Improving AdaBoost Detection Rate by Wobble and Mean Shift *

Helmut Grabner[†], Csaba Beleznai[‡] and Horst Bischof[†]

[†]Inst. for Computer Graphics and Vision, Graz Univ. of Technology, Austria

e-mail: {hgrabner, bischof}@icg.tu-graz.ac.at

[‡]Advanced Computer Vision GmbH - ACV, Vienna, Austria

e-mail: csaba.beleznai@acv.ac.at

Abstract

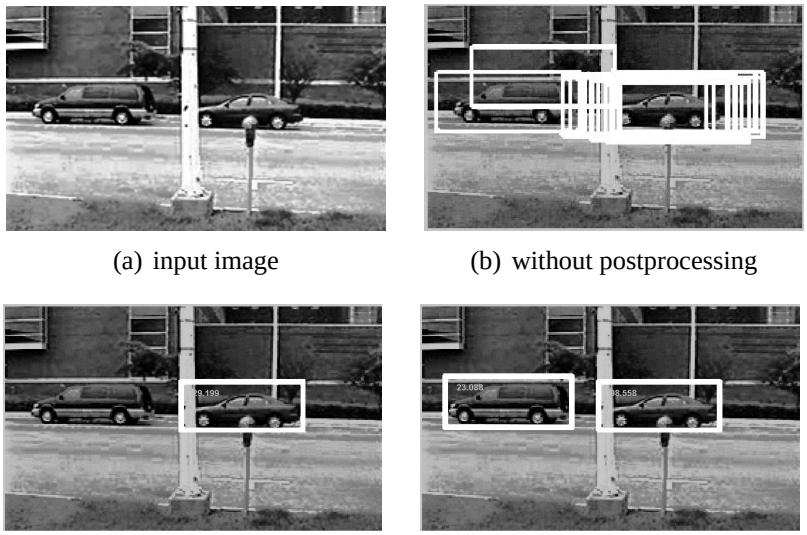
This paper describes a postprocessing algorithm for improving the performance of a detection system, i.e. to increase the detection rate and simultaneously decrease the number of false positives. In order to increase the detection rate we propose to use an approach called *wobble*, i.e. applying small random affine transformations to the image and repeating the detection step. To decrease the false positive rate we propose a fast variant of a scale adaptive mean shift algorithm. We test the proposed algorithm on a car detection task using the well known AdaBoost algorithm.

1 Introduction

Recently many different object detection algorithms have been proposed. The goal is to detect a target object at all locations and scales in a given image. Given the ever increasing computational power, nowadays exhaustive search techniques are becoming quite popular. A classifier trained on the respective object is evaluated at each location and scale of the image. Usually one will obtain multiple detections for a single object. The problem is now to decide which detection is correct and to reject incorrect ones. This is the task of the post-processing step. A common technique is to use bounding boxes to represent the spatial dimensions of the detected objects and to merge overlapping bounding boxes for multiple detections at close locations [16], [12], [1].

Figure 1 illustrates the problem we are dealing with. Using a classifier (in our case AdaBoost) on the input image of Figure 1(a) one obtains a typical result as shown in Figure 1(b).

*This work has been carried out within the K plus Competence Center ADVANCED COMPUTER VISION. This work was funded from the K plus Program. This research was partially supported by the Federal Ministry for Education, Science and Culture of Austria under the CONEX program, by the Austrian Joint research Project Cognitive Vision under sub-projects S9103-N04, S9104-N04 and by the Federal Ministry of Transport, Innovation and Technology under P-Nr. I2-2-26p Vitus2.



(a) input image (b) without postprocessing

(c) threshold (d) mean shift and wobble

Figure 1: Input image (a) using a classifier evaluated at all possible locations (b), after applying a threshold (c) is obtained (one car is missing). The method proposed in this paper obtains the result shown in (d).

The task is now to select from these multiple detections the correct one. Using a threshold-based criterion one might obtain a result shown in Figure 1(c) (one car is missed) or using a higher threshold one might obtain false positives. In general, it will be hard to define a reliable threshold. The approach proposed in this paper consisting of the wobble transform and mean shift clustering will produce the result in Figure 1(d).

