Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories

Svetlana Lazebnik[1]
slazebni@uiuc.edu
[1]*Beckman Institute*
University of Illinois

Cordelia Schmid[2]
Cordelia.Schmid@inrialpes.fr
[2]*INRIA Rhône-Alpes*
Montbonnot, France

Jean Ponce[1,3]
ponce@cs.uiuc.edu
[3]*Ecole Normale Supérieure*
Paris, France

Abstract

This paper presents a method for recognizing scene categories based on approximate global geometric correspondence. This technique works by partitioning the image into increasingly fine sub-regions and computing histograms of local features found inside each sub-region. The resulting "spatial pyramid" is a simple and computationally efficient extension of an orderless bag-of-features image representation, and it shows significantly improved performance on challenging scene categorization tasks. Specifically, our proposed method exceeds the state of the art on the Caltech-101 database and achieves high accuracy on a large database of fifteen natural scene categories. The spatial pyramid framework also offers insights into the success of several recently proposed image descriptions, including Torralba's "gist" and Lowe's SIFT descriptors.

1. Introduction

In this paper, we consider the problem of recognizing the semantic category of an image. For example, we may want to classify a photograph as depicting a scene (forest, street, office, etc.) or as containing a certain object of interest. For such whole-image categorization tasks, *bag-of-features* methods, which represent an image as an orderless collection of local features, have recently demonstrated impressive levels of performance [7, 22, 23, 25]. However, because these methods disregard all information about the spatial layout of the features, they have severely limited descriptive ability. In particular, they are incapable of capturing shape or of segmenting an object from its background. Unfortunately, overcoming these limitations to build effective structural object descriptions has proven to be quite challenging, especially when the recognition system must be made to work in the presence of heavy clutter, occlusion, or large viewpoint changes. Approaches based on generative part models [3, 5] and geometric correspondence search [1, 11] achieve robustness at significant computational expense. A more efficient approach is to augment a basic bag-of-features representation with pairwise relations between neighboring local features, but existing implementations of this idea [11, 17] have yielded inconclusive results. One other strategy for increasing robustness to geometric deformations is to increase the level of invariance of local features (e.g., by using affine-invariant detectors), but a recent large-scale evaluation [25] suggests that this strategy usually does not pay off.

Though we remain sympathetic to the goal of developing robust and geometrically invariant structural object representations, we propose in this paper to revisit "global" non-invariant representations based on aggregating statistics of local features over fixed subregions. We introduce a kernel-based recognition method that works by computing rough geometric correspondence on a global scale using an efficient approximation technique adapted from the *pyramid matching* scheme of Grauman and Darrell [7]. Our method involves repeatedly subdividing the image and computing histograms of local features at increasingly fine resolutions. As shown by experiments in Section 5, this simple operation suffices to significantly improve performance over a basic bag-of-features representation, and even over methods based on detailed geometric correspondence.

Previous research has shown that statistical properties of the scene considered in a holistic fashion, without any analysis of its constituent objects, yield a rich set of cues to its semantic category [13]. Our own experiments confirm that global representations can be surprisingly effective not only for identifying the overall scene, but also for categorizing images as containing specific objects, even when these objects are embedded in heavy clutter and vary significantly in pose and appearance. This said, we do not advocate the direct use of a global method for object recognition (except for very restricted sorts of imagery). Instead, we envision a subordinate role for this method. It may be used to capture the "gist" of an image [21] and to inform the subsequent

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

search for specific objects (e.g., if the image, based on its global description, is likely to be a highway, we have a high probability of finding a car, but not a toaster). In addition, the simplicity and efficiency of our method, in combination with its tendency to yield unexpectedly high recognition rates on challenging data, could make it a good baseline for "calibrating" new datasets and for evaluating more sophisticated recognition approaches.

