating between zebras and horses just by their shape), and more generally it loses the possibility of using feature types which are particularly discriminating for a category.

In this paper we investigate two algorithms which are variations on methods of combining different types of features. In each case the algorithms: (i) learn the best fitting features, (ii) use complementary features, and (iii) are able to *detect* objects instead of categorizing images. The first algorithm (named "CM" for "Combined-Models") combines models for different feature types, and the second (called "SF" for "Selected-Features") learns a single model using a mixture of the available feature types. As a foundation model for the two algorithms we choose the Boundary-Fragment-Model (BFM) from Opelt *et al.* [17], though other similar models would work equally well, for example [3, 14, 20].

The benefits of using different types of features (detectors or descriptors) is well illustrated by the evolution of the Implicit Shape Model of Leibe *et al.* [14]. In its original

form only one type of feature (patches around interest points) was used, but it has now been extended to other types of features (shape context, SIFT) by Seemann *et al.* [19] with a corresponding increase in performance. Mixed/complementary feature types have been used previously [9, 16, 23, 22], though, for the most part, these have been used for image classification rather than detection. For example, Opelt *et al.* [16] presented an algorithm which learns suitable category descriptors from a pool of different types of descriptors for appearance regions, and Zhang *et al.* [23], used complementary descriptors (PCA-SIFT and shape context). Fergus *et al.* [10] investigated detection with mixed types of features and this is the most similar to our work in terms of the used features (regions and edge boundaries), however their algorithm does not learn which features to use.

2 An Overview...

...of the basic model from Opelt *et al.*[17]: As mentioned we use the Boundary Fragment Model from [17]. This is essentially a combination of the geometric model from [14] with a discriminative codebook influenced by the idea of Sali and Ullman [18]. The BFM consists of a set of curve fragments representing the edges of the object, both internal and external (silhouette), with additional geometric information about the object centroid (in the manner of [14]). A BFM is learnt in two stages. First, random boundary fragments are extracted from the training images. Then costs are calculated for each fragment on a validation set. Low costs are achieved for boundary fragments that match well on the positive validation images, not so well on the negative ones, and have good centroid predictions on the positive validation images. Second, combinations of $k = 2$ boundary fragments are learnt as weak detectors (not just classifiers) within an AdaBoost [11] framework. Note that this learning procedure does not need pre-segmented training data as in similar methods (e.g. [14]), but works with the given bounding boxes. Detecting instances of the object category in a new test image is done by applying the weak detectors and collecting their votes in a Hough voting space. An object is detected if a mode (obtained using Mean-Shift mode estimation) is above a detection threshold. Following the detection, boundary fragments that contributed to that mode are back-projected into the test image and provide an object segmentation. An overview of the detection method is shown in figure 1. Invariance to translation is given by the nature of the mode search in the Hough space. Scale invariance is achieved by using re-scaled versions of the model. Each model votes in a separate Hough space and then maxima are searched over these Hough spaces.

...of our Combined-Model (CM) algorithm: In figure 2 we show the general idea of the combination of two feature types (appearance regions and fragments of the boundary). We train two separate models. First, as shown at the top of figure 2, we train a geometric model using boundary fragments as features as proposed by Opelt *et al.* [17]. Second, as shown at bottom of figure 2, we use the same learning method to obtain a model which is based on a different feature type, namely intensity regions around salient points. The salient points are extracted on the same training and validation data as for the model which is based on shape features. Then, these region based features are described by vectors which can be obtained by various description techniques (e.g. SIFT [15]). Evaluating these feature vectors on the validation set results in a discriminative codebook of regions.

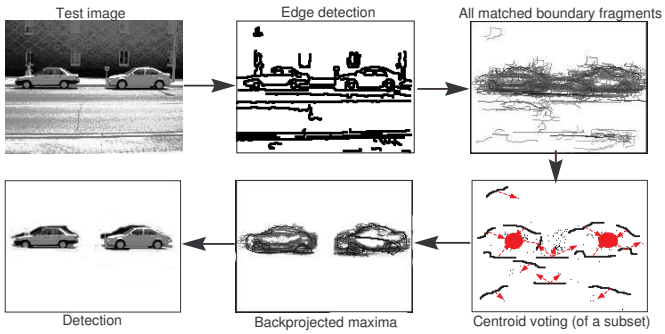

Figure 1: Overview of the BFM from Opelt *et al.* [17].

