Incorporating Geometry Information with Weak Classifiers for Improved Generic Visual Categorization

Gabriela Csurka, Jutta Willamowski, Christopher R. Dance, and Florent Perronnin

Xerox Research Centre Europe,
6 Rue de Maupertuis, 38240 Meylan, France
{gsurka, willamow, cdance, fperronn}@xeroxlabs.com

Abstract. In this paper[1], we improve the performance of a generic visual categorizer based on the "bag of keypatches" approach using geometric information. More precisely, we consider a large number of simple geometrical relationships between interest points based on the scale, orientation or closeness. Each relationship leads to a weak classifier. The boosting approach is used to select from this multitude of classifiers (several millions in our case) and to combine them effectively with the original classifier. Results are shown on a new challenging 10 class dataset.

1 Introduction

The proliferation of digital imaging sensors in mobile phones and consumer-level cameras is producing a growing number of large digital image collections and increasing the pervasiveness of images on the web and in other documents. To search and manage such collections it is useful to have access to high-level information about objects contained in the images. We are interested in recognizing several objects or image categories within a multi-class categorization system, but not in the localization of the objects unnecessary for most applications involving tagging and search. Therefore, in this paper we propose a system which is sufficiently generic to cope with many object types simultaneously and which can readily be extended to new categories. It can handle variations in view, background clutter, lighting and occlusion as well as intra-class variations.

The main novelty is to exploit a boosting approach based on interest points and simple geometrical relationships (similar scales, similar orientation, closeness) between them. We chose to adopt the boosting approach because there are many possible geometric relationships and boosting offers an effective way to select from this multitude of possible features. It was used with success in [11] to detect the presence of bikes, persons, cars or airplanes against background. However their approach differs from ours as they do not include any geometry and consider every appearance descriptor (over 2 million for our data set) without considering a vocabulary, which is impractical if geometric combinations of such descriptors are to be exploited.

The main advantage of our approach is that geometric constraints are introduced as weak conditions in contrast to others such as [4,7], where due to relatively strong

[1] This work was funded by the EU project LAVA (IST-2001-34405).

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 612–620, 2005.

geometrical (shape) constraints the methods requires the alignment and segregation of different views of objects in the dataset.

Several categorization approaches have recently been developed that are based on image segmentation [1,8,12,2], rather than the interest point descriptors exploited here. Some of these works attempt to solve the more difficult problem of labeling several regions per image. In [1] geometry has been included through MRF models of neighboring relations between segmented regions. In contrast we prefer to take a discriminative classifier approach in order to optimize overall accuracy.

The remainder of this paper is organized as follows: Section 2 describes the original SVM approach; in Section 3 we introduce an alternative based on the boosting framework; in section 4 we then describe how to incorporate weak geometry in the boosting approach and we finally present preliminary experimental results in section 4.

2 The Original Approach

We describe very briefly the original visual categorization approach introduced in [3]. The main steps of our method as applied to labeling a previously unseen image are as follows. We detect image patches and assign each of them them to one of a set of predetermined clusters (a vocabulary), on the basis of their appearance descriptors[2]. We then apply one classifier (SVM) per visual category with a one-against-all approach to handle the multiple visual categories.

The extracted descriptors of image patches should be invariant to the variations that are irrelevant to the categorization task (image transformations, lighting variations and occlusions) but rich enough to carry all necessary information to be discriminative at the category level. We used Lowe's SIFT approach [9] to detect and describe image patches. This produces scale-invariant circular patches that are associated with 128-dimensional feature vectors of Gaussian derivatives. While in [3] we used affine invariant elliptical patches [10], similar performance was obtained with circular patches. Moreover, the use of circular patches makes it simpler to deal with geometric issues.

The visual vocabulary was constructed using the k-means algorithm applied to a completely independent set of images with over 10,000 patches. We are not interested in a correct clustering in the sense of feature distributions, but rather in an accurate categorization. Therefore, to overcome the initialization dependence of k-means, we run it several times with different initial cluster centers and select the final clustering giving the highest categorization accuracy using an SVM classifier (without any geometric properties) on a subset of the dataset.