2. Previous Work

In computer vision, histograms have a long history as a method for image description (see, e.g., [16, 19]). Koenderink and Van Doorn [10] have generalized histograms to *locally orderless images*, or histogram-valued scale spaces (i.e., for each Gaussian aperture at a given location and scale, the locally orderless image returns the histogram of image features aggregated over that aperture). Our spatial pyramid approach can be thought of as an alternative formulation of a locally orderless image, where instead of a Gaussian scale space of apertures, we define a fixed hierarchy of rectangular windows. Koenderink and Van Doorn have argued persuasively that locally orderless images play an important role in visual perception. Our retrieval experiments (Fig. 4) confirm that spatial pyramids can capture perceptually salient features and suggest that "locally orderless matching" may be a powerful mechanism for estimating overall perceptual similarity between images.

It is important to contrast our proposed approach with *multiresolution histograms* [8], which involve repeatedly subsampling an image and computing a global histogram of pixel values at each new level. In other words, a multiresolution histogram varies the resolution at which the features (intensity values) are computed, but the histogram resolution (intensity scale) stays fixed. We take the opposite approach of fixing the resolution at which the features are computed, but varying the spatial resolution at which they are aggregated. This results in a higher-dimensional representation that preserves more information (e.g., an image consisting of thin black and white stripes would retain two modes at every level of a spatial pyramid, whereas it would become indistinguishable from a uniformly gray image at all but the finest levels of a multiresolution histogram). Finally, unlike a multiresolution histogram, a spatial pyramid, when equipped with an appropriate kernel, can be used for approximate geometric matching.

The operation of "subdivide and disorder" — i.e., partition the image into subblocks and compute histograms (or histogram statistics, such as means) of local features in these subblocks — has been practiced numerous times in computer vision, both for global image description [6, 18, 20, 21] and for local description of interest regions [12]. Thus, though the operation itself seems fundamental, previous methods leave open the question of what is the right subdivision scheme (although a regular 4×4 grid seems to be the most popular implementation choice), and what is the right balance between "subdividing" and "disordering." The spatial pyramid framework suggests a possible way to address this issue: namely, the best results may be achieved when multiple resolutions are combined in a principled way. It also suggests that the reason for the empirical success of "subdivide and disorder" techniques is the fact that they actually perform approximate geometric matching.

3. Spatial Pyramid Matching

We first describe the original formulation of pyramid matching [7], and then introduce our application of this framework to create a *spatial pyramid* image representation.

3.1. Pyramid Match Kernels

Let X and Y be two sets of vectors in a d-dimensional feature space. Grauman and Darrell [7] propose *pyramid matching* to find an approximate correspondence between these two sets. Informally, pyramid matching works by placing a sequence of increasingly coarser grids over the feature space and taking a weighted sum of the number of matches that occur at each level of resolution. At any fixed resolution, two points are said to match if they fall into the same cell of the grid; matches found at finer resolutions are weighted more highly than matches found at coarser resolutions. More specifically, let us construct a sequence of grids at resolutions $0, \ldots, L$, such that the grid at level ℓ has 2^ℓ cells along each dimension, for a total of $D = 2^{d\ell}$ cells. Let H_X^ℓ and H_Y^ℓ denote the histograms of X and Y at this resolution, so that $H_X^\ell(i)$ and $H_Y^\ell(i)$ are the numbers of points from X and Y that fall into the ith cell of the grid. Then the number of matches at level ℓ is given by the *histogram intersection* function [19]:

$$\mathcal{I}(H_X^\ell, H_Y^\ell) = \sum_{i=1}^{D} \min\left(H_X^\ell(i), H_Y^\ell(i)\right). \qquad (1)$$

In the following, we will abbreviate $\mathcal{I}(H_X^\ell, H_Y^\ell)$ to $\mathcal{I}^\ell$.