The final strong detector is trained using that discriminative codebook and, again, the validation set. The part shown in the centre of figure 2 illustrates how detection can be improved using these two models learnt from complementary features: A new test image is put into the detection procedure of each model and then we form a linear combination of the resulting probabilistic Hough voting spaces. The influence of each model can be controlled by a weight vector $\mathbf{f}$ (each entry f_τ of the vector contains the weight for a certain model of feature type τ). Finally the decision on whether there is enough evidence for the appearance of an object is made on that combined voting space.

. . . of our Selected-Features (SF) algorithm: The combination of different feature types as proposed above increases the number of categories for which this approach is suitable. However, training a new model for each type of feature for each category often requires needless effort. A suitable algorithm should not have to train a separate model for each feature type but select on its own what types of features are suitable for that category. Figure 3 illustrates the general idea. In the same manner as above a discriminative codebook is learnt *jointly* for all available feature types. Then the second stage learns a strong detector from weak detectors using features from this codebook.

3 The combined model (CM)

This algorithm sets out to explore how the evidence of detection of two models which were learnt using different types of features can be combined. First we need to explain how the model from [17] was extended to regions around salient points which are described by a feature vector. Subsequently, we explain how the models are combined.

Scoring of regions: We extract F_i features from each grayscale image I_i (we do not use colour information). In our evaluation we use the Harris-Laplace combined with the Hessian-Laplace implementation from [13]. But our implementation is flexible and could use other techniques, e.g. Affine-Harris, as well. The regions are scale normalized and described by a SIFT descriptor [15]. Here we use the binary from [13] generally enabling our framework to use a variety of different techniques (e.g. Gloh, PCA-Sift,

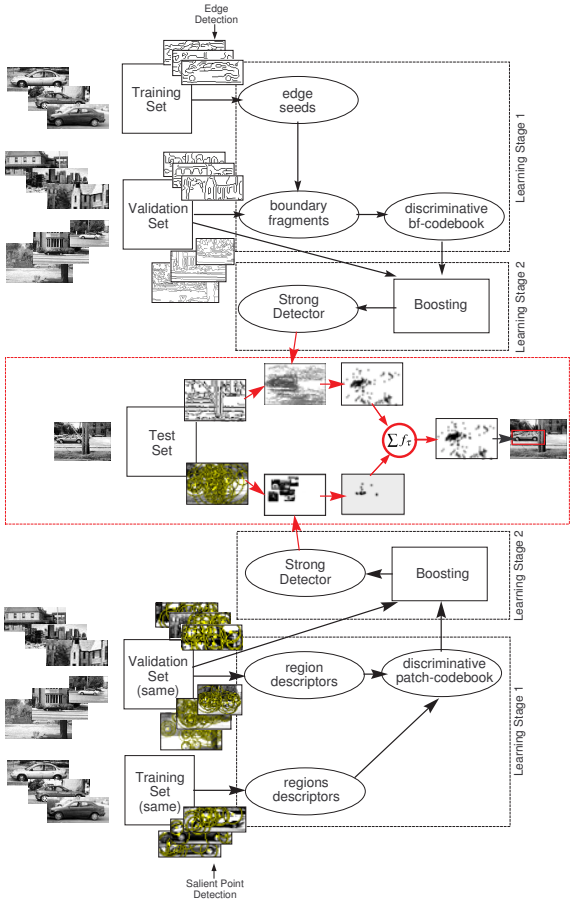

Figure 2: **Overview of the CM algorithm:** Combining models trained using different types of features. Top: a shape based model is trained (for cars-side). Bottom: training a model based on descriptors of local regions. In the centre (surrounded by the red dashed box) it is shown how these two models are then used to detect objects in new images.

shape context). Hence, each image I_i (training images and positive and negative validation images) is represented by a set of feature vectors v_j with $j = 1 \ldots F_i$.