For categorization we use the SVM, which finds the hyperplane that separates two-class data with maximal margin [17]. The SVM decision function can be expressed as $f(\mathbf{x}) = \text{sign}(\sum_i y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b)$, where $\mathbf{x}_i$ are the training features from data space and $y_i \in \{-1, 1\}$ is the label of $\mathbf{x}_i$. In this paper, the input features $\mathbf{x}_i$ are the binned histograms formed by the number of occurrences of keypatches in the input image. K is a kernel function corresponding to an inner product between two transformed feature vectors, usually in a high and possibly infinite dimensional space. In our experiences

[2] In this paper, we will refer to patches assigned in this way as *keypatches* instead of *keypoints*.

we used a linear kernel, which is the dot product of $\mathbf{x}$ and $\mathbf{x}_i$. The parameters α_i are zero for most i, the sum is taken only over a selected set of $\mathbf{x}_i$ known as support vectors.

3 The Boosting Approach

An alternative to the SVM classifier is the boosting approach. Here we exploit the generalized version of the AdaBoost algorithm described in [15] and improved by Rätsch *et al* [14] by adding soft margins. Boosting is a method of finding an accurate classifier H by combining M simpler classifiers h_m:

$$H(\mathbf{x}) = \Big(\sum_{m=1}^{M} \alpha_m h_m(\mathbf{x}) \Big) / \Big(\sum_{m=1}^{M} \alpha_m \Big)$$

Each simpler classifier $h_m(\mathbf{x}) \in [-1, 1]$ needs only to be moderately accurate and is therefore known as a *weak classifier*. They are chosen from a classifier space to maximize correlation[3] of the predictions and labels $r_m = \sum_i D^m(i) h_m(\mathbf{x}_i) y_i$, where $D^m(i)$ is a set of weights (distribution) over the training set. At each step the weigths are updated by increasing the weights of the incorrectly predicted training examples:

$$D^{m+1}(i) = D^m(i) \exp\{-\alpha_t y_i h_m(\mathbf{x}_i)\} / Z_m$$

where $\alpha_t = \frac{1}{2} \log \frac{1+r_m}{1-r_m}$ and Z_m is a normalization constant, such that $\sum_i D^{m+1}(i) = 1$. In the case of soft margins, α_t and $D^{m+1}(i)$ depend also on a regularization term which takes into consideration the predictions and weights produced by the previous steps in order to eliminate the influence of outliers [14].

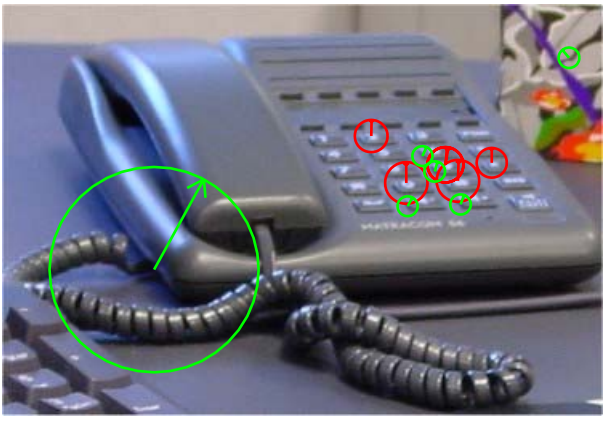
To define the weak classifiers we consider the same inputs as for the SVM, i.e. the binned histograms $\mathbf{x}_i$. The simplest keypatch-based weak classifier $h^{k,T}$ counts the number of patches whose SIFT features belong to cluster k, which is equivalent to compare $\mathbf{x}_i^k$ to the threshold T. If this number is at least T, then the classifier output is 1, otherwise 0. We may build similar weak classifiers $h^{k,l,T}$ from a pair of keypatch types k, l. If at least T keypatches of both types are observed, then the classifier output is 1.

In practice we select weak classifiers by searching over a predefined set of thresholds such as $\{1, 5, 10\}$. The opposite weak classifier $h^{k,\bar{T}}$ can also be defined by inversing the inequality. Four such definitions are possible for pairs of keypatches $h^{kl,T}$, $h^{kl,\bar{T}}$, $h^{kl,T\bar{T}}$ and $h^{kl,\bar{T}T}$. In practice, we search over the full set of different possibilities when working with weak classifiers and refer to them collectively as h^k and h^{kl}. Obviously, it would be possible to further extend the definition for pairs to applying a different threshold to each keypatch type. In practice, we avoid this as it results in a prohibitively large number of possible weak classifiers.