Note that the number of matches found at level ℓ also includes all the matches found at the finer level $\ell + 1$. Therefore, the number of *new* matches found at level ℓ is given by $\mathcal{I}^\ell - \mathcal{I}^{\ell+1}$ for $\ell = 0, \ldots, L-1$. The weight associated with level ℓ is set to $\frac{1}{2^{L-\ell}}$, which is inversely proportional to cell width at that level. Intuitively, we want to penalize matches found in larger cells because they involve increasingly dissimilar features. Putting all the pieces together, we

get the following definition of a *pyramid match kernel*:

$$\kappa^L(X,Y) = \mathcal{I}^L + \sum_{\ell=0}^{L-1} \frac{1}{2^{L-\ell}} \left(\mathcal{I}^\ell - \mathcal{I}^{\ell+1}\right) \quad (2)$$

$$= \frac{1}{2^L}\mathcal{I}^0 + \sum_{\ell=1}^{L} \frac{1}{2^{L-\ell+1}}\mathcal{I}^\ell \,. \quad (3)$$

Both the histogram intersection and the pyramid match kernel are Mercer kernels [7].

3.2. Spatial Matching Scheme

As introduced in [7], a pyramid match kernel works with an orderless image representation. It allows for precise matching of two collections of features in a high-dimensional appearance space, but discards all spatial information. This paper advocates an "orthogonal" approach: perform pyramid matching in the two-dimensional image space, and use traditional clustering techniques in feature space.[1] Specifically, we quantize all feature vectors into M discrete types, and make the simplifying assumption that only features of the same type can be matched to one another. Each channel m gives us two sets of two-dimensional vectors, X_m and Y_m, representing the coordinates of features of type m found in the respective images. The final kernel is then the sum of the separate channel kernels:

$$K^L(X,Y) = \sum_{m=1}^{M} \kappa^L(X_m, Y_m) \,. \quad (4)$$

This approach has the advantage of maintaining continuity with the popular "visual vocabulary" paradigm — in fact, it reduces to a standard bag of features when $L = 0$.

Because the pyramid match kernel (3) is simply a weighted sum of histogram intersections, and because $c\,\min(a,b) = \min(ca,cb)$ for positive numbers, we can implement K^L as a single histogram intersection of "long" vectors formed by concatenating the appropriately weighted histograms of all channels at all resolutions (Fig. 1). For L levels and M channels, the resulting vector has dimensionality $M\sum_{\ell=0}^{L} 4^\ell = M\frac{1}{3}(4^{L+1}-1)$. Several experiments reported in Section 5 use the settings of $M = 400$ and $L = 3$, resulting in 34000-dimensional histogram intersections. However, these operations are efficient because the histogram vectors are extremely sparse (in fact, just as in [7], the computational complexity of the kernel is linear in the number of features). It must also be noted that we did not observe any significant increase in performance beyond $M = 200$ and $L = 2$, where the concatenated histograms are only 4200-dimensional.

[1] In principle, it is possible to integrate geometric information directly into the original pyramid matching framework by treating image coordinates as two extra dimensions in the feature space.

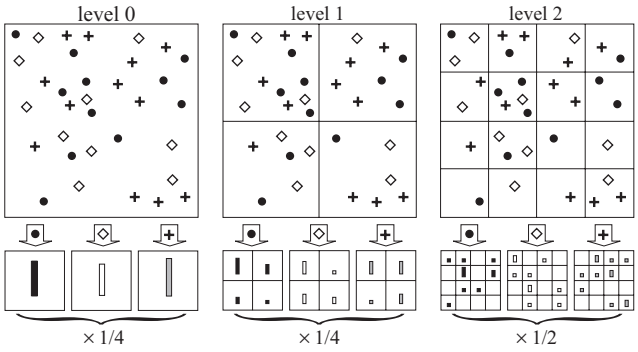

Figure 1. Toy example of constructing a three-level pyramid. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, we subdivide the image at three different levels of resolution. Next, for each level of resolution and each channel, we count the features that fall in each spatial bin. Finally, we weight each spatial histogram according to eq. (3).