Then we calculate costs on a validation set. Low costs are achieved for patches that match well in their appearance on the positive validation images, match poorly on the negative validation images *and* give a good average centroid prediction (in the manner of [17]). As a distance function between feature vectors describing regions we use:

$$distance(v_i, I_p) = \min_{j=1\ldots F_p} \sqrt{\sum_{q=1}^{|v_i|} (v_i(q) - v_j(q))^2} \qquad (1)$$

denoting the minimum Euclidean distance of a certain feature vector v_i to all feature

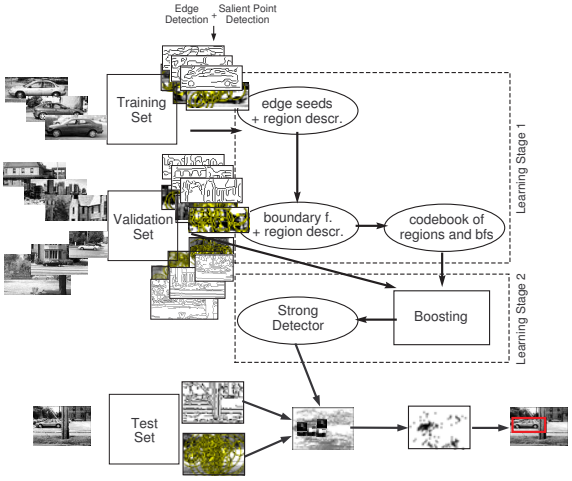

Figure 3: **Overview of the SF algorithm:** In the training procedure all feature types are used (here we use regions and boundary fragments). Boosting selects a combination of weak detectors of different type to form a suitable strong detector for a category.

vectors of a certain validation image I_p. With that distance function the costs are calculated in the same manner as in [17]. To be robust to matches in background clutter we use the 10 best matches (with *distance* below a threshold t_M, 500 in our implementation) for the calculation of the matching costs (the lowest costs are taken). Thus, we have calculated costs for each feature vector on the validation set.

Learning an incremental codebook of regions: Each codebook entry is formed by a feature vector describing a region. Additionally each entry contains location information for centroid(s), and the costs it achieved on the validation set. The distance between entries is here just the Euclidean distance for the feature vectors of those entries. Codebook entries are clustered using a threshold on the costs θ_C (250 in our implementation).

Figure 4 shows examples of entries from an alphabet incrementally learnt on the category cars-side (UIUC). Note that patches might come from different positions in the training images (e.g. front wheel and back wheel of the cars) and thus clustering updates the geometric information of each alphabet entry (if necessary).

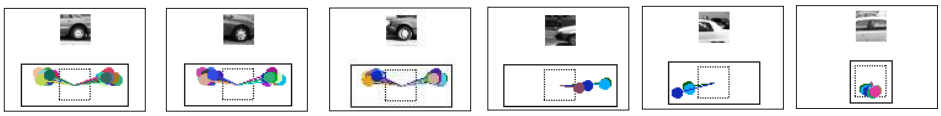

Figure 4: **Examples of a region based alphabet**. Each entry shows one element of the alphabet learnt for the category cars-side from the UIUC database. One entry consists of a scale normalized region (on the top) and the geometric information (blobs denoting the centroid vectors) shown on the bottom of each entry.

A strong detector of regions: Matching single regions separately is often sidetracked by matches in the background clutter. Hence, we build weak detectors of combinations of $k = 2$ regions. A weak detector is valid if both regions match and agree in their prediction for the objects centroid (with an uncertainty distance $d_c = 10$ pixels). Additionally this prediction of the centroid has to be within a radius of $r = 15$ pixels of the true object centroid of the positive validation images. This is done in the same manner for the boundary fragments and for the regions where the search for matches is restricted to the detected interest points.

