Hierarchical Part-Based Visual Object Categorization

Guillaume Bouchard
Xerox Research Center Europe
6, chemin de Maupertuis
38240 Meylan, France
Guillaume.Bouchard@xrce.xerox.com

Bill Triggs
LEAR, GRAVIR-INRIA
655 av. de l'Europe
38330 Montbonnot, France
Bill.Triggs@inrialpes.fr

Abstract

We propose a generative model that codes the geometry and appearance of generic visual object categories as a loose hierarchy of parts, with probabilistic spatial relations linking parts to subparts, soft assignment of subparts to parts, and scale invariant keypoint based local features at the lowest level of the hierarchy. The method is designed to efficiently handle categories containing hundreds of redundant local features, such as those returned by current keypoint detectors. This robustness allows it to outperform constellation style models, despite their stronger spatial models. The model is initialized by robust bottom-up voting over location-scale pyramids, and optimized by Expectation-Maximization. Training is rapid, and objects do not need to be marked in the training images. Experiments on several popular datasets show the method's ability to capture complex natural object classes.

1. Introduction

In object categorization from digital images, existing geometrical models are typically very specific to a particular object class (for example 3D human body models). There is a need for generic models that are suitable for more general object categories. "Part" or "fragment" based models that combine local image features or regions into loose geometric assemblies offer one possible solution to this [9, 11, 5, 4, 8]. Constellation models [5, 4] provide a probabilistic way to mix the appearance and location of local descriptors. One of their major limitations is the fact that they require an explicit enumeration over possible matchings of model features to image ones. This optimal, but combinatorially expensive, step limits the model to relatively few detected features ('parts'), typically 6 or at most 7. This often means that a good deal of the available image information must be ignored, especially in cases where the objects have many parts, either naturally, or because fine grained local visual features are being used to characterize them. Indeed, such structural approaches often fail to compete with geometry-free "bag of features" style approaches because the latter make better use of the available image information [9, 10, 1]. Hence, it is useful to investigate variants of structural approaches that can efficiently handle models with hundreds of local features.

Secondly, many natural object categories (humans and animals, man made classes with variable forms) have relatively rigid shape locally, but significant shape variability at larger scales, so that nearby object features have strongly correlated positions while more distant ones are much more weakly correlated. This suggests a local parts based representation focused on modelling correlations between feature positions. In fact different parts (groups of features that move together in 3D) often overlap in the image owing to occlusion and parallax effects, so common movement is a more important cue than common (image) position for defining parts. A correlation-based model easily represents this kind of structure. Also, to do these things well, it is natural to include some levels of part hierarchy, with loosely connected parts containing more tightly connected subparts. Hence the overall model becomes a tree-structured graphical model [7].

The current paper proposes a hierarchical model of this sort that is capable of handling hundreds of local feature classes efficiently, so that it is suitable for use with very basic feature detectors. The position of the object in the training images and the model structure are unknown and treated as hidden variables, to be estimated using E-M after a suitable initialization. The method is totally scale-invariant, and all of the model parameters are learned by maximum likelihood, so the only tuning needed is the number of parts at each level. Cross-validation shows that using multi-part models is often advantageous.

Below, we first present the probabilistic model, then explain the learning method, including the initialization and

Published in CVPR'05, pp I 710–715. G. Bouchard was at GRAVIR-INRIA when this work was performed. We gratefully acknowledge the support of the European Union research projects LAVA and PASCAL.