The final implementation issue is that of normalization. For maximum computational efficiency, we normalize all histograms by the total weight of all features in the image, in effect forcing the total number of features in all images to be the same. Because we use a dense feature representation (see Section 4), and thus do not need to worry about spurious feature detections resulting from clutter, this practice is sufficient to deal with the effects of variable image size.

4. Feature Extraction

This section briefly describes the two kinds of features used in the experiments of Section 5. First, we have so-called "weak features," which are oriented edge points, i.e., points whose gradient magnitude in a given direction exceeds a minimum threshold. We extract edge points at two scales and eight orientations, for a total of $M = 16$ channels. We designed these features to obtain a representation similar to the "gist" [21] or to a global SIFT descriptor [12] of the image.

For better discriminative power, we also utilize higher-dimensional "strong features," which are SIFT descriptors of 16×16 pixel patches computed over a grid with spacing of 8 pixels. Our decision to use a dense regular grid instead of interest points was based on the comparative evaluation of Fei-Fei and Perona [4], who have shown that dense features work better for scene classification. Intuitively, a dense image description is necessary to capture uniform regions such as sky, calm water, or road surface (to deal with low-contrast regions, we skip the usual SIFT normalization procedure when the overall gradient magnitude of the patch is too weak). We perform k-means clustering of a random subset of patches from the training set to form a visual vocabulary. Typical vocabulary sizes for our experiments are $M = 200$ and $M = 400$.

Figure 2. Example images from the scene category database. The starred categories originate from Oliva and Torralba [13].

	Weak features ($M = 16$)		Strong features ($M = 200$)		Strong features ($M = 400$)	
L	Single-level	Pyramid	Single-level	Pyramid	Single-level	Pyramid
0 (1×1)	45.3 ±0.5		72.2 ±0.6		74.8 ±0.3	
1 (2×2)	53.6 ±0.3	56.2 ±0.6	77.9 ±0.6	79.0 ±0.5	78.8 ±0.4	80.1 ±0.5
2 (4×4)	61.7 ±0.6	64.7 ±0.7	79.4 ±0.3	**81.1** ±0.3	79.7 ±0.5	**81.4** ±0.5
3 (8×8)	63.3 ±0.8	**66.8** ±0.6	77.2 ±0.4	80.7 ±0.3	77.2 ±0.5	81.1 ±0.6

Table 1. Classification results for the scene category database (see text). The highest results for each kind of feature are shown in bold.

5. Experiments

In this section, we report results on three diverse datasets: fifteen scene categories [4], Caltech-101 [3], and Graz [14]. We perform all processing in grayscale, even when color images are available. All experiments are repeated ten times with different randomly selected training and test images, and the average of per-class recognition rates[2] is recorded for each run. The final result is reported as the mean and standard deviation of the results from the individual runs. Multi-class classification is done with a support vector machine (SVM) trained using the one-versus-all rule: a classifier is learned to separate each class from the rest, and a test image is assigned the label of the classifier with the highest response.

[2]The alternative performance measure, the percentage of all test images classified correctly, can be biased if test set sizes for different classes vary significantly. This is especially true of the Caltech-101 dataset, where some of the "easiest" classes are disproportionately large.

5.1. Scene Category Recognition

Our first dataset (Fig. 2) is composed of fifteen scene categories: thirteen were provided by Fei-Fei and Perona [4] (eight of these were originally collected by Oliva and Torralba [13]), and two (industrial and store) were collected by ourselves. Each category has 200 to 400 images, and average image size is 300×250 pixels. The major sources of the pictures in the dataset include the COREL collection, personal photographs, and Google image search. This is one of the most complete scene category dataset used in the literature thus far.