We use AdaBoost [11] to select a number of possible candidates for weak detectors by iteratively choosing the one with the best detection performance on the current weighting of the validation images. Figure 5 shows some of the weak detectors learnt to form the final strong detector based on region descriptions.

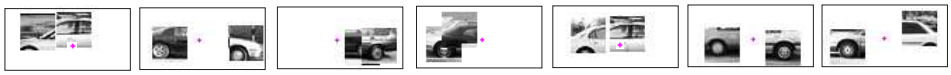

Figure 5: **Examples of weak detectors using regions** learnt from the alphabet of cars-side (UIUC). Obviously but notably the wheels appear as the strongest cue using appearance based regions for this category. The centroid is denoted by the red cross.

We have now learnt a region-based strong detector and additionally we learn a shape based BFM (in the manner of [17]). Thus, complementary information is available for the subsequent detection procedure.

Detection using regions and shape, separately: Detection is performed with each model learnt for a special type of information τ separately. In our case this is one model for shape and one for regions. This results in two voting spaces with the votes of the corresponding weak detectors. As recently shown in [21] this can be seen as probability distributions of the sums of the weak detectors beliefs in object evidences in the image. The evidence from various models can simply be combined by fusing these probabilistic voting spaces. Given Q different models, the confidence for an object appearing at position x_n can be defined as:

$$conf(x_n) = \frac{\sum_{\tau \in Q} f_\tau \sum_i^T p(c, h_i^\tau)}{\sum_\tau f_\tau} \tag{2}$$

with h_i^τ denoting a certain weak detector i of information type τ, c denotes a certain category, T the numbers of weak detectors in a model, and f_τ the weight of each model in the combination. The object instances then correspond to modes in this combined space, and these are obtained using Mean-Shift-Mode estimation [6].

4 The selective algorithm (SF)

This algorithm combines various feature types in one model. The learning algorithm selects weak detectors of different feature types from a discriminative codebook which contains a mixed pool of features.

Learning the mixed codebook: Learning a mixed codebook is done in a similar manner to learning a discriminative codebook for just one type of features. Sequentially each feature type is processed, costs on the validation set are calculated, features with low costs are selected and then similar codebook entries are merged. Because of the differences in the distances and feature dimensions we need two thresholds for each feature type τ. First th_K^τ sets a threshold on the costs where features with lower costs are selected. A second threshold th_C^τ is used for merging similar codebook entries. Small variations in those thresholds do not have much impact. Still the rough selection of meaningful thresholds for reach feature type is obviously crucial to achieve good results. This mixed codebook consists now of entries where each entry has a feature type, a description and localization information for the object centroid.

Learning the mixed strong detectors: As a basis for our Boosting procedure we form combinations of $k = 2$ elements of the same feature type from the mixed codebook. Weak detectors are built in the manner of [17] independent of the feature type. From that pool of possible weak detectors AdaBoost [11] is used to learn a set of T weak detectors h_i^τ. Figure 6 illustrates the first 10 weak detectors of such mixed strong detectors for some categories.

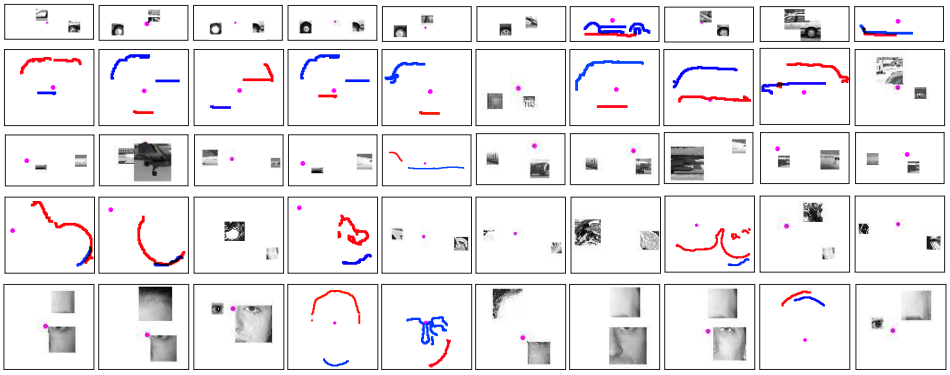

Figure 6: The first ten weak detectors from the mixed detectors in each row for the categories: carsSide(UIUC); CarsRear, Airplanes, Motorbikes and Faces (Caltech).

