Object Class Recognition Using Multiple Layer Boosting with Heterogeneous Features

Wei Zhang[1] Bing Yu[1] Gregory J. Zelinsky[2] Dimitris Samaras[1]
Dept. of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794
{wzhang, ybing, samaras}@cs.sunysb.edu[1]

Dept. of Psychology
SUNY at Stony Brook
Stony Brook, NY 11794
Gregory.Zelinsky@stonybrook.edu[2]

Abstract

We combine local texture features (PCA-SIFT), global features (shape context), and spatial features within a single multi-layer AdaBoost model of object class recognition. The first layer selects PCA-SIFT and shape context features and combines the two feature types to form a strong classifier. Although previous approaches have used either feature type to train an AdaBoost model, our approach is the first to combine these complementary sources of information into a single feature pool and to use Adaboost to select those features most important for class recognition. The second layer adds to these local and global descriptions information about the spatial relationships between features. Through comparisons to the training sample, we first find the most prominent local features in Layer 1, then capture the spatial relationships between these features in Layer 2. Rather than discarding this spatial information, we therefore use it to improve the strength of our classifier. We compared our method to [4, 12, 13] and in all cases our approach outperformed these previous methods using a popular benchmark for object class recognition [4]. ROC equal error rates approached 99%. We also tested our method using a dataset of images that better equates the complexity between object and non-object images, and again found that our approach outperforms previous methods.

1. Introduction

The problem of object recognition has long challenged the computer vision community. Changes in pose, scale, occlusion, and lighting conditions can dramatically alter the appearance of objects, and in so doing require models of object recognition to accommodate an enormous degree of appearance-based variability. Yet despite the difficulties of the object recognition problem, the problem of object class recognition is arguably even more difficult. In addition to the variability associated with instance-based object recognition, object class recognition is complicated by the variability existing within an object class. Moreover, whereas instance-based object recognition requires discriminating a particular object from other objects of the same class, object class recognition requires discriminating a class of objects from every other object or pattern in the world not belonging to the target object class. However, despite the difficulty of the problem, the field of object class recognition has enjoyed a recent surge in popularity fueled by the emergence of new approaches [1, 2, 3, 4, 12, 7, 11, 13, 14, 16]. Our approach adds to this growing body of work by showing how a single model that combines multiple sources of feature information can yield superior class detection performance.

Learning is crucial in any recognition system, be it computer or human. Two machine learning methods have recently been applied to the problem of object class recognition. Fergus et al. [4] proposed a probability model to represent an object class in terms of a constellation of learned parts. The parameters of this model are in turn learned using an EM algorithm. This model has been tested with great success using the Caltech database, which has since become a benchmark for other methods of object class recognition. In 2001, Viola and Jones proposed the Adaboost model of rapid object detection [14]. Although this approach uses only three types of rectangular features, an enormous number of combinations can be generated by allowing feature variation in size, orientation, and placement in the image. The boosting technique then selects from these combinations a small set of "good" features and uses these for classification. This method has been applied with great success to face detection, a specific case of the more general problem of object class detection. In our paper we describe a method for improving Adaboost's detection performance by broadening the pool from which it selects useful features.

Various methods have been developed for using texture-based local features in object recognition. These methods