Table 1 shows detailed results of classification experiments using 100 images per class for training and the rest for testing (the same setup as [4]). First, let us examine the performance of strong features for $L = 0$ and $M = 200$, corresponding to a standard bag of features. Our classification rate is 72.2% (74.7% for the 13 classes inherited from Fei-Fei and Perona), which is much higher than their best results of 65.2%, achieved with an orderless method and a feature set comparable to ours. We conjecture that Fei-Fei and Perona's approach is disadvantaged by its re-

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

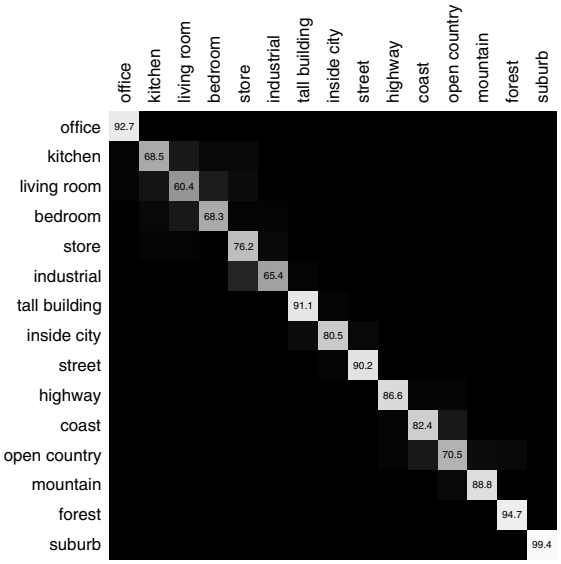

Figure 3. Confusion table for the scene category dataset. Average classification rates for individual classes are listed along the diagonal. The entry in the ith row and jth column is the percentage of images from class i that were misidentified as class j.

liance on latent Dirichlet allocation (LDA) [2], which is essentially an unsupervised dimensionality reduction technique and as such, is not necessarily conducive to achieving the highest classification accuracy. To verify this, we have experimented with probabilistic latent semantic analysis (pLSA) [9], which attempts to explain the distribution of features in the image as a mixture of a few "scene topics" or "aspects" and performs very similarly to LDA in practice [17]. Following the scheme of Quelhas et al. [15], we run pLSA in an unsupervised setting to learn a 60-aspect model of half the training images. Next, we apply this model to the other half to obtain probabilities of topics given each image (thus reducing the dimensionality of the feature space from 200 to 60). Finally, we train the SVM on these reduced features and use them to classify the test set. In this setup, our average classification rate drops to 63.3% from the original 72.2%. For the 13 classes inherited from Fei-Fei and Perona, it drops to 65.9% from 74.7%, which is now very similar to their results. Thus, we can see that latent factor analysis techniques can adversely affect classification performance, which is also consistent with the results of Quelhas et al. [15].

Next, let us examine the behavior of spatial pyramid matching. For completeness, Table 1 lists the performance achieved using just the highest level of the pyramid (the "single-level" columns), as well as the performance of the complete matching scheme using multiple levels (the "pyramid" columns). For all three kinds of features, results improve dramatically as we go from $L = 0$ to a multi-level setup. Though matching at the highest pyramid level seems to account for most of the improvement, using all the levels together confers a statistically significant benefit. For strong features, single-level performance actually drops as we go from $L = 2$ to $L = 3$. This means that the highest level of the $L = 3$ pyramid is too finely subdivided, with individual bins yielding too few matches. Despite the diminished discriminative power of the highest level, the performance of the entire $L = 3$ pyramid remains essentially identical to that of the $L = 2$ pyramid. This, then, is the main advantage of the spatial pyramid representation: because it combines multiple resolutions in a principled fashion, it is robust to failures at individual levels.