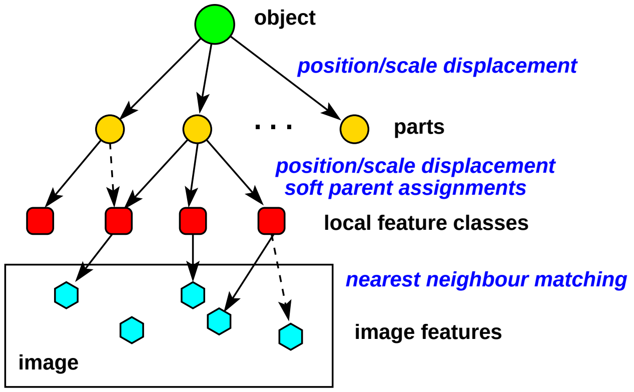
object
position/scale displacement
parts
position/scale displacement
soft parent assignments
local feature classes
nearest neighbour matching
image features
image

Figure 1. The overall structure of our hierarchical object model.

EM steps. Finally experiments on real images show that the model is effective for object categorization.

2. Model Structure

Our model (see figure 1) is a hierarchy of parts and subparts, with the object at the top level, and position-appearance classes of local image features that will be attached to observed image features at the bottom. In each layer of the hierarchy, the parts are softly assigned to parents from the preceding layer. Soft assignment is included mainly to help the model structure adapt to the object class during training. Once the models have been learned, most of the parts tend to have relatively certain parent assignments.

Spatial structure: Parts and their sub-trees are attached to their parents by uncertain spatial transformations. In the experiments below, we have used translations and rescalings, *i.e.* transformations of the form $\mathbf{T}_{qp} = \begin{pmatrix} s & 0 & u \\ 0 & s & v \\ 0 & 0 & 1 \end{pmatrix}$ where s is the relative scale and (u,v) is the relative translation of part p relative to its parent q from the previous layer[1]. We assume that $\mathbf{T}_{qp}$ is sampled from a Normal distribution over translations and a log-Normal distribution over relative scales. We write the corresponding mean and variance symbolically as $\overline{\mathbf{T}}_{qp}$ and $\mathbf{Var}\,(\mathbf{T}_{qp})$. These are model parameters that need to be learned. Formally, $\overline{\mathbf{T}}_{qp}$ is a non-random transformation and $\mathbf{Var}\,(\mathbf{T}_{qp})$ can be thought of as a 3×3 covariance matrix for $(u, v, \log s)$, which is assumed to be diagonal below.

There is a minor complexity relating to the fact that we use soft parent assignments. We do not introduce separate model parameters for the transformation of each part relative to each of its possible parents. This would be a huge number of parameters, and those with low parenting probabilities would not be estimated stably. Instead, we expect parts to represent stable, identifiable regions of the object with their own identities and positions. Parent attributions are uncertain only because it is unclear before training which parent best codes the target part's overall position variability, *i.e.* parts are essentially assigned to the parent whose spatial position variations best explain (covary most strongly over the training set with) their own. To capture this notion, each part p is given just *one* set of mean transformation parameters $\overline{\mathbf{T}}_p$, representing the mean position of p relative the root of the object frame, and a corresponding set of (reduced[2]) variance parameters $\mathbf{Var}\,(\mathbf{T}_p)$. Given a parent attribution q for p, the uncertain transformation $\mathbf{T}_{qp}$ is then sampled with mean $\overline{\mathbf{T}}_{qp} \equiv \overline{\mathbf{T}}_q^{-1}\overline{\mathbf{T}}_p$ and the correspondingly back-transformed variance, which we can denote by $\overline{\mathbf{T}}_q^{-1}(\mathbf{Var}\,(\mathbf{T}_p))$ say. (In our case, this is just the 3×3 $(u, v, \log s)$ covariance $\mathbf{Var}\,(\mathbf{T}_p)$ with its (u,v) block scaled by $1/\overline{s}_q^2$). In this way, the same few parameters control the part's position, whatever its parent assignment. If we suppose that the (random) parent locations $\mathbf{T}_q$ are already known, the part location relative to the object frame is a mixture of random transformations $\mathbf{T}_q\,\mathbf{T}_{qp}$, where $\mathbf{T}_{qp}$ is a random transformation (Gaussian in $(u, v, \log s)$) and the mixture weights are $\boldsymbol{\tau}_p(q)$, the model parameters representing the prior probabilities of p's parent being q:

$$\mathbf{p}_p^{\mathrm{loc}}(\mathbf{T}_p\,|\,\{\mathbf{T}_q\}) = \sum_q \boldsymbol{\tau}_p(q)\,\mathcal{N}(\mathbf{T}_q\mathbf{T}_p^{-1}\,|\,\overline{\mathbf{T}}_{qp}, \mathbf{Var}\,(\mathbf{T}_{qp}))$$