4 Incorporating Weak Geometric Information

In this section we describe some of the possibilities to construct weak geometrical constraints on image patches. As input, we assume each patch i in a query image has been

[3] This is equivalent to minimizing the training error equal to $(1 - r_m)/2$.



$h_{\sigma}^{g,5}$, $h_{\sigma\theta}^{g,4}$, $h_{\sigma}^{rg,2}$, $h_{\theta}^{rg,5}$, $h_{\sigma\theta}^{rg,2}$, $h_{g\cap r}^{1}$ and $h_{g\in r}^{1} = 1$
$h^{r,6}$, $h_{\theta}^{g,6}$, $h^{rg,6}$, $h_{\sigma=}^{rg,1}$, $h_{\theta=}^{rg,1}$, $h_{\sigma\theta=}^{1}$ and $h_{r\subset r}^{1} = 0$

Fig. 1. Examples of weak classifiers on a typical image for keypatches of type r, g (red or green). For clarity, only the patches of type r and g are shown. In these examples, the threshold T on which the weak classifiers depend has been chosen as large as possible for output 1 (first row) and as small as possible for output 0 (second row).

labeled according to its appearance, via the index of the cluster centre k_i to which it is assigned. We associate to each patch its orientation θ_i and a ball (circular patch) B_i which has position p_i and scale σ_i.

A simple way to incorporate geometrical information in weak classifiers depending on one keypatch is to threshold the number of interest points belonging to a cluster k and having a particular *orientation*. A large number of different orientations are produced by interest point detectors. Therefore we exploit a coarse quantization of the orientations into eight bins. Two keypatches are considered to have the same orientation if they fall into the same bin. This does not constitute exact orientation invariance, as a small rotation could cause two keypatches in one bin to move to different bins. However, this approach is more efficient than directly measuring and thresholding the difference in orientations $\|\theta_i - \theta_j\|$ between pairs of keypatches.

Likewise, we define sets of weak classifiers that count the number of keypatches with the same *scale* and a set that count patches with both the same *scale and orientation*. The scale bins are selected with logarithmic spacing, in order to approximate scale invariance. Collectively[4] these classifiers are denoted by $h_{\theta}^{k}, h_{\sigma}^{k}, h_{\sigma,\theta}^{k}$.

Another way to incorporate geometry is to count the *number of interest points in the ball* around a keypatch of a given type. This count is made irrespective of the type of keypatches in the ball. As with the other weak classifiers, this property is invariant to shift, scaling and rotation. In a given image, there may be multiple keypatches of a given type containing different numbers of points. We define h_{N}^{k} in terms of the keypatch of type k with the maximum number of points in its ball.

[4] Considering similar threshold reversals as for h^{k} and h^{kl}, e.g. $h_{\theta}^{k,T}$ and $h_{\theta}^{k,\bar{T}}$.

Taking two types of keypatches k and l into consideration, there are more ways to introduce geometry. Classifiers based on common scale or orientation can be extended in two obvious ways. Firstly we can require that the patches of type k and those of type l have *identical* scale and/or orientation, giving $h^{kl}_{\sigma=}, h^{kl}_{\theta=}, h^{kl}_{\sigma\theta=}$. Alternatively we can allow each type to have their own independent scales or orientations, giving $h^{kl}_{\sigma}, h^{kl}_{\theta}, h^{kl}_{\sigma\theta}$. The latter corresponds to a Boolean combination of single point classifiers h^k_σ and h^l_θ.

A weak classifier h^{kl}_N can be constructed similarly to h^k_N that checks for the existence of a pair of interest points labeled k, l such that both of them have at least T interest points inside their balls.