It is also interesting to compare performance of different feature sets. As expected, weak features do not perform as well as strong features, though in combination with the spatial pyramid, they can also achieve acceptable levels of accuracy (note that because weak features have a much higher density and much smaller spatial extent than strong features, their performance continues to improve as we go from $L = 2$ to $L = 3$). Increasing the visual vocabulary size from $M = 200$ to $M = 400$ results in a small performance increase at $L = 0$, but this difference is all but eliminated at higher pyramid levels. Thus, we can conclude that the coarse-grained geometric cues provided by the pyramid have more discriminative power than an enlarged visual vocabulary. Of course, the optimal way to exploit structure both in the image and in the feature space may be to combine them in a unified multiresolution framework; this is subject for future research.

Fig. 3 shows a confusion table between the fifteen scene categories. Not surprisingly, confusion occurs between the indoor classes (kitchen, bedroom, living room), and also between some natural classes, such as coast and open country. Fig. 4 shows examples of image retrieval using the spatial pyramid kernel and strong features with $M = 200$. These examples give a sense of the kind of visual information captured by our approach. In particular, spatial pyramids seem successful at capturing the organization of major pictorial elements or "blobs," and the directionality of dominant lines and edges. Because the pyramid is based on features computed at the original image resolution, even high-frequency details can be preserved. For example, query image (b) shows white kitchen cabinet doors with dark borders. Three of the retrieved "kitchen" images contain similar cabinets, the "office" image shows a wall plastered with white documents in dark frames, and the "inside city" image shows a white building with darker window frames.

5.2. Caltech-101

Our second set of experiments is on the Caltech-101 database [3] (Fig. 5). This database contains from 31 to 800 images per category. Most images are medium resolution, i.e., about 300×300 pixels. Caltech-101 is probably the most diverse object database available today, though it

Figure 4. Retrieval from the scene category database. The query images are on the left, and the eight images giving the highest values of the spatial pyramid kernel (for $L = 2, M = 200$) are on the right. The actual class of incorrectly retrieved images is listed below them.

is not without shortcomings. Namely, most images feature relatively little clutter, and the objects are centered and occupy most of the image. In addition, a number of categories, such as minaret (see Fig. 5), are affected by "corner" artifacts resulting from artificial image rotation. Though these artifacts are semantically irrelevant, they can provide stable cues resulting in misleadingly high recognition rates.

We follow the experimental setup of Grauman and Darrell [7] and J. Zhang et al. [25], namely, we train on 30 images per class and test on the rest. For efficiency, we limit the number of test images to 50 per class. Note that, because some categories are very small, we may end up with just a single test image per class. Table 2 gives a breakdown of classification rates for different pyramid levels for weak features and strong features with $M = 200$. The results for $M = 400$ are not shown, because just as for the scene category database, they do not bring any significant improvement. For $L = 0$, strong features give 41.2%, which is slightly below the 43% reported by Grauman and Darrell. Our best result is 64.6%, achieved with strong features at $L = 2$. This exceeds the highest classification rate previously published,[3] that of 53.9% reported by J. Zhang et al. [25]. Berg et al. [1] report 48% accuracy using 15 training images per class. Our average recognition rate with this setup is 56.4%. The behavior of weak features on this database is also noteworthy: for $L = 0$, they give a classification rate of 15.5%, which is consistent with a naive graylevel correlation baseline [1], but in conjunction with a four-level spatial pyramid, their performance rises to 54% — on par with the best results in the literature.

Fig. 5 shows a few of the "easiest" and "hardest" object classes for our method. The successful classes are either dominated by rotation artifacts (like minaret), have very little clutter (like windsor chair), or represent coherent natural "scenes" (like joshua tree and okapi). The least successful classes are either textureless animals (like beaver and cougar), animals that camouflage well in their environment

[3] See, however, H. Zhang et al. [24] in these proceedings, for an algorithm that yields a classification rate of $66.2 \pm 0.5\%$ for 30 training examples, and $59.1 \pm 0.6\%$ for 15 examples.