This mixture has the peculiarity that if all of the possible parents q are in their mean positions $\overline{\mathbf{T}}_q$, all of its components coincide exactly — it becomes multimodal only when several parents have nonzero mixing proportions $\boldsymbol{\tau}_p(q)$ and deviate from their means.

Image correspondence: The lowest level of the spatial hierarchy contains elementary parts representing appearance classes of scale-invariant local features, similar to those used in other constellation and bag of features models [12,5,4,3,9,10,1]. When the model is in use, each elementary part acts as a “bonding site” for a nearby image feature of similar appearance. Image features are characterized by their locations (positions and scales) and their appearance vectors $\mathbf{a}$. In the experiments below they are SIFT descriptors calculated over scale-invariant Harris keypoints [9,10], but any other feature / descriptor combination could be used. Each elementary part p has the usual location model $\mathbf{T}_p$, and also a corresponding feature appearance model — here, a Gaussian with model parameters $\overline{\mathbf{a}}_p$ and $\mathbf{Var}\,(\mathbf{a}_p)$. When an image feature is bound to an elementary part, the part's location is instantiated to the feature's location and scale, and its

[1] $\mathbf{T}_{qp}$ is the transformation taking point coordinates in the frame of p to point coordinates in the frame of q, *e.g.* a point at the origin of p with scale 1 has scale s and position (u,v) with respect to q.

[2] This is not the variance of the full uncertain transformation $\mathbf{T}_p$, just the part of this variance that is introduced at the level of part p.

appearance is instantiated to the feature's appearance. The model is designed to support large numbers (hundreds) of elementary parts, only some of which are seen in any given image. So it is important to allow parts to remain effectively unassigned. In practice we nominally assign every part to some feature, but we use a robust assignment probability that effectively discounts any overly distant assignments:

$$\mathbf{p}_p(\mathbf{a},\mathbf{T}) = (1-\boldsymbol{\pi}_p)\,\mathbf{p}_{\text{bkgd}}(\mathbf{a},\mathbf{T}) + \boldsymbol{\pi}_p\,\mathbf{p}_p^{\text{app}}(\mathbf{a})\,\mathbf{p}_p^{\text{loc}}(\mathbf{T})$$

Here: $\mathbf{a}$, $\mathbf{T}$ are the appearance and location of the assigned feature f; $\boldsymbol{\pi}_p$ is a learned inlier proportion for elementary part / feature class p; $\mathbf{p}_{\text{bkgd}}$ is a background model, uniform in appearance and position; $\mathbf{p}_p^{\text{app}}$ is p's appearance model, a Gaussian with mean $\overline{\mathbf{a}}_p$ and variance $\mathbf{Var}\,(\mathbf{a}_p)$; and $\mathbf{p}_p^{\text{loc}}$ is the above mentioned spatial mixture over p's location, parametrized by $\overline{\mathbf{T}}_p, \mathbf{Var}\,(\mathbf{T}_p)$, the corresponding parameters of all p's parents, grandparents, *etc.*, and the corresponding mixing proportions $\boldsymbol{\tau}_p(q)$ for all parents q, *etc.*

When the model is in use, each elementary model part is bound to the single observed image feature that is most probable according to the above likelihood model given the current model parameters. One could also use soft assignments to several nearby image features. This would be more consistent with our overall philosophy, but at present we do not do it, mainly because it would make part-feature matching much less efficient.