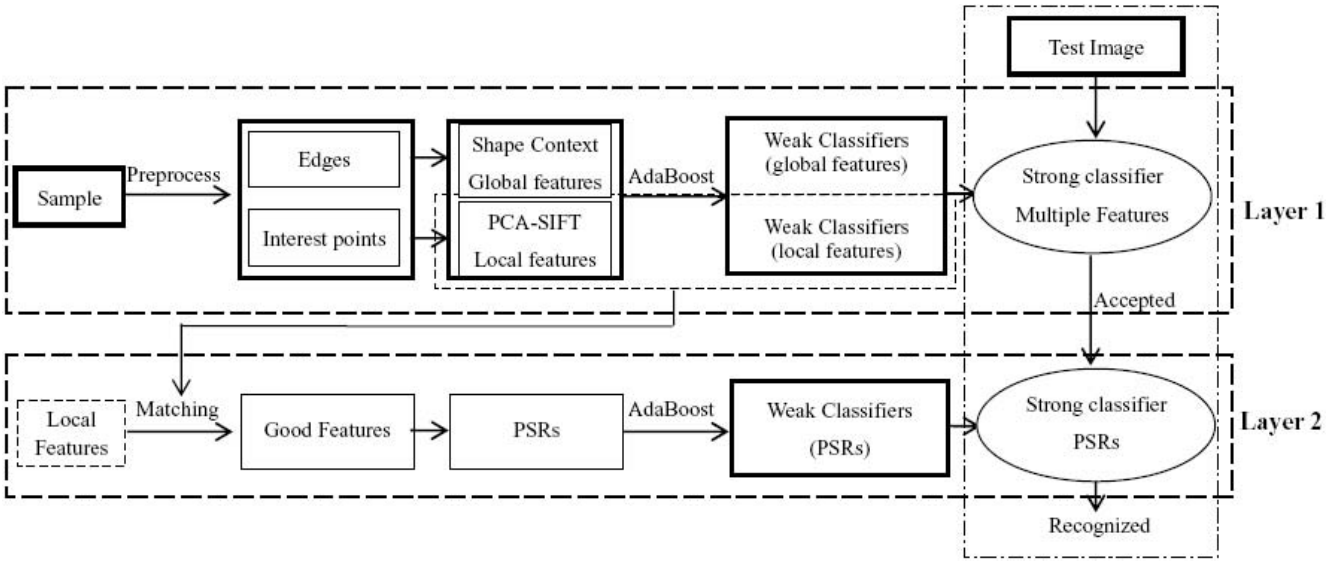


Figure 1. Multi-layered Adaboost learning model. Strong classifiers are formed from the local and global features in Layer 1 and the PSR features in Layer 2. A cascaded decision requires acceptance from both classifiers before a sample is recognized as a member of the target class.

typically include an interest point detector and a local feature descriptor, which are generally invariant to translation and in-plane rotation. Recent work[10] summarized and compared several of these local feature descriptors, including Scale Invariant Feature Transform (SIFT), steerable filters, differential invariants, and moment invariants, and concluded that the SIFT descriptor performs the best according to several evaluation criteria [10]. Opelt et al. [12] proposed a model of object class recognition that combines three interest point detectors and four types of local feature descriptors, with AdaBoost used to choose features for classification. However, all of these features were local texture features, making this approach very different from our feature combination method. Levi and Fink [7] adopted a similar multiple feature approach to object recognition. Their model used boosting in conjunction with Haar-like features, orientation features, and even color features. However, as in the case of [12], they neglected to consider global and spatial features. Our method therefore differs from these previous methods by considering global and spatial features in addition to local features. To represent local features, we use PCA-SIFT, a recent variant of SIFT [6]. Our decision to use this local descriptor was based on pilot work showing that PCA-SIFT generally outperforms SIFT for generic object recognition.

Object class recognition can also be accomplished using shape features. For example, one method [13] finds the gradient directions in an image (based on triplets of points) and uses these gradient indices to form a histogram feature vector. The global shape of an object is defined indirectly in terms of these image gradients. The similarity between two images is measured by the inner products of their histogram features, and an image is classified as a member of the target class if this similarity value is sufficiently close to one of the training images. Our method uses global shape very differently. Rather than matching a global feature vector to a training image, we add shape context descriptors to the pool of features available to Adaboost. By doing this, Adaboost is free to select the best set of features for discrimination and recognition, be they local or global, from this more diverse feature pool.

Object recognition can also exploit the spatial relationships between features. One example uses geometric constraints for person detection [11]. Simple spatial features were represented by the relative locations of human parts (e.g., head, arms, shoulders) learned through training samples. Similarly, Fergus et al. [4] used joint Gaussian density to describe the distribution of feature locations. Yet another method, introduced by Agarwal and Roth [1], coded the spatial relationship between each pair of detected parts as a binary feature vector, which was then input to a Winnows learning network. Of these various methods, our approach is most similar to [1]. Like Agarwal and Roth, we represent pairwise spatial relationships (PSRs) between features. However, unlike Agarwal and Roth, we use a second round of boosting to define PSR descriptors from the pool of local selected features during the initial application of Adaboost.