We additionally consider five other ways of exploiting the position information associated with patches:

- $h_{k\in l}$ tests if there are at least T keypatches labeled l which contain an interest point labeled k within their ball.
- $h_{k\subset l}$ tests if there are at least T keypatches labeled l whose balls contain the whole ball of an interest point labeled k.
- $h_{k\cap l}$ tests if there are at least T keypatches labeled l whose balls intersect with the ball of at least one interest point labeled k.
- $h_{k\propto l}$ tests if there are at least T keypatches labeled l such that their closest neighboring interest points in the image are labeled k.
- $h_{k\in\aleph^N_l}$ tests if there are at least T keypatch labeled l such that there exist a keypatch labeled k among its N closest neighbors.

The set of weak classifiers we considered is summarized in Table 1 and Figure 1 illustrates some of them on a concrete example. Of course there are a lot of other possibilities that can be further experimented.

Table 1. Complete list of weak classifiers investigated. $\big|A\big|$ denotes the cardinality of the set A. $p \propto q$ indicates that p is the closest point to q and $\aleph^N_{p_j}$ is the set of the N closest neighbors of p_i

h	$h=1$ if this quantity $\geq T$	h	$h=1$ if this quantity $\geq T$
$h^{k,T}_{\sigma}$	$\max_{\sigma}\big\|\{i : k_i = k, \sigma_i = \sigma\}\big\|$	$h^{k,T}_{\sigma\theta}$	$\max_{\sigma,\theta}\big\|\{i : k_i = k, \sigma_i = \sigma, \theta_i = \theta\}\big\|$
$h^{k,l,T}_{\sigma}$	$\min_{u\in\{k,l\}}\max_{\sigma}\big\|\{i : k_i = u, \sigma_i = \sigma\}\big\|$	$h^{kl,T}_{\sigma\theta}$	$\min_{u\in\{k,l\}}\max_{\sigma,\theta}\big\|\{i : k_i = u, \sigma_i = \sigma, \theta_i = \theta\}\big\|$
$h^{k,T}_{\theta}$	$\max_{\theta}\big\|\{i : k_i = k, \theta_i = \theta\}\big\|$	$h^{kl,T}_{\sigma\theta=}$	$\max_{\sigma,\theta}\min_{u\in\{k,l\}}\big\|\{i : k_i = u, \sigma_i = \sigma, \theta_i = \theta\}\big\|$
$h^{kl,T}_{\theta}$	$\min_{u\in\{k,l\}}\max_{\theta}\big\|\{i : k_i = u, \theta_i = \theta\}\big\|$	$h^{T}_{k\in l}$	$\big\|\{j : k_j = l, \exists k_i = k, p_i \in B_j\}\big\|$
$h^{kl,T}_{\sigma=}$	$\max_{\sigma}\min_{u\in\{k,l\}}\big\|\{i : k_i = u, \sigma_i = \sigma\}\big\|$	$h^{T}_{k\subset l}$	$\big\|\{j : k_j = l, \exists k_i = k, B_i \subset B_j\}\big\|$
$h^{kl,T}_{\theta=}$	$\max_{\theta}\min_{u\in\{k,l\}}\big\|\{i : k_i = u, \theta_i = \theta\}\big\|$	$h^{T}_{k\cap l}$	$\big\|\{j : k_i = l, \exists k_i = k, B_i \cap B_j \neq \emptyset\}\big\|$
$h^{k,T}_{B}$	$\max_{i}\big\|\{j : k_i = k, p_j \in B_i\}\big\|$	$h^{T}_{k\propto l}$	$\big\|\{j : k_j = l, \exists k_i = k, p_i \propto p_j\}\big\|$
$h^{kl,T}_{B}$	$\max_{i}\min_{u\in\{k,l\}}\big\|\{j : k_i = u, p_j \in B_i\}\big\|$	$h^{T}_{k\in\aleph^N_l}$	$\big\|\{j : k_j = l, \exists k_i = k, p_i \in \aleph^N_{p_j}\}\big\|$