As we will illustrate in the paper the proposed wobble reduce the false negative rate while on the other hand the mean shift selection reduces the false positive rate. Therefore the combined approach reduces both the false negative and the false positive rate.

The rest of this paper is organized as follows. Section 2 gives a short overview on the employed classifier. Section 3 and Section 4 introduces the two main components of the post-processing step: the mean shift clustering and the wobble transform approach. In Section 5, we apply these methods to the well known *UIUC Car Database* [1]. Finally, we present results and a conclusion.

2 Classifier

The key issue is to create an efficient classifier because the entire image is scanned at multiple scales and therefore the classifier is evaluated very often.

Our proposed concept can in principle used with any classifier, but due to its popularity we have used the classical AdaBoost classifier from Viola and Jones [16]. It allows a very fast

[1] `http://l2r.cs.uiuc.edu/~cogcomp/Data/Car/` (2004-06-21)

processing while achieving a high detection rate. The main assumption from Viola and Jones is that a small set of important features can separate the object classes from the background. This feature selection is done by using AdaBoost [7], a popular method from machine learning. As shown in the literature AdaBoost minimizes the error on the training set exponentially and tries to achieve a large margin from the decision boundary [14]. Thereby it achieves a good generalization error similar to support vector machines. For more details, see [9].

To achieve a fast evaluation a set of classifiers are trained and combined in a cascade structure. The main assumption is that most search windows will not contain the object. Therefore a small and efficient boosted classifier can be constructed which rejects many of the sub-windows not containing the object. More complex classifiers are applied to the remaining regions to achieve a low false positive rate. The obtained classifier can evaluate a sub-window very efficiently. In addition for each detection a scalar value of the likelihood is also available which corresponds to the margin.

3 Mean shift clustering

The cascaded classifier generates a probabilistic output. For each image location X we obtain multiple outputs Y_k representing object probabilities at each scale k.

We propose a non-parametric clustering-based object detection derived from the distribution of classifier output probabilities. To obtain a distribution of object probabilities at each scale, we apply kernel density estimation. Let denote $\{X_i\}_{i=1..n}$ as the image locations where classification is performed. For each scale k we obtain a probability density estimate

$$\hat{f}_k(x) = \sum_{i=1}^{n} Y_k(X_i)\, K_k\left(\frac{x - X_i}{W_k}\right), \tag{1}$$

where K_k is two-dimensional Gaussian kernel with a size equivalent to the object size W_k at the current scale and scaled by the classifier output. A similar method is used by Leibe et al. [11] in his approach.

The obtained set of two-dimensional density estimates usually contains maxima with respect to the scale. Maxima within multiple scales are determined for each image location. The derived probability density distribution is denoted as $\hat{f}_c$. It corresponds to a cumulative density estimate containing the sum of probabilities over all scales. Mean shift clustering is applied to this density estimate to delineate objects.

The mean shift algorithm is a nonparametric technique to locate density extrema or modes of a given distribution by an iterative procedure [5]. Starting from a location x the local mean shift vector represents an offset to x', which is a translation towards the nearest mode along the direction of maximum increase in the underlying density function. The local density is estimated within the local neighborhood of a kernel by kernel density estimation where at a data point a kernel weights $K(a)$ are combined with weights associated with the data, i.e. with sample weights. In our case sample weights are defined by the values of the density estimate $\hat{f}_c(a)$ at pixel locations a. The new location vector x' obtained after applying the mean shift

offset

$$x' = \frac{\sum_a K(a-x)\hat{f}_c(a)a}{\sum_a K(a-x)\hat{f}_c(a)} \,. \tag{2}$$

For a uniform kernel K it was shown that fast evaluation of Equation (2) is feasible using integral images. For more details, see [3].

Object delineation by mean shift procedure is performed using the following steps:

1. A sample set of n points $X_1 \ldots X_n$ is defined by locating local maxima in the probability density function.

2. A single mean shift iteration (see Equation (2)) is carried out at the points of the sample set with a very small initial window size, typically of 10-by-10 pixels.

3. The local covariance within the window is estimated by computing local statistical moments of the zeroth- (M_{00}), first- (M_{10}, M_{01}) and second-order (M_{20}, M_{11}, M_{02}).