We propose combining three popular recognition methods, local texture, global shape, and PSR features, within a single multi-layer Adaboost model. These three approaches

have complimentary strengths, which ideally should be combined. By putting these three feature types into a common pool and allowing Adaboost to select the features best suited to a given class detection task, our method approaches this ideal.

2. Layer 1: local and global features

Our approach to object class recognition is to use a two-layer AdaBoost training network (see Figure 1). The function of the first layer is to choose the set of local and global features that best describe the object class. We chose PCA-SIFT over SIFT to represent local texture features, and the shape context method to represent global features. These two sets of features are then boosted into a strong Layer 1 classifier. Layer 2 boosting requires first to locate the good features from each sample based on the distances between the most discriminant local features selected by Layer 1. Pairwise spatial relationships are then computed between these features using the method described in [2]. These PSR features are then input to the second layer of Adaboost. This two-layered boosting method produces two strong classifiers, which can then be used in a cascaded fashion for recognition. An image is classified as containing an object class if conditions set on both classifiers are satisfied.

2.1. Local features: PCA-SIFT

The Scale Invariant Feature Transform (SIFT) descriptor is a widely used texture-based feature introduced by Lowe [8, 9]. A SIFT feature for a point consists of a histogram representation of the gradient orientation and magnitude information within a small image patch surrounding this point. Recently, the PCA-SIFT descriptor further improved the matching accuracy of SIFT [6]. PCA-SIFT differs from SIFT in that horizontal and vertical gradients are computed in a local neighborhood of the image patch surrounding the key point. Principle component analysis (PCA) is then applied on these gradient features to extract a more compact representation of the local patches.

Our decision to use PCA-SIFT rather than SIFT was motivated by three reasons. First, the feature vector produced by SIFT will contain a great deal of redundant information, which is undesirable for a description of an object class. This redundancy stems mainly from the inclusion of background features in the local image patches, as well as the redundancy expected from different objects of a class sharing the same local features. PCA-SIFT minimizes this redundancy. Second, PCA-SIFT is much faster than SIFT, which is a non-trivial concern when using Adaboost. Because of the large number of features involved, and the fact that a distance matrix must be computed between each pair of features, Adaboost training in most cases is very slow [14]. Given that the feature vector size of PCA-SIFT is 20 whereas the SIFT descriptor has a length of 128, the computation time of SIFT is more than 6 times that of PCA-SIFT. Our third reason for choosing PCA-SIFT over SIFT is pragmatic. Based on our own experiments we have determined that, for most cases, a PCA-SIFT descriptor simply outperforms a SIFT descriptor (see Table 1 and 3).

Object class recognition under PCA-SIFT is the same as under SIFT in that both methods require key points to be located within each image. We detect peaks in difference of Gaussian maps to define these interest points [9]. PCA-SIFT features are then computed at each of these key points, with the similarity between any two features quantified by simple Euclidean distance.

2.2. Global features: shape context

Knowledge of an object's global shape can be a powerful source of information for object detection. To obtain the shape features of an object, we first apply a Canny edge detector on the sample image. We iteratively remove points from the edge contour that are near to other points until only n points remain. A shape context operator is applied to these remaining points to describe the shape of an object. Shape context is a scale and rotation invariant local descriptor that discretizes and indexes the distances and orientations between all of the n points on the shape, where n is a freely chosen parameter. The distribution of these indices are described by a coarse histogram feature consisting of uniform bins in log-polar space [2]. The similarity between two shape features is measured by χ^2 statistic.

Although shape context can be used as a standalone method of object recognition by computing the shape histogram distance between two shapes, the approach suffers from two limitations. First, because it is a global feature, it is relatively sensitive to object occlusion. Second, when an object appears on a very complex background, it may be hard to extract the shape contour needed to create the shape context histogram [2]. By combining local (PCA-SIFT) and global (shape context) features into a single pool, a novel contribution of our approach is that Adaboost can select shape features from this pool when it is advantageous to do so, and ignore them when it is not.