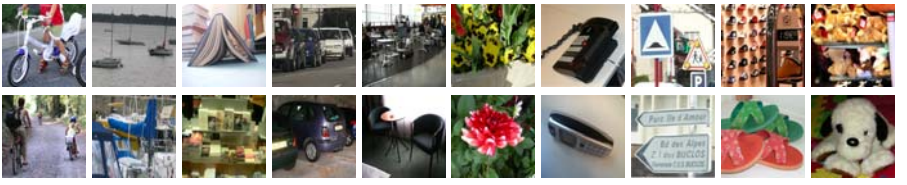


Fig. 2. Examples from our 10 class dataset

Table 2. Correct classification rates for: boosting without geometry ($h_k, h_{k,l}$); SVM with a linear kernel; boosting all types of weak classifiers h_{all} and boosting SVM with all type of weak classifiers (SVM_{all}). The standard error on the correct rate for each category is about 0.4% of the mean over the folds and is 0.2% for the overall mean.

classes	bikes	boats	books	cars	chairs	flowers	phones	road signs	shoes	soft toys	mean
h_k	61.7	74.5	67.0	55.6	50.7	82.5	67.6	61.4	73.9	68.9	**66.4**
$h_{k,l}$	64.6	76.1	68.5	61.0	50.7	84.6	69.6	64.8	76.6	69.2	**68.6**
SVM	69.2	79.3	70.3	72.1	58.8	86.7	70.4	69.0	86.3	79.2	**74.1**
h_{all}	70.0	73.8	68.2	64.1	57.4	82.9	68.0	61.9	75.2	76.2	**69.8**
SVM_{all}	74.6	81.8	78.2	77.5	65.2	89.6	76.0	76.2	83.8	83.8	**78.7**

5 Results

In our experiments we used a dataset of 3715 images from 10 categories[5]. The database can be downloaded from ftp://ftp.xrce.xerox.com/pub/ftp-ipc. Figures 2 shows some images from the database.

We experimented with several types of classifiers. In all cases we worked with a vocabulary of 1,000 keypatch types. This was selected as beeing a good trade-off between computational efficiency and classification accuracy.

First we compared directly the boosting approach with the SVM. The first three rows of Table 2 show the correct classification rate for each class: an image I of category i was considered as correctly classified if the output of the classifier $H_i(I) > H_j(I), \forall j \neq i$. All results are means of a 5-fold cross validation scheme. We can see that the SVM outperforms the Boosting approach.

Willing to incorporate weak geometry, we first preselected[6] from each type of geometry presented in Section 4 a certain number ($M = 200$) of weak classifiers that performed the best when only considering this type of weak hypotheses (e.g. $h^{kl}_{\sigma\theta}$). This preselection step allow us also to investigate how well each type of weak classifier combined trought a boosting framework perform independently (see Table 3).

[5] The number of images per class were: bikes(243), boats(439), books(272), cars(315), chairs(346), flowers(242), phones(250), road signs(211), shoes(525) and soft toys(262).

[6] This is neccesary as searching in the space of all possible weak classifiers of all 16 types proved to be too time consuming. Indeed, searching $M = 200$ times over one type of weak classifier space takes about 30min for one fold and one class on a 3GHz Pentium 4 PC.

Table 3. Mean results on boosting individual type of weak classifiers (first row) and their percentage of beeing choosed when combined with SVM

h^k_σ	h^{kl}_σ	$h^{kl}_{\sigma=}$	h^k_θ	h^{kl}_θ	$h^{kl}_{\theta=}$	$h^k_{\sigma\theta}$	$h^{kl}_{\sigma\theta}$	$h^{kl}_{\sigma\theta=}$	h^k_B	h^{kl}_B	$h_{k\cap l}$	$h_{k\in l}$	$h_{k\subset l}$	$h_{k\propto l}$	$h_{k\in\aleph^5_l}$	$h_{k\in\aleph^{10}_l}$
63.6	66.5	46.2	62.1	62.6	48.8	61.6	64.5	48.8	62.8	63.8	63.3	64.1	53.8	58.5	62.4	64.5
2	21.2	2.8	0.4	13.5	3.2	0.4	9	1.6	3.8	35.7	2.7	0.8	0.1	0.3	0.9	1.6

Fig. 3. Examples of the most relevant clusters (3) for h^k and the pair of clusters h^{kl}_σ, $h_{k\subset l}$ and $h_{k\cap l}$ respectively in case of chairs, road signs, boats and flowers. In case only green keypatches are shown means that we obtained $k = l$.