   Note that mean shift computation using uniform kernel K implicitly computes the zeroth- and first order moments. Using the above quantities an elliptic approximation for the underlying distribution is obtained [4]

   $$L_1 = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c^2)}}{2}}, \qquad L_2 = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c^2)}}{2}} \,. \tag{3}$$

   L_1 and L_2 denote the major and minor axis of the elliptic estimate. The coefficients a, b and c are defined as $a = \frac{M_{20}}{M_{00}} - x'^2$, $b = 2\left(\frac{M_{11}}{M_{00}} - x'y'\right)$, $c = \frac{M_{02}}{M_{00}} - y'^2$. x' and y' are the coordinates obtained after a mean shift iteration. The orientation of the elliptic approximation is constrained to an ellipse with horizontally aligned major axis.

4. The size of the mean shift kernel is updated using L_1 and L_2. The process is repeated from step 2. Therefore during the mean shift procedure, the kernel size adapts to the shape of the underlying distribution.

5. Detected mode candidates obtained for the different points of the sample set are grouped. If within a kernel of final size several mode candidates exist, they are merged and the average of involved kernel dimensions is taken. The obtained kernels represent the spatial dimensions of the detected objects.

4 Wobble

The main idea of the *wobble* approach is that if an object could not be detected in a given image, a small change of the image, e.g. a slight change of the viewpoint may improve the detection. On the other hand, if we have already a detection a change of the viewpoint should not reduce the detectability . Since we can not move the camera (we have only a single image) we simulate

this by applying an affine transformation on the original image. Note that we do this in the detection stage and not during training, e.g. Rowley et al [13] use small amounts of translation, scale, and rotation were randomly in the training images.

Formally, we can write an affine transformation as in Equation (4). Usually, the extent of the transform, or the so-called wobble factor f is small, e.g. we use 0.1 for the experiments reported bellow.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = W \cdot \begin{bmatrix} x \\ y \end{bmatrix}, \quad W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + f \cdot \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}, \quad \alpha, \beta, \gamma, \delta \sim [-0.5\ 0.5] \tag{4}$$

This transformation is repeated several times with randomly chosen factors α, β, γ, δ and the classifier is evaluated on all these images. All the detections are summed to produce the distribution for the mean shift-based clustering.

5 Experimental results

When measuring the performance of an object detection system the two quantities of interests are clearly the number of correct detections (which we want to maximize) and the number of false detections (which we want to minimize). A system performance can only be specified by knowing both of these parameters. This is captured by recall and precision. The recall is the detection rate and the precision tells us how safe a detection is. The tradeoff between the two quantities can be measured by the F-measure [1].

On the UIUC home page an automatic evaluation program is available to determine these quantities. The located detections in position $center_x$, $center_y$ and scale $scale$ are compared to a hand labelled ground truth $center_x^\star$, $center_y^\star$, $scale^\star$. A detection is considered to be correct when Equation (5) holds [1] where α_{width}, α_{hight}, α_{scale} determine the size of the allowed region. The $error$ provides an information on the closeness of the found bounding box to the ground truth.

$$error = \frac{|center_x - center_x^\star|^2}{\alpha_{width}^2} + \frac{|center_y - center_y^\star|^2}{\alpha_{height}^2} + \frac{|scale - scale^\star|^2}{\alpha_{scale}^2} \leq 1 \tag{5}$$