2.3. Boosting with multiple features

Boosting refers to the general method of producing a very accurate prediction rule by combining relatively inaccurate rules-of-thumb [5]. It has been used widely in computer vision, particularly for object recognition, since the success of Viola and Jones' face detector [14].

AdaBoost is a supervised learning algorithm. It takes a training set and images $\{I_i, i = 1, ..., N\}$ and their associated labels, $\{l_i, i = 1, ..., N\}$, where N is the number of training images, and $l_i = 1$ if the image contains the object

Figure 2. Red squares show a subset of the good features on several sample images. The green circle in each image shows one of the PSR features selected by Adaboost in Layer 2. Note the consistency among the PSR features for the three images in each row (please print in color for best results).

and $l_i = 0$ otherwise. Each training image is represented by a set of features $\{F_{i,j}(v_{i,j}, t_{i,j}), j = 1, ...n_i\}$, where n_i is the number of features in sample I_i, $v_{i,j}$ is a vector indicating the value of the feature, and $t_{i,j}$ is the feature type. The feature set $\{F_{i,j}\}$ of each sample includes PCA-SIFT and shape context features. To select features from this set, Adaboost first has to initialize the weights of the training samples w_i to $\frac{1}{2N_p}$, $\frac{1}{2N_n}$, where N_p and N_n are the number of positive and negative samples respectively. Then, for each round of AdaBoost, we select one feature as a weak classifier and update the weights of the training samples.

The goal is to choose the T^m (The superscript 'm' is used to indicate the multiple features layer.) features that have the best ability to discriminate the target object class from the non-target object class. Each selected feature forms a weak classifier, h_k^m, consisting of three components: a feature vector (f_k), a distance threshold (θ_k), and an output label (u_k). We only use the features from the positive training samples as weak classifiers in this paper. For each feature vector $F_0(v_0, t_0)$, we compute the distance between it and the training sample, i, defined as $d_i = \min_{1 \le j \le n_i, t_{i,j} = t_0} D(F_{i,j}(v_{i,j}, t_{i,j}), F_0)$. Note that each feature type will have its own distance metric, as discussed in Sections 2.1 and 2.2. The classification rule is:

$$h(f, \theta) = \begin{cases} 1, d < \theta \\ 0, d \ge \theta \end{cases}. \quad (1)$$

The algorithm for each round of boosting is as follows:

- *Normalize the weights of the training samples such that* $\sum_{i=1}^{N} w_i = 1$.
- *For each feature,* f_k*, in each positive training sample, train a weak classifier,* $h_k^m(f_k, \theta_k)$*. Following the method described in [15], we define the classification error to be:* $\epsilon_k = \sum_{i=1}^{N} w_i |h_k^m - l_i|$.
- *Find the classifier* h_t^m *having the lowest classification error,* ϵ*, and update the sample weights according to:* $w_i^* = w_i \beta_t^{1-|h_t^m - l_i|}$*, where* $\beta t = \sqrt{\frac{1-\epsilon}{\epsilon}}$.

After the desired number of weak classifiers have been found, the final strong classifier can be defined as:

$$H^m = \begin{cases} 1, \sum_{t=1}^{T^m} \alpha_t h_t^m > \Omega^m \\ 0, otherwise \end{cases} \quad (2)$$

where $\alpha_t = log(1/\beta_t)$ and Ω^m is a threshold that can be adjusted to achieve a desired recognition rate[1].

3. Layer 2: spatial relations

So far, we have discussed the local texture and global shape features used by our model of object class recognition. We will now describe how PSR features are computed and used under our feature combination approach. The creation of the PSR features occurs in a second layer of our model, with spatial relationships being found only for the good local features suggested by Layer 1. Based on preliminary work, we found that it was unnecessary and computationally very expensive to code the spatial relationships between all of the interest points from Layer 1. We further restrict this pool to include only the good local features from Layer 1 to avoid any redundancy between the shape context features from Layer 1 and the PSR features from Layer 2.