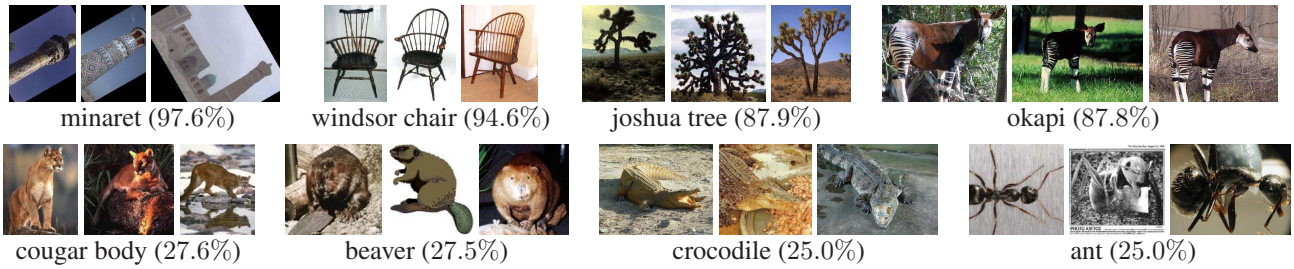

Figure 5. Caltech-101 results. Top: some classes on which our method ($L = 2, M = 200$) achieved high performance. Bottom: some classes on which our method performed poorly.

	Weak features		Strong features (200)	
L	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

Table 2. Classification results for the Caltech-101 database.

class 1 / class 2	class 1 misclassified as class 2	class 2 misclassified as class 1
ketch / schooner	21.6	14.8
lotus / water lily	15.3	20.0
crocodile / crocodile head	10.5	10.0
crayfish / lobster	11.3	9.1
flamingo / ibis	9.5	10.4

Table 3. Top five confusions for our method ($L = 2, M = 200$) on the Caltech-101 database.

Class	$L = 0$	$L = 2$	Opelt [14]	Zhang [25]
Bikes	82.4 ±2.0	86.3 ±2.5	86.5	92.0
People	79.5 ±2.3	82.3 ±3.1	80.8	88.0

Table 4. Results of our method ($M = 200$) for the Graz database and comparison with two existing methods.

(like crocodile), or "thin" objects (like ant). Table 3 shows the top five of our method's confusions, all of which are between closely related classes.

To summarize, our method has outperformed both state-of-the-art orderless methods [7, 25] and methods based on precise geometric correspondence [1]. Significantly, all these methods rely on sparse features (interest points or sparsely sampled edge points). However, because of the geometric stability and lack of clutter of Caltech-101, dense features combined with global spatial relations seem to capture more discriminative information about the objects.

5.3. The Graz Dataset

As seen from Sections 5.1 and 5.2, our proposed approach does very well on global scene classification tasks, or on object recognition tasks in the absence of clutter with most of the objects assuming "canonical" poses. However, it was not designed to cope with heavy clutter and pose changes. It is interesting to see how well our algorithm can do by exploiting the global scene cues that still remain under these conditions. Accordingly, our final set of experiments is on the Graz dataset [14] (Fig. 6), which is characterized by high intra-class variation. This dataset has two object classes, bikes (373 images) and persons (460 images), and a background class (270 images). The image resolution is 640×480, and the range of scales and poses at which exemplars are presented is very diverse, e.g., a "person" image may show a pedestrian in the distance, a side view of a complete body, or just a closeup of a head. For this database, we perform two-class detection (object vs. background) using an experimental setup consistent with that of Opelt et al. [14]. Namely, we train detectors for persons and bikes on 100 positive and 100 negative images (of which 50 are drawn from the other object class and 50 from the background), and test on a similarly distributed set. We generate ROC curves by thresholding raw SVM output, and report the ROC equal error rate averaged over ten runs.