Detection using the mixed detector: Given a new test image we have to compute the features of the different types (here edge representation and SIFT descriptors of salient points). Then each weak detector h_i^τ of the strong detector of a category is evaluated on the corresponding representation of the image. For example if $\tau = \{edges, regions\}$ then for h_i^{edges} we evaluate this weak detector on the edge representation of the test image. If a weak detector fires on the test image it votes in a *joint* Hough Voting space. As before Mean-Shift mode estimation is used to search for maxima in this voting space.

5 Experiments

The experiments illustrate three main issues: First, that an additional source of information increases the performance of an approach. Second, that our selective learning procedure often achieves comparable results using just one model with mixed feature types instead of a separate model for each feature type (as in the CM algorithm). Finally, certain object categories prefer certain feature types.

We measure the performance as RPC-equal error rates and count a detection as correct if the detected bounding box BB_{det} and the ground truth bounding box BB_{gt} have an overlap of: $\frac{area(BB_{det} \bigcap BB_{gt})}{area(BB_{det} \bigcup BB_{gt})} \geq 0.5$. Additional detections of the same object are counted as false positives.

The parameters for the shape based BFM are set as the ones reported by Opelt *et al.* [17]. For the region based model and the two combinatorial algorithms (CM and SF) we set the number of iterations for the Boosting procedure $T = 200$. For the CM algorithm, all weights f_τ are equal (0.5), unless stated otherwise.

UIUC dataset: In table 1 we show the RPC-equal error rates for this dataset. Our region based model achieves comparable results to state-of-the-art approaches. Directly compared to the BFM of Opelt *et al.*[17], we can improve the detection performance. The two novel algorithms (CM and SF) achieve even lower detection errors. Similar results have been reported by the boundary based method of Shotton *et al.* [20]. However, Shotton *et al.* [20] use 10 pre-segmented training images whereas we are using bounding boxes as the only supervision. Additional verification steps can decrease the error further (to 2.5% in [14]) as reported by Leibe *et al.* [14]. But here we compare the plain algorithms before verification. Our method would of course also benefit from the same verification method.

Method	RM	CM	SF	Opelt[17]	Fergus[8]	Leibe[14]	Amores[2]	Shotton[20]
RPC-EER	10.5	6.2	7.0	15.0	11.5	9.0	10.0	7.2

Table 1: Comparison of the different methods: regions (RM) used separately, two models combined (CM), and our selective learning algorithm (SF). We also compare our results to the standard BFM from [17] and others on the UIUC car database.

Caltech dataset: We performed experiments on categories of the commonly used Caltech database [8]: Airplane, CarsRear, Motorbike and Faces sticking to the testing protocol from [8]. We separate the training set into training data and a validation set, and use the same splitting as reported in [17]. Table 2 shows the RPC-equal-error rates of our region based model and the two novel algorithms for combining features compared with the BFM. We achieve comparable or better detection results than the BFM and other state-of-the-art work. For classification an image is classified as positive if it contains one detected instance of this object category. The classification results are reported in table 3.

In general identical weights f_τ improve the results. More detailed investigation of the weight parameter shows e.g. for motorbikes that the lowest error rate of 1.3% can be achieved at $\mathbf{f} = [0.3, 0.7]$. However, tuning of this parameter requires human supervision as there is often no error on the validation set (which could serve as possibility for automatic tuning) and is thus not always useful.