The boosting in Layer 1 will select a set of features, $\{f_k, k = 1, ...T^m\}$, to be used as weak classifiers, $\{h_k^m, k = 1, ...T^m\}$. For each of the local discriminant features, $f_k(v_k, t_k)$, we find a "good" feature, $g_{i,k}$, in sample (I_i, l_i) using a distance-based similarity metric:

$$g_{i,j,k} = \arg \min_{1 \le j \le n_i, t_{i,j} = t_k} D(F_{i,j}(v_{i,j}, t_{i,j}), f_k(v_k, t_k)). \quad (3)$$

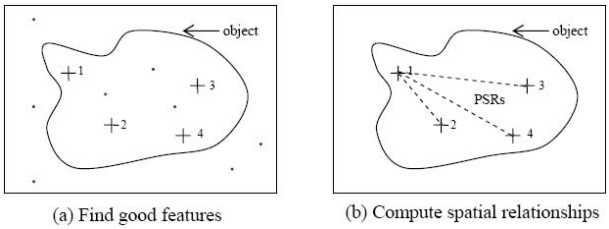
Note that the relative distances and orientations between these features $\{g_{i,j,k}, k = 1, ..., T^m\}$ are computed using the shape context descriptor from Layer 1, but the other assumptions of the shape context method are not imposed at this level. Specifically, the shape context method of finding points is not used; the points in our second layer are those selected by Adaboost in the first layer. Having defined the PSR features in Layer 2, $\{s_{i,k}, i = 1, ..., N, k = 1, ..., T^m\}$, we then feed these features to Adaboost to obtain PSR weak classifiers. Figure 3 shows a graphical overview of this method.

The Adaboost learning method used in Layer 2 is similar to the method used in Layer 1 except for the calculation of distance between a PSR feature, $s_{ii,k}$, and a training sample, I_i, given the feature set $\{s_{i,k}, k = 1, ..., T^m\}$. In Layer 1 this is defined as the smallest distance between features and the sample. In Layer 2, we define distance as $D(s_{ii,k}, I_i) = D(s_{ii,k}, s_{i,k})$ using the χ^2 statistic metric.

After selecting T^s (The superscript 's' is used to indicate the PSR layer.)PSR-based weak classifiers, we build the Layer 2 strong classifier:

$$H^s = \{ \begin{matrix} 1, \sum_{t=1}^{T^s} \alpha_t h_t^s > \Omega^s \\ 0, otherwise \end{matrix} \quad (4)$$

[1]Due to the large number of features required for training, the boosting process can be slow. Given that the main computational burden lies in the calculation and sorting of distances between features, it is possible to speed up the training process by pre-computing and pre-sorting this distance matrix.



(a) Find good features (b) Compute spatial relationships

Figure 3. Computing spatial relationships between features. (a) Dots and crosses represent key points on an image. The best matching features from Layer 1 are extracted and indexed (1-4). (b) To find the PSR feature for Feature 1, distances and orientations relative to the other 3 features are calculated and then represented as a histogram.

where Ω^s is a again a recognition threshold for classification.

Fig 2 shows a subset of the good features (red squares) selected in Layer 1 on several sample images of motorbikes and airplanes. It also shows some PSR features (green circles) selected by AdaBoost in Layer 2. As can be seen, common parts of an object class (e.g., the tailfin of the airplanes) seem to be consistently represented by PSR features.

4. Recognition

Recognition is accomplished in two steps, with each step corresponding to the cascaded use of the two strong classifiers. First, we detect key points in a test image and extract PCA-SIFT and shape context features. For each weak classifier, h_k^m, and its associated feature, f_k^m, selected by Adaboost in Layer 1 (Equation 2), we find the corresponding feature in the test image that has the smallest distance d_k^m to feature f_k^m. We then compare this minimum distance to the classification threshold, and a binary decision is made using Equation 1. After all of the weak classifiers have been processed, we test if the output of the strong classifier H^m exceeds the Ω^m threshold. This threshold should be set such as to maximize the acceptance of positive images while minimizing probability of accepting negative images. If a test image is accepted by the first layer, processing passes to Step 2 for the final decision. For the second step, we locate the features in the test image that best match the PSR features selected from Adaboost in Layer 2 (Equation 4). If the output of this match is above the threshold for this strong classifier, the test image is classified as belonging to the target object class.