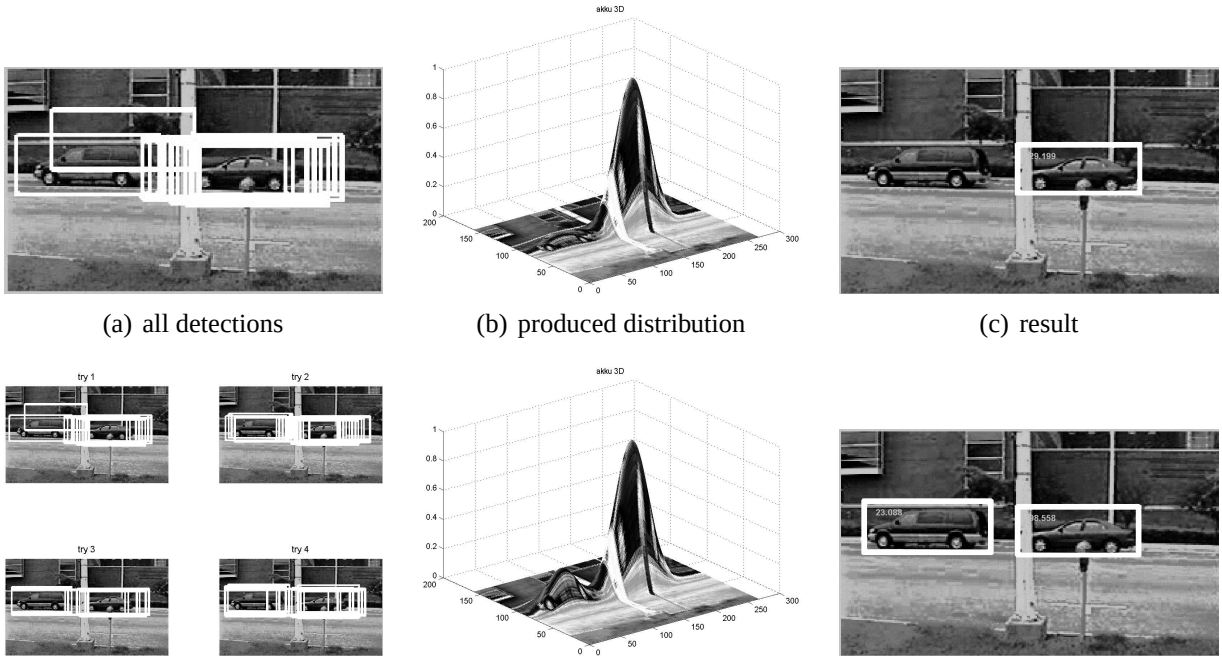
In order to get a statistically significant result each experiment is repeated ten times and then averaged (because of the random affine transformation).

We first show two specific examples to highlight the main points of the proposed method. The first example shows how to increase the detection rate, while keeping the false positive rate, the second example shows how to increase the detection rate and how to decrease the false positive rate.

5.1 Example 1

The upper row of Figure 2 depicts the detections produced by the classifier and the obtained distribution. Using the mean shift-based clustering on this distribution only the right car is detected. In order to improve the detection rate the wobble approach is used consisting of four

random transformations. As one can see from the bottom row of Figure 2 many detections occur in the image on slightly different positions this give raise to the peak corresponding to the left car.



(a) all detections (b) produced distribution (c) result

(d) wobble: all detections (e) wobble: produced distribution (f) wobble: result

Figure 2: The subplots visualize from left to right all detections and the calculated distribution. This map is then analyzed by the mean shift-based clustering and the final results are shown on the right. The upper plots show the case where no wobble was applied and plots at the bottom when wobble was employed using four transformations.

Varying the number of wobble operations results in Table 1. Without wobble the algorithm managed to detect only the right car. After performing a few wobble operations we can improve the probability to detect both objects. Note, that the false positives always zero therefore the precision of the detection is equal to 100 percent. Another nice feature is that the error on the bounding box and the recognized one decreases therefore also the quality of the detections improves.

5.2 Example 2

If in the given image a car is present but has not been detected, a relative small (low likelihood) false detection is produced. After applying four wobble operations the distribution is corrected and therefore the post-processing focuses on the real object. This process is shown in Figure 3. Similar to the example above the performance quantities over the number of wobble operations are listed in Table 2.

wobble op.	detect.	false-pos.	recall	precision	F-measure	$error_{right}$	$error_{left}$
1	10	0	50 %	100 %	67 %	0.018	-
2	17	0	85 %	100 %	92 %	0.044	0.264
4	20	0	100 %	100 %	100 %	0.032	0.112
8	20	0	100 %	100 %	100 %	0.013	0.039
16	20	0	100 %	100 %	100 %	0.022	0.039

Table 1: Performance for Example 1. Increasing the number of wobble operations the detection rate increases while achieving the same (optimal) precision, i.e. no false positives are added. The quality of the detection, the error on the bounding box improves.

wobble operations	detect.	false-pos.	recall	precision	F-measure	$error$
1	0	10	0 %	0 %	0 %	-
2	6	7	60 %	45 %	50 %	0.251
4	8	5	80 %	65 %	70 %	0.169
8	10	6	100 %	70 %	80 %	0.110
16	10	6	100 %	70 %	80 %	0.019
32	10	4	100 %	80 %	87 %	0.018

Table 2: Performance for Example 2: Increasing the number of wobble trials yields a better probability to detect the object. Again the localization error deceases.