During testing, we do nothing to prevent several elementary parts from binding to the same image feature. This is again for efficiency reasons — otherwise a combinatorial matching process would be needed to find the best set of correspondences. However during model training we enforce unique assignments by greedy matching, as otherwise the learned appearance classes of nearby parts tend to merge.

We effectively ignore any unbound features (sometimes even the majority of the features detected). This prevents problems when there are multiple detections of essentially the same feature, but it also means that the current model has no efficient means of representing textured regions. We are currently investigating the use of a Poisson field binding model, where elementary parts can bind to many similar features at once. This also suggests that the model may have problems with hallucinations in very cluttered regions where many types of features occur.

3. Training

The model is fitted to a given image, and also trained over the full training set, using Expectation-Maximization. The model parameters to be adjusted during training are: for each part, the mean and variance of its location and scale, $\overline{\mathbf{T}}_p, \mathbf{Var}\,(\mathbf{T}_p)$, and its vector of parent assignment probabilities $\boldsymbol{\tau}_p$; and for each elementary part, the mean and variance of its feature appearance $\overline{\mathbf{a}}_p, \mathbf{Var}\,(\mathbf{a}_p)$, and its probability of occurrence $\boldsymbol{\pi}_p$. In addition, in each image there are continuous hidden variables to be estimated for the part locations $\mathbf{T}_p$; and discrete ones for the elementary part-to-feature bindings, the background / foreground decision for each bound feature, and the parent assignment for each part.

The E-M processes are straightforward to implement. Once initialized, the method converges in about 5–10 iterations. Training takes around 1 second per image in MATLAB, most of this time being spent in the (currently unoptimized) part-to-feature assignment process. Note that every parameter of the model is learned using E-M: apart from the number of parts in each layer and some thresholds used during initialization, the model has no hand-set parameters, regularization terms, *etc.*

3.1. Instantiating the Model in an Image

The effective cost function has many local minima so a robust instantiation method is needed. We use a hierarchical, heuristic, Hough transform like voting method, based on voting into a position/scale pyramid of possible locations ($\mathbf{T}_p$ values) for each part.

1. For each part q in the penultimate layer of the hierarchy (the direct parents of the elementary parts), each image feature f (with appearance $\mathbf{a}_f$ and location $\mathbf{T}_f$) votes into a position/scale pyramid for q's location $\mathbf{T}_q$, essentially by taking the expected mixture distribution over feature appearances and locations generated by q's elementary subparts, and using it backwards as a likelihood for voting:

$$\text{Vote}_q(\mathbf{T}_q) \;=\; \sum_f \max_p \frac{\boldsymbol{\tau}_p(q)}{w_p}\,\mathbf{p}_p^{\text{app}}(\mathbf{a}_f)\,\mathbf{p}_p^{\text{loc}}(\mathbf{T}_f \,|\, \mathbf{T}_q)$$

   where $w_p \;\equiv\; \sum_f \mathbf{p}_p^{\text{app}}(\mathbf{a}_f)$. Here the sum is over features f, and the maximum is over the elementary parts p whose best parent is q. For speed, the vote uses just the best elementary part attribution p for f. Note that we re-weight each elementary part's vote by the estimated number of image features assigned to its appearance class, $w_p \;=\; \sum_f \mathbf{p}_p^{\text{app}}(\mathbf{a}_f)$. This heuristic helps to suppress common background features and enhance rarer object ones.

2. We work up the spatial tree, combining votes for subpart locations into votes for their parent's locations using the mean location offsets learned for the model:

$$\text{Vote}_q(\mathbf{T}_q) \;=\; S(\sum_p \log(1+\text{Vote}_p(\mathbf{T}_q\,\overline{\mathbf{T}}_{qp}))) \tag{1}$$

   Here, the sum is over subparts p for which q is the most probable parent ($\arg\max_{q'} \boldsymbol{\tau}_p(q') = q$). Again we use hard assignments for speed. The $\log(1 + \ldots)$ nonlinearity makes it harder for high peaks in outlier subparts to dominate the valid contributions of the other subparts. S is a heuristic smoothing function, currently a Gaussian convolution. To be more rigorous, we should smooth by using $\mathbf{T}_p = \mathbf{T}_q \, \mathbf{T}_{qp}$ as the argument and integrating over samples from the uncertain transform $\mathbf{T}_{qp}$.