5. Experiments

Experiments were carried out using 100 positive and 100 negative images for both the training and testing sets. All sample images were randomly selected (from either the Caltech or GRAZ datasets) and were not preprocessed in any

Dataset	SIFT	PCA-SIFT	Shape Context	Multiple features	Fergus[4]	Opelt[12]	Thureson[13]
Motorbikes	95.0	98.3	87.4	99.0	92.5	92.2	93.2
Airplanes	94.4	97.9	90.0	98.3	90.2	88.9	83.8
Faces	99.7	99.7	64.7	99.7	96.4	93.5	83.1

Table 1. ROC equal error rates using the original Caltech database. Results are shown for SIFT features only, PCA-SIFT features only, shape context features only, and PCA-SIFT and shape context features combined (multiple features). These results are also compared to other recent methods reporting equal error rates using this database.

way. PCA-SIFT features were defined using the PCA space described in [6], and images were upsampled by a factor of 2 in order to maximize the number of key points and thereby improve recognition. Those smallest scale PCA-SIFT features were discarded during Adaboost training as this was found to reduce greatly the training time without affecting recognition performance. However, all PCA-SIFT features were used during testing and when matching features. The shape context features were computed based on $n = 200$ sample points detected in the edge image. The shape context descriptor used 5 distance bins and 12 orientation bins in log-polar space. The same settings were used for describing the PSR features. The number of features selected in Layer 1 ranged from 100 to 300, and in Layer 2 from 100 to 500. Recognition performance is reported as receiver-operating characteristic (ROC) curves and as equal error rates (true positive rate = false positive rate).

5.1 Caltech database

Our first experiment used the Caltech database [2]. It has 6 classes of objects: motorbikes, airplanes, faces, cars(side), cars(rear) and spotted cats, and a background set. Object class recognition is peformed between an object class and the non-object class (backgrounds).

Table 1 summarizes our results. For all datasets, the combination of PCA-SIFT and shape context features resulted in very high recognition rates, clearly outperforming recent state-of-the-art methods. We did not include PSR features in this experiment as recognition performance was already at ceiling and could not be meaningfully improved with additional information. Note also that the high multi-feature recognition rate for this dataset can be attributed mainly to the use of PCA-SIFT features. Further investigation of this matter revealed that the background and object images in the Caltech database are not equally complex. As a result of this unequal complexity, far fewer key points can be found on the background images compared to the object images, and this difference benefits local feature-based methods of object recognition. The following experiments were conducted specifically to explore this relationship.

[2] Available at: http://www.robots.ox.ac.uk/~vgg/data.html

5.2. Complex background

The difficulty of object class recognition depends on the complexity of the non-object class as well as the complexity of the target object class. If the non-object class is so simple that it contains very few features, then the object class will be easily recognized regardless of the variability in the class. To better equate the object and non-object classes, we created a new background set consisting of 620 non-object images [3] obtained using Google. Note that this dataset, consisting of natural objects, manmade objects, indoor scenes, and outdoor scenes, is far more variable than the background images in the Caltech dataset, which consisted of images captured around the Caltech campus and vision lab.

Background image size	200	300	400	500
Average #key points	231	491	850	1343
Equal error rate	99.3	94.6	89.2	79.8

Table 2. Performance of boosting with PCA-SIFT features decreases rapidly as the number of features on non-object samples increases.

Our first experiment describes the relationship between recognition rate and the number of key points in the images. We used the motorbike set from the Caltech database combined with our new background set. Bilinear interpolation was used in each experiment to normalize the background image such that the longer side was 500, 400, 300, or 200 pixels, respectively. Each experiment also used the same sets of training and testing samples. Table 2 shows recognition performance as a function of image size using boosting with only PCA-SIFT features. As the number of key points on the background images increase, PCA-SIFT features become less able to discriminate motorbikes from non-objects.