5.3 UIUC data sets

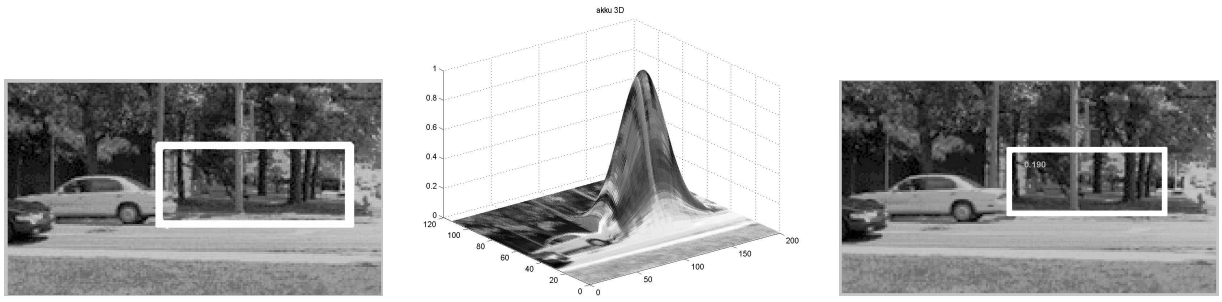
For evaluation and comparison we used the UIUC Car Database. It contains two sets of test images, the first for the single scale case and the second for the multi scale case. UIUC test set I consists of 170 images containing 200 cars. The cars have all roughly the same size. UIUC test set II consists 108 images containing 139 cars of different size. The images are of different resolution and include instances of partially occluded cars, cars that have a low contrast to the background and images with highly textured background. The evaluation criteria are the same as described in [1]. Therefore a comparison to other published methods is possible.

description	detect.	false-pos.	recall	precision	F-measure
no wobble	177	17	88.5 %	91.2 %	89.9 %
affine transf. on training	178	17	89.0 %	91.3 %	90.1 %
wobble (4 operations)	182	14	91.0 %	92.9 %	91.9 %
wobble (8 operations)	186	12	93.0 %	93.9 %	93.5 %

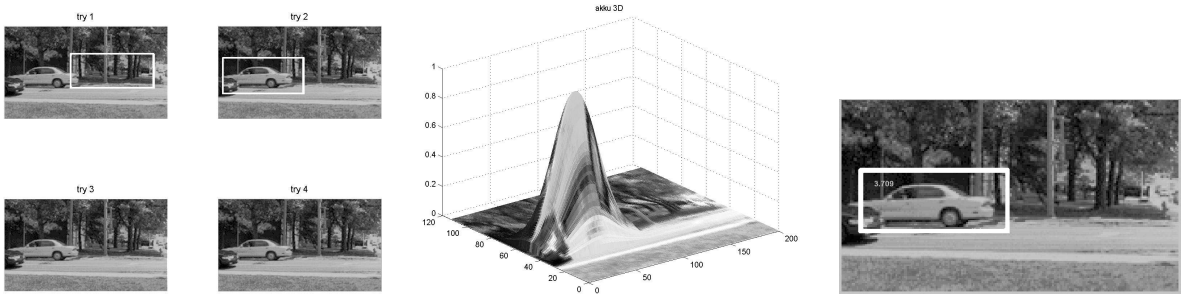
Table 3: Results depending on the number of wobble trials evaluated on the UIUC test set I.

The performance on the UIUC test set I are shown in Table 3. A comparison with other approaches is depicted in Table 4. As one can see our approach yields very good results compared to the best performing, published methodes[2]. Only Leibe et al. achieved better results but unlike

[2]The current evaluation program uses a slightly stricter criterium as published in [2] which was used by most



(a) all detections (b) produced distribution (c) result



(d) wobble: all detections (e) wobble: produced distribution (f) wobble: result

Figure 3: First, without wobble, the car could not be detected - a false positives occurs (upper row). After applying four wobble operations the distribution is corrected and the mean shift clustering finds the car correctly.

other approaches he needs a segmentation for training. For comparison we also evaluated the commonly approach applying the wobble transform - consisting of four operations - directly to the training data. It turns out that on the used test set the performance is slightly better but not as good as the use of the proposed approach.