Table 4 summarizes our results for strong features with $M = 200$. Note that the standard deviation is quite high because the images in the database vary greatly in their level of difficulty, so the performance for any single run is dependent on the composition of the training set (in particular, for $L = 2$, the performance for bikes ranges from 81% to 91%). For this database, the improvement from $L = 0$ to $L = 2$ is relatively small. This makes intuitive sense: when a class is characterized by high geometric variability, it is difficult to find useful global features. Despite this disadvantage of our method, we still achieve results very close to those of Opelt et al. [14], who use a sparse, locally invariant feature representation. In the future, we plan to combine spatial pyramids with invariant features for improved robustness against geometric changes.

6. Discussion

This paper has presented a "holistic" approach for image categorization based on a modification of pyramid match kernels [7]. Our method, which works by repeatedly subdividing an image and computing histograms of image features over the resulting subregions, has shown promising re-

bike person background

Figure 6. The Graz database.

sults on three large-scale, diverse datasets. Despite the simplicity of our method, and despite the fact that it works not by constructing explicit object models, but by using global cues as indirect evidence about the presence of an object, it consistently achieves an improvement over an orderless image representation. This is not a trivial accomplishment, given that a well-designed bag-of-features method can outperform more sophisticated approaches based on parts and relations [25]. Our results also underscore the surprising and ubiquitous power of global scene statistics: even in highly variable datasets, such as Graz, they can still provide useful discriminative information. It is important to develop methods that take full advantage of this information — either as stand-alone scene categorizers, as "context" modules within larger object recognition systems, or as tools for evaluating biases present in newly collected datasets.

Acknowledgments. This research was partially supported by the National Science Foundation under grants IIS-0308087 and IIS-0535152, and the UIUC/CNRS/INRIA collaboration agreement.

References

[1] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. CVPR*, volume 1, pages 26–33, 2005.

[2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop on Generative-Model Based Vision*, 2004. http://www.vision.caltech.edu/Image_Datasets/Caltech101.

[4] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, 2005.

[5] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, volume 2, pages 264–271, 2003.

[6] M. Gorkani and R. Picard. Texture orientation for sorting photos "at a glance". In *IAPR International Conference on Pattern Recognition*, volume 1, pages 459–464, 1994.

[7] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features. In *Proc. ICCV*, 2005.

[8] E. Hadjidemetriou, M. Grossberg, and S. Nayar. Multiresolution histograms and their use in recognition. *IEEE Trans. PAMI*, 26(7):831–847, 2004.

[9] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

[10] J. Koenderink and A. V. Doorn. The structure of locally orderless images. *IJCV*, 31(2/3):159–168, 1999.

[11] S. Lazebnik, C. Schmid, and J. Ponce. A maximum entropy framework for part-based texture and object recognition. In *Proc. ICCV*, 2005.

[12] D. Lowe. Towards a computational model for object recognition in IT cortex. In *Biologically Motivated Computer Vision*, pages 20–31, 2000.

[13] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[14] 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. http://www.emt.tugraz.at/˜pinz/data.

[15] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica, T. Tuytelaars, and L. V. Gool. Modeling scenes with local descriptors and latent aspects. In *Proc. ICCV*, 2005.

[16] B. Schiele and J. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *IJCV*, 36(1):31–50, 2000.

[17] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Proc. ICCV*, 2005.

[18] D. Squire, W. Muller, H. Muller, and J. Raki. Content-based query of of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In *Proceedings of the 11th Scandinavian conference on image analysis*, pages 143–149, 1999.

[19] M. Swain and D. Ballard. Color indexing. *IJCV*, 7(1):11–32, 1991.

[20] M. Szummer and R. Picard. Indoor-outdoor image classification. In *IEEE International Workshop on Content-Based Access of Image and Video Databases*, pages 42–51, 1998.

[21] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proc. ICCV*, 2003.

[22] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proc. ICCV*, volume 1, pages 257–264, 2003.

[23] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *ICPR Workshop on Learning for Adaptable Visual Systems*, 2004.

[24] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proc. CVPR*, 2006.

[25] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classifcation of texture and object categories: An in-depth study. Technical Report RR-5737, INRIA Rhône-Alpes, 2005.