To determine how this relationship affected our Section 5.1 experiment, we repeated that experiment using our new background set. The background images were normalized so that their longer sides were 500 pixels, making them comparable in size to the object images. The results from this experiment appear in Table 3 and Figure 4. When

[3] Available at: http://www.cs.sunysb.edu/~samaras/data/background.zip

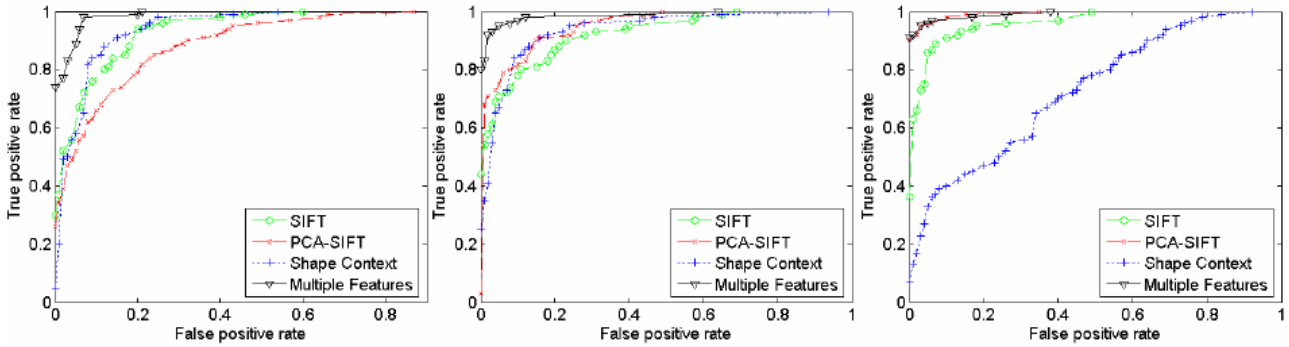


Figure 4. ROC curves using the Caltech database and our complex backgrounds.

Dataset	SIFT	PCA-SIFT	Shape Context	Multiple features
Motorbikes	84.5	79.8	88.0	94.0
Airplanes	82.6	87.3	87.8	95.0
Faces	90.6	95.5	65.4	95.7

Table 3. ROC equal error rates using the Caltech datasets and our complex background dataset. Note that the recognition rate using PCA-SIFT features is reduced relative to the Table 1 data, and that our multiple feature method consistently outperforms any single-feature recognition method.

background complexity is increased, neither SIFT nor PCA-SIFT can achieve good recognition performance across all three of the tested Caltech datasets. Unlike the Section 5.1 experiment, we also now see the shape context method outperforming the PCA-SIFT method for the motorbike dataset, and roughly equaling PCA-SIFT performance in the airplane dataset. The fact that PCA-SIFT features are preferable for some datasets (faces) and shape context features are preferable for other datasets (motorbikes) suggests that our multi-feature model should yield uniformly superior recognition performance, which indeed was the case. When local texture and global shape features are combined, equal error recognition rates average 95% across the three tested datasets despite increases in background complexity and variability.

These findings validate our feature-combination approach. Different class recognition methods are best suited to different datasets. As discussed above, datasets having complex and highly variable non-object backgrounds create problems for methods relying on local features. Methods relying on global features, such as shape context, have problems when the object to be detected appears in a cluttered scene. Because faces in the Caltech database often appeared in cluttered scenes, this latter relationship explains in part why the results in [13] for face recognition were much worse than for motorbike recognition. An advantage of our feature-combination method, in addition to its superior performance, is that it is stable over different datasets. By including both local and global information in the feature pool available to Adaboost, our model can automatically select the optimal features for a given object and background dataset. It will therefore tend to use global features when the shape of an object can be well defined, and local features when the object shape is obscured. In fact, for the face dataset, 297 of the 300 features chosen by our model were local features.

