International Conference on Pattern Recognition, Vol. 2, pp. 395-398, 2006.

Unsupervised Learning of Dense Hierarchical Appearance Representations

Fabien Scalzo and Justus H. Piater
Montefiore Institute, University of Liège, 4000 Liège, Belgium
{fscalzo,Justus.Piater}@ulg.ac.be

Abstract

We describe an unsupervised, probabilistic method for learning visual feature hierarchies. Starting from local, low-level features computed at random locations, the method combines features hierarchically. At each level of the hierarchy, pairs of features are identified that tend to occur at stable positions relative to each other, by clustering the configurational distributions of observed feature co-occurrences using Expectation-Maximization. Stable pairs of features thus identified are combined into higher-level features. This learning scheme results in a graphical model that constitutes a probabilistic representation of a flexible visual feature hierarchy. For detection, evidence is propagated using Nonparametric Belief Propagation, a recent generalization of particle filtering. In experiments, the proposed approach demonstrates effective learning and robust detection of objects in the presence of clutter and occlusion.

1. Introduction

Most natural and human-made objects have rigid local shape and appearance, but often present more variability at larger scales. Hierarchical approaches to object recognition have recently received increasing attention [1, 9, 2, 7]. They are well suited to represent shape variability at different scales and granularities within a single coherent framework. The current paper proposes an unsupervised generative method for learning hierarchical models that integrate spatial constraints between features [4]. We use a graphical model to represent the hierarchical feature structure (Section 2). In such a representation, each node corresponds to a visual feature class and is annotated with its possible locations in the image. The edges model both the spatial arrangement and the statistical dependence between two feature classes.

As was mentioned by G. Bouchard and B. Triggs [1], nearby object features have strongly correlated positions while more distant features are much more weakly correlated. The proposed learning framework (Section 3) focus on modelling correlations between features. It accumulates statistical evidence from feature observations in order to find feature classes that are often detected in the same neighborhood. Such features are said to be spatially correlated. The graph is incrementally built by combining correlated features into higher-level abstractions.

Detection (Section 4) is achieved by using Nonparametric Belief Propagation [8], a message-passing algorithm, to infer the belief associated with higher-level feature classes, that are not directly observable. Finally, the functioning and the efficacy of our method are demonstrated on both artificial and real image data sets (Section 5).

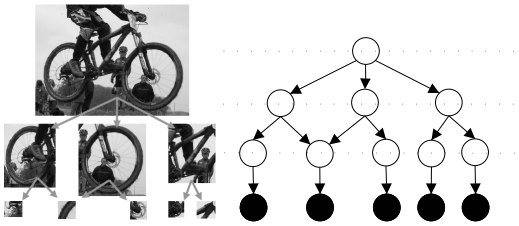

Figure 1. Object part decomposition (left) and corresponding graphical model (right).

2. A Hierarchical Representation of Features

In order to represent a hierarchy of visual features, a flexible statistical framework is required. A graphical model is perfectly suitable for this task (Figure 1). Nodes represent feature classes and are annotated with the detection information for a given scene. Edges represent two types of information; the distance between features and their hierarchical composition. The formulation in terms of graphical models is attractive because it provides a statistical model of the variability of shape and appearance. These are specified separately by the edges and the low-level nodes of the graph, respectively.

The vertices $s \in \mathcal{V}$ of the graph represent feature classes. They contain the feature activations for a given image. The graphical model associates each node with a hidden random variable $x \in \mathcal{R}^2$ representing the spatial probability distribution of the feature in the image.

The edges $e \in \overrightarrow{\mathcal{E}}$ of the graph model two aspects of

the feature structure. First, an edge signifies that the destination feature is a part of the source feature. Second, it describes the spatial relation between the compound feature and the subfeature in terms of distance. The annotation does not contain any specific information about a given scene but represents a static view of the feature hierarchy and shape. Each edge $e_{t\to s}$ from feature t to s is associated with parameters (d_{st}, σ_{st}) that respectively define the distance between features and the corresponding variance.