The performance for the UIUC test set II and a comparison to other approaches is shown in Table 5. Note the significant increase in accuracy compared to the method of Agarwal et al.

6 Conclusions

We have presented a novel approach to improve both the false detection rate and the correct detection rate for an object detection system based on classifiers.

The *wobble* transform approach in combination with the mean shift-based clustering is used to increase the performance. The main drawback of the proposed methods is an increase in computation time. The main effect observed in the experiments is that an independent and random scaling of the images in both dimensions has the largest effect on the performance. This is not very surprising since not all cars have the same ratio of height and width. Roughly all

other approaches.

approach	recall	precision	F-measure
Agarwal, Roth (walkout) [2]	90.5%	64.9%	75.6%
	$\sim$ 79%	$\sim$ 80	79.5%
	70.0%	82.8%	75.9%
Agarwal et al. (walkout) [1]	72.5%	81.5%	76.7%
Fergus et al. [6]	88.5%	$\sim$ 80%	84.0%
Garg et al. [8]	94.0%	75.5%	83.7%
Schneiderman [15]	97.0%	$\sim$ 80%	87.7%
Leibe et al. [10]	97.5%	97.5%	97.5%
proposed method	93.0%	93.9%	93.5%

Table 4: Comparison of different approaches evaluated on the UIUC test set I.

approach	recall	precision	F-measure
Agarwal et al. (walkout)[1]	80.6%	8.4%	15.3%
	39.6%	49.6%	44.0%
	12.2%	77.3%	21.2%
proposed method	82.7%	71.4%	76.7%

Table 5: Comparison of different approaches evaluated on the UIUC test set II.

cars are oriented horizontally so a rotation has less effect. When we use only a random scaling in x- and y-direction, no transform of the input image is needed. An efficient implementation using scaled features is possible. Therefore, the evaluation can be performed very efficiently with the used AdaBoost classifier. Furthermore, the speed of the whole system can be improved by including prior knowledge to reduce the search space.

The proposed method is quite general and can also be used with different classifiers to improve the detection rate. It is even possible to use only binary classifiers since the superimposition of all the detections gives us different scalar values. This should be investigated in future work.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. In *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26, pages 1475–1490, 2004. 1, 5, 7, 9

[2] S. Agarwal and D. Roth. Learning a Sparse Representation for Objet Detection. In *Proceedings of the ECCV*, 2002. 7, 9

[3] C. Beleznai, B. Frühstück, H. Bischof, and W. Kropatsch. Detecting Humans in Groups Using a Fast Mean Shift Prozedure. In *Workshop of the AAPR/ÖAGM*, pages 71–78, Hagenberg, Austria, 2004. 4

[4] G.R. Bradski. Computer Vision Face Tracking For Use in a Perceptual User Interface. *Intel Technology Journal*, 1998. 4

[5] D. Comaniciu and P. Meer. Mean Shift Analysis and Applications. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1197–1203, Kerkyra, Greece, 1999. 3

[6] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003. 9

[7] Y. Freund and R. Schapire. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999. 3

[8] A. Garg, S. Agarwal, and T. Huang. Fusion of Global and Local Information for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2002. 9

[9] H. Grabner. Autodetektion mit AdaBoost. Master's thesis, Graz University of Technologie, 2004. 3

[10] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, Prague, May 2004. 9

[11] B. Leibe and B. Schiele. Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search. Springer LNCS, Vol. 3175, pages 145–153, Tübingen, Germany, Aug. 2004. 3

[12] R. Lienhart and J. Maydt. An Extended Set of Haar-like Features for Object Detection. In *Proceedings of the IEEE ICIP*, pages 900–903, 2002. 1

[13] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998. 5

[14] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. In *Proceedings of the 14th International Conference on Machine Learning*, pages 322–330. Morgan Kaufmann, 1997. 3

[15] H. Schneiderman. Feature-Centric Evaluation for Efficient Cascaded Object Detection, 2004. 9

[16] P. Viola and M. Jones. Robust Real-time Object Detection. *International Journal of Computer Vision*, 2002. 1, 2