Table 4. Confusion matrix and mean ranks for SVM_{all}

true classes →	bikes	boats	books	cars	chairs	flowers	phones	r. signs	shoes	s. toys
bikes	**74.6**	1.6	1.5	0.6	5.5	1.2	0	1	0.6	0.8
boats	3.3	**81.8**	4.1	4.7	4.6	1.2	0.4	1	2.1	0.8
books	0	2.1	**78.2**	1.3	2.3	0	4.8	3.3	0.7	1.1
cars	3.8	3.7	2.2	**77.5**	7	0.4	2	3.3	1.3	0.4
chairs	10.4	2.8	4.4	4.8	**65.2**	2.1	2.8	4.3	2.5	0
flowers	1.2	0.9	0	0.6	1.8	**89.6**	0	1.4	0.6	0.8
phones	0.8	0.7	3	2.9	2.3	0.4	**76**	1	2.1	0.4
road signs	1.3	0.9	2.2	1.6	4.6	1.3	2.4	**76.2**	1.3	0
shoes	2.9	5.3	3.7	6	5.5	1.3	10.4	7.6	**83.8**	11.9
soft toys	0.4	0.2	0.7	0	1.2	2.5	1.2	0.9	5	**83.8**
mean ranks	1.4	1.3	1.4	1.3	1.8	1.2	1.9	1.5	1.5	1.1

Figure 3 illustrates examples of most relevant weak classifiers selected by boosting single type classifiers and Table 3 second raw the percentage of beeing choosed when combined with SVM through boosting. Mainly weak classifiers based on pairs of keypatches were selected.

We then combined the selected base learners across the different types through boosting. First the 17 type of geometry based weak learners were combined with hypotheses h_k and $h_{k,l}$. This (see fourth row of Table 2) slightly improved the boosting resuls without geometry (first two rows) but gave still much lower performance than applying the SVM. Therefore we rather combined the SVM outputs with geometrical

weak classifiers trough generalized AdaBoost (Table 2 fifth row). The SVM outputs were normalized to $[0, 1]$ using a sigmoid fit[7] [13] and then mapped to $[-1, 1]$.

The SVM performance was significantly improved. Table 4 shows the confusion matrix and the mean ranks (the mean positions of the correct labels when labels output by the multi-class classifier are sorted by the classifier score) for this combined classifier.

6 Conclusions

We have investigated how weak classifiers depending on geometric properties can be exploited for generic visual categorization. Results have been given on a challenging ten-class dataset which is publicly available. The benefits of the proposed method are its efficiency, invariance and good accuracy on a challenging dataset. Overall improvement in error rate has been demonstrated through the use of geometric information, relative to results obtained in the absence of geometric information.

While we have explored 19 types (17 with geometry) of weak classifier, many more can be envisaged for future work. Geometric properties are of course widely used in matching. It will be interesting to explore how recent progress in this domain such as techniques in [5,6] can be exploited for categorization. It will also be interesting to evaluate other approaches to boosting in the multiclass case such as the joint-boosting proposed in [16], which promise improved generalization performance and the need for fewer weak classifiers.

References

1. P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *Proc. ECCV*, volume 1, pages 350–362, 2004.
2. Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *JMLR*, 5:913–939, 2004.
3. G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.
4. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, volume 2, pages 264–271, 2003.
5. V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *Proc. ECCV*, volume 1, pages 40–54, 2004.
6. S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *Proc. BMVC*, volume 2, pages 959–968, 2004.
7. B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc ECCV Workshop on Statistical Learning in Computer Vision*, pages 17–32, 2004.
8. Y. Li, J. A. Bilmes, and L. G. Shapiro. Object class recognition using images of abstract regions. In *Proc. ICPR*, volume 1, pages 40–44, 2004.

[7] This transformation of SVM outputs to confidence was also applied when we ranked the outputs from different classes

9. D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.
10. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*, volume 1, pages 128–142, 2002.
11. A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. ECCV*, volume 2, pages 71–84, 2004.
12. J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. GCap: Graph-based automatic image captioning. In *Proc. CVPR Workshop on Multimedia Data and Document Engineering*, 2004.
13. J. C. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.
14. G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for Adaboost. *ML*, 42(3):287–320, 2000.
15. R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
16. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *Proc. CVPR*, volume 2, pages 762–769, 2004.
17. V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.