Cat.	RM	CM	SF	Opelt *et al.* [17]	Leibe *et al.* [14]	Shotton *et al.* [20]
Cars-rear	2.9	0.0	0.5	2.3	6.1	-
Airplane	22.5	4.2	13.4	7.4	-	-
Motorbikes	4.0	2.0	3.7	4.4	6.0	7.6
Faces	2.4	1.0	3.2	3.6	-	6.0

Table 2: Comparison of our region based model (RM), the combination of two models (CM) and the selective learning approach (SF). We also show how the BFM from [17] and other state-of-the-art approaches perform on this data. Note that we report detection results here in terms of RPC-equal-error rates.

Cat.	RM	CM	SF	Opelt[17]	Fergus[8]	BarHillel[3]	Zhang[23]
Cars-rear	1.7	0.5	0.5	0.5	9.7	2.3	-
Airplane	10.8	2.9	7.1	2.6	7.0	10.3	5.6
Motorbikes	0.0	0.0	2.3	3.2	6.7	6.7	5.0
Faces	0.7	0.3	0.7	1.9	3.6	7.9	0.3

Table 3: Comparison of our region based model (RM), the combination of two models (CM) and the selective learning approach (SF) for image classification. We also show comparisons to the BFM from [17] and other state-of-the-art approaches. Note that ROC-equal-error rates are reported.

6 Discussion

It is interesting to consider the merits and limitations of the two algorithms. The CM algorithm is robust to the combination of a reliable with an unreliable model (i.e. one that achieves poor detection results). This is because the method of searching modes by Mean-Shift mode estimation in a Hough space is robust against the addition of a random distribution (the votes of the poor model) and thus the correct modes from the reliable model do not get too distracted by the addition of this second Hough voting space. For the SF algorithm we would expect it to achieve at least the minimum of the error rate that the separate models (RM and BFM) for each feature type achieve. This is generally true. However, in the case of airplanes the SF model achieves poor results. More detailed investigation shows that this is caused by over-fitting on the validation set, whereas restricting the model to only one feature type is sufficient to prevent over-fitting in this case. One solution that we are currently investigating is to either obtain more validation images or to do cross-validation on the current data.

Acknowledgements

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

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proc. ECCV*, volume 4, pages 113–130, 2002.

[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 1, pages 702–709, June 2005.

[4] E. Borenstein and S. Ullman. Learning to segment. In *Proc. ECCV*, volume 3, pages 315–328, 2004.

[5] G. Bouchard and B. Triggs. Hierarchical part-based visual object categroization. In *Proc. CVPR*, pages 710–715, 2005.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. CVPR*, volume 2, pages 142–149, 2000.

[7] 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.

[8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, pages 264–271, 2003.

[9] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. ECCV*, pages 242–256, 2004.

[10] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. CVPR*, volume 1, pages 380–387, 2005.

[11] Y. Freund and R. Schapire. A decision theoretic generalisation of online learning. *Computer and System Sciences*, 55(1):119–139, 1997.

[12] D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proc. ICCV*, pages 87–93, 1999.

[13] http://www.robots.ox.ac.uk/∼vgg/software/.

[14] 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.

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

[16] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Generic object recognition with boosting. *PAMI*, 28(3), 2006.

[17] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proc. ECCV*, volume 2, pages 575–588, May 2006.

[18] E. Sali and S. Ullman. Combining class-specific fragments for object classification. In *Proc. BMVC*, volume 1, pages 203–213, 1999.

[19] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele. An evaluation of local shape-based features for pedestrian detection. In *Proc. BMVC*, 2005.

[20] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *Proc. ICCV*, volume 1, pages 503–510, 2005.

[21] C. K. I. Williams and M. Allan. On a connection between object localization with a generative template of features and pose-space prediction methods. Technical Report Informatics Research Report 0719, School of Informatics, University of Edinburgh, 2006.

[22] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: An in-depth study. Technical Report Rapport de recherche no 5737, INRIA, France, 2005.

[23] W. Zhang, B. Yu, G.J. Zelinsky, and D. Samaras. Object class recognition using multiple layer boosting with heterogenous features. In *Proc. CVPR*, pages 66–73, 2005.