3. Maxima in the voting pyramid for the top-level part 0 give potential object placements $\mathbf{T}_0$.

4. For the best (or if necessary, for each) maximum, work back down the tree assigning part positions. If the part's voting pyramid has a good maximum near the resulting expected part position, use this value. Otherwise, assume that the part was not seen and use its default offset $\overline{\mathbf{T}}_{qp}$.

Although it can not replace a fully-fledged object detector, this heuristic procedure gives reasonably reliable automatic model initialization results on test sets such as the Caltech ones, even when the object has unknown position and scale and is surrounded by a moderate amount of background clutter. E-M is run after it, so despite the heuristics the resulting instantiation is at least locally optimal for the full cost function.

3.2. Training — Model Initialization

The above procedure assumes a fully trained model. We also automatically initialize the entire training process, so that no manual location or pre-segmentation of the objects in the training images is required. The number of parts in each layer of the hierarchy is currently fixed by hand. The method assumes that each training image contains an instance of the class, but the instance's position and scale can be unknown and some background clutter is tolerated. It works as follows:

1. Heuristically rank the training images according to their expected "quality" as training examples (see below), and use just the best image to estimate the initial model parameters. If this fails we could potentially use the second, third, *etc.*, images, but this has not been necessary in our experiments.

2. Using K-means, cluster all of the features in the initial image into n location-appearance classes, where n is the desired number of elementary parts, and initialize an elementary part at each cluster center. The experiments below actually assume that n is the number of observed features, so that there is one elementary part per observed feature. Some of these elementary parts will correspond to background clutter. We do not currently attempt to remove these. Some feature classes may have the same appearance but different locations. This is intentional: it allows for (small numbers of) repeated features such as eyes or car wheels.

3. Work up the hierarchy, clustering the subpart centers into the desired number of parent part centres, and initializing one parent for each cluster. Cluster membership gives initial (hard) parent assignments for the subparts. The corresponding $\boldsymbol{\tau}$ matrix is initialized to a slightly softer version of these. The cluster centre gives the part location, and the median scale of the part's children gives its initial scale estimate.

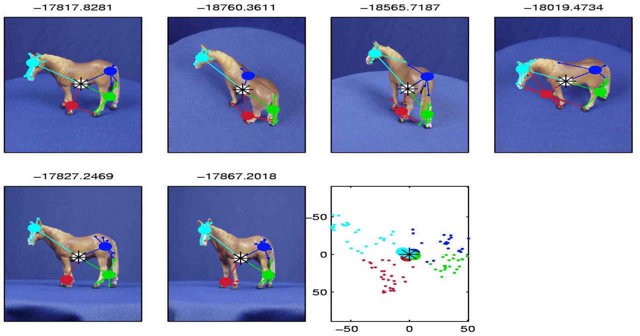

Figure 2. Model fitting on test horse toy images. The bottom right points are the model average positions of the subparts.

The use of a single initial image in the first step could certainly be criticised, but so far our attempts to initialize from averages over several or many images have given much worse results. The critical point seems to be to provide an initial model with cleanly separated appearances and parts, from which a relatively unambiguous training phase can proceed. Averaging tends to confuse the part relationships and produce a less effective overall model.

Our method of ranking the training images by their probable quality as model initializers is as follows:

1. Use K-means to cluster the features from all (positive) training images into about 500 classes. Encode each image as a 500-D signature vector $\mathbf{S}$ (the vector of class counts).

2. Rank the feature classes by an informativeness measure (see below) and select about the 30 most informative ones. Rank the images according to the number of these classes that they contain (*i.e.* the number of classes c for which $\mathbf{S}_c \neq 0$).