5.3 GRAZ database

Because the object class recognition rates are already very high using the Caltech database and our multiple feature method, it would not be possible to achieve meaningfully higher rates by adding PSR features. We therefore chose to test the contribution of these Layer 2 features by switching to a more challenging dataset. Toward this end we chose the GRAZ bicycle database [4]. Objects in this database have high class variability, the backgrounds are cluttered and there are large view changes. Sample images also depict either single or multiple views of the target object class, and sometimes only isolated parts of an object (e.g. a bicycle wheel). We divided the GRAZ database into two subsets: BIKE contains samples with exactly one instance of the object class per image, and BIKES contains samples with multiple object instances. Note that because the focus of this paper is on whole object recognition, we excluded those samples showing only a part of an object.

[4] Available at: http://www.emt.tugraz.at/~pinz/data/

Our expectation was that the object variability and background clutter in this dataset would limit the usefulness of both local features and shape context features. Under these conditions, PSR features might improve recognition performance. Table 4 shows these results as ROC equal error rates, broken down by sample subset. As expected, adding PSR features improved performance in the BIKE dataset over our already high multiple feature recognition rate. Information about the spatial relationships between local features is clearly beneficial to object class recognition. However, as is also clear from Table 4, this benefit is limited to the BIKE dataset. We attribute this result to the global nature of the PSR feature. When multiple instances of an object appear in the sample (as in the BIKES dataset), spatial relationships between the local features of two different bikes may be computed, resulting in the selection of poor PSR features. Despite this limitation, our multi-layered model yielded better recognition performance than the 86.5 rate reported by Opelt et al. [12], although it should also be noted that they did not segregate their data by single and multiple-instance samples.

Dataset	Multiple features	Adding PSRs
BIKE	85.7	90.0
BIKES	87.2	86.0

Table 4. ROC equal error rates using subsets of the GRAZ bicycle database. The BIKE samples show only a single bike object; the BIKES samples show multiple bikes in the image. Adding PSRs improves recognition in the BIKE dataset.

6. Conclusions

We have presented a multi-layered Adaboost model for object class recognition. Our model makes two important contributions to this literature. First, it combines information from three very different feature sources: local texture (PCA-SIFT), global shape (shape context), and the spatial relationships between selected local features (PSRs). Although previous approaches have adopted a similar feature-combination method, our method is the first to use such a diverse pool of features (both local and global). Second, a novelty of our method is that we use Adaboost to select from this pool those features that are best suited to a given recognition task. Rather than being specific to a particular type of image database, our method is therefore highly versatile, able to adapt to the demands of different datasets by simply choosing different features. We compared our model to other state-of-the-art methods and found that it uniformly outperforms these previous approaches, particularly under conditions of high non-object class complexity (which we showed to be singularly detrimental to local feature models). This combination of performance and versatility makes our method preferable to any single-feature method of object class detection. Future work will seek to further refine our component features so as to better optimize recognition performance.

Acknowledgements

This work was supported by grants from the Army Research Office (DAAD19-03-1-0039) to G.J.Z., U.S. Department of Justice (2004-DD-BX-1224) and National Science Foundation (ACI-0313184).

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV02*, page IV: 113 ff., 2002.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, April 2002.

[3] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *ICCV03*, pages 634–640, 2003.

[4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR03*, pages II: 264–271, 2003.

[5] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[6] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *CVPR04*, pages II: 506–513, 2004.

[7] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In *CVPR04*, pages II: 53–60, 2004.

[8] D. Lowe. Object recognition from local scale-invariant features. In *ICCV99*, pages 1150–1157, 1999.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004.

[10] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR03*, pages II: 257–263, 2003.

[11] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *PAMI*, 23(4):349–361, April 2001.

[12] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV04*, pages Vol II: 71–84, 2004.

[13] J. Thureson and S. Carlsson. Appearance based qualitative image description for object class recognition. In *ECCV04*, pages Vol II: 518–529, 2004.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR01*, pages I:511–518, 2001.

[15] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, May 2004.

[16] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV00*, pages I: 18–32, 2000.