For the feature informativeness ranking, we have studied two methods, one supervised, the other unsupervised.

The supervised method requires a negative training set (non-class images) as well as the positive one (class images).

It trains a linear classifier to predict the image class (±1 for positive or negative) from the binarized signature vector ($\mathbf{S} \neq 0$). The features with the highest weights are chosen as the most informative ones. Any appropriate classification method can be used: linear SVM or RVM, LASSO, *etc*.

The unsupervised method is somewhat more heuristic, but it seems to work equally well and it requires no negative images. For each feature, it counts the number of (positive) images in which it occurs *exactly once* (or alternatively, exactly 1–2 times), and chooses the features with the highest counts as the most informative. This works because it selects *distinctive features representing unique object parts*. Many object classes contain such features, whereas background features are much more variable and seldom occur exactly once per image. This method would fail for object classes dominated by repetitive texture though.

4. Experiments

Here we will only test a three-layer model (object → part → feature class). Figure 2 demonstrates the model's ability to handle local image deformations due to parallax, for which rigid matching would fail. We learned the model from 6 images of the same toy horse seen from different viewing positions, using 100 feature classes and 4 parts. The model was then instantiated on 6 test images with the method described above. The change of viewing angle between views is considerable, but the model still finds and locks on to the correct object parts, even if only a few points are found on each one.

Datasets: We used five different image classes from the "Caltech 101 Object Categories" dataset[3] [4], which contains many example images from 101 objects categories, including faces (435 images), leopards (200), motorbikes (800), aeroplanes (800) and side views of cars (123). These datasets have already been used by several groups [12,6,4, 3]. Half of the images in each class were held out for testing.

Some examples of the learned models are shown in figure 3. To test whether the models really managed to learn the most important appearance parameters and spatial interrelationships, and whether they were sufficiently selective for a given object category, we assessed their discriminative power by fitting several class models to unseen test images, using model likelihoods as decision variables. For each class, a decision threshold was computed to minimize the average training error rate. We used 10 EM iterations during training and 5 during testing. Confusion matrices are given in table 1 for the original 7 class Caltech dataset using 200 feature classes for the one-level and two level (with 3 parts) hierarchical models. The number of errors depends on the class, but despite the fact that our models are generative not discriminative, the results seem to be competitive with the state of the art on these datasets [2,5]. The basic partless model is already highly discriminative for these data sets, but using a 3 part model still reduces the error rates by a factor of about two.

one-level, partless model	F.	L.	M.	A.	C.
Faces	189	18	2	8	0
Leopards	0	93	6	1	0
Motorbikes	1	0	379	19	0
Airplanes	0	0	8	392	0
Car sides	0	1	2	13	45

Two-level, 3 part model	F.	L.	M.	A.	C.
Faces	191	14	10	2	0
Leopards	0	95	5	0	0
Motorbikes	1	1	382	15	0
Airplanes	0	0	8	392	0
Car sides	0	1	0	5	55

Table 1. The confusion matrix for part based multiclass categorization on the original Caltech 7 class dataset.

one-level model					
	Acc	Air	Anc	Ant	Bar
Accordion	18	0	0	9	0
Airplanes	0	359	6	35	0
Anchor	0	1	4	12	4
Ant	0	2	1	17	1
Barrel	0	3	1	9	10
Two-level, three part model					
Accordion	25	0	1	1	0
Airplanes	1	384	0	12	0
Anchor	0	3	6	12	0
Ant	0	4	1	18	1
Barrel	0	6	0	8	9

Table 2. Confusion matrix for best-class classifiers based on 80 feature classes, on the first few classes of the Caltech 101 class dataset.

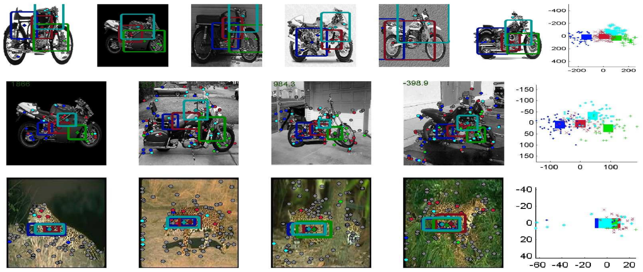

Figure 3. Examples of fits to images from the motorbikes and leopard datasets. The first line shows a close-up of the initialisation based on location/scale voting.

Figure 4 shows that the results are not too sensitive to the

[3] Available at http://www.vision.caltech.edu/feifeili/Datasets.htm

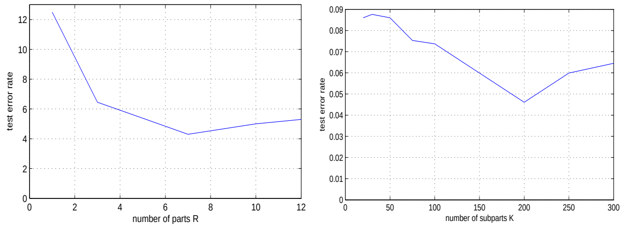

Figure 4. The test error rate of the leaves/faces classifier against the numbers of parts (left) and feature classes (right).

number of parts, although over-fitting starts to worsen the results beyond about 8–10 parts. Relatively large numbers of elementary parts are needed to get optimal results — about 200 in this case.

Soft *vs*. hard assignments: The matrix τ coding for the structure can be constrained to have only ones and zeros, so that a given part can only be generated by a single parent. To illustrate the advantages of soft parenting, for binary classification of motorbikes against background images using 40 training images, 200 feature classes and 4 parts, hard assignment based learning produces a test-set classification rate of 83%, while our standard soft assignments gave 88%. Similar results occur for other datasets and training parameters.

5. Conclusions and Future Work

We have described a multi-layered part-based generative model for category-level visual object recognition using large numbers of local features. The model managed to adapt well to the object categories tested in our supervised classification experiments. Reasons for this include its well-graded spatial flexibility, and the fact that it can efficiently incorporate a large number of interest points, each carrying a worthwhile amount of discriminant information. The resulting multiclass object classifier performs well on the benchmark databases tested. We also note that so long as the model uses sufficiently many detected points, the matching of elementary parts to image features does not need to be very accurate. We showed how a simple three-layer hierarchy of object, parts and features can give satisfying visual intuition and probabilistic accuracy.

Future work: The model applies to arbitrary spatial transformations between parts and their subparts, and arbitrary numbers of layers, although here we applied it only with translation-scale transformations and 3 layers. Future work will study the advantages of more general transformations and additional layers. The main difficulty is likely to be getting a good initialization for these more complex models. Another promising direction is to learn mixtures of such generative models, for image clustering or to handle more complex classes such as 3D models viewed from all possible directions.

References

[1] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV'04 workshop on Statistical Learning in Computer Vision*, pages 59–74, Prague, 2004.

[2] G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, 2003.

[3] Gy. Dorko and C. Schmid. Object class recognition using discriminative local features. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2004. submitted.

[4] Li Fei-Fei, Rob Fergus, and Pietro Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages 1134–1141, Nice, France, 2003.

[5] R. Fergus, P. Perona, and A.Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, 2003.

[6] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, 2003.

[7] William T. Freeman Kevin Murphy, Antonio Torralba. Using the forest to see the trees: A graphical model relating features, objects, and scenes. In *Neural Info. Processing Systems*, 2003.

[8] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with and implicit shape model. In *ECCV'04 workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, 2004.

[9] D. G. Lowe. Local feature view clustering for 3D object recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, pages 682–688, December 2001.

[10] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, June 2003.

[11] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In *4th International Workshop on Visual Form, Capri, Italy*, May 2001.

[12] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, pages 18–32, 2000.