In our representation, we distinguish between two different types of features, primitive and compound features. Primitive features correspond to low-level features. They are represented by a local descriptor made of orientation-normalized pixel values. Orientation normalization is obtained by rotating the region around the point in the gradient direction. Descriptors are extracted at random locations, any other feature detector can be used. Compound features consist of flexible geometric combinations of other subfeatures (primitive or compound). Compound feature classes are represented by a non-leaf node whereas primitive feature classes are leaf nodes at the first level of the graph.

3. Learning a Hierarchy of Feature Classes

The basic concept behind the learning algorithm is to accumulate statistics of the relative positions of observed features in order to find *conspicuous coincidences* of feature co-occurrences. The structure of our graphical model is built incrementally by combining spatially correlated features into new feature abstractions.

Before the first iteration, the learning method applies a K-means clustering algorithm to the set of feature descriptors. This identifies primitive classes from the training set. These classes are used to create the first level (leaves) of the graphical model. After clustering, the procedure votes to accumulate information on the distance ($\Lambda_{i,j}$) between features and extracts the feature pairs ($\mathcal{S} \leftarrow [f_i, f_j]$) that tend to be located in the same neighborhood. Then it estimates the parameters (mean and variance of the distance) of the spatial relation using Expectation-Maximization (EM). It finally creates new feature nodes in the graph by combining the closest correlated features. This learning procedure is applied iteratively to each new level in the graph. An outline is given in Algorithm 1, and the main steps are described in the following paragraphs.

Algorithm 1 Learning: learn($\mathcal{G}$, level)

1: $\Lambda, \mathcal{S} \leftarrow$ extract($\mathcal{G}$,level) *// extract correlated features*
2: **for all** feature pairs $[f_i, f_j] \in \mathcal{S}$ **do**
3: $\quad\mathcal{T}_{ij} \leftarrow$ EM($\Lambda_{i,j}$) *// estimate models of spatial relation*
4: $\quad\mathcal{T} \leftarrow \mathcal{T} \cup$ closest($\mathcal{T}_{ij}$) *// keep the best relations*
5: **end for**
6: $\mathcal{G} \leftarrow$ generate($\mathcal{G}, \mathcal{T}$) *// generate new features*

3.1. Finding Spatially Correlated Features

A spatial correlation exists between two feature classes if they are often detected in the same neighborhood. The key idea to finding correlated features is to apply a voting scheme that will collect co-occurrence statistics (distances) from the training set. These statistics are collected from multiple feature occurrences within one or across many different images by considering each detected feature class and by voting for the class of the closest features detected in the same region. After applying this procedure over the training set, we obtain a set of distances Λ for every pair of classes. To extract spatially correlated classes, we keep the pairs that obtained the largest number of votes in a table $\mathcal{S}$.

3.2. Estimating Spatial Relations

In our framework, spatial relations are defined in terms of distances between features. In order to estimate this parameter from co-occurrence statistics, we use an Expectation-Maximization (EM) algorithm. In principle, a sample of observed distances $\Lambda_{i,j}$ between two given feature classes $[f_j, f_i]$ can be approximated by a Gaussian mixture, where each component represents a cluster of distances between these features. EM is used to estimate the parameters of the spatial relation between each correlated feature pair $[f_i, f_j] \in \mathcal{S}$. It maximizes the likelihood of the observed distances over the model parameters $\Theta = (w_{1\ldots K}; d_{1\ldots K}; \sigma_{1\ldots K})$. In practice, a mixture of only two Gaussian components ($K = 2$) is sufficient to model spatial relations. The first component represents the most probable distance, and the second is used to model the noise. When the distance d_1 and variance σ_1 of the first component are estimated between features i and j, we store them at entry $[i, j]$ in a table $\mathcal{T}$.

3.3. Generating New Features

When a reliable spatial relation has been estimated between a pair of feature classes $[f_i, f_j]$, the generation of a new feature is straightforward. We combine these features to create a new higher-level feature by adding a new node f_n in the graph. We connect it to its subfeatures $[f_i, f_j]$ by two edges $\{e_{n\to i}, e_{n\to j}\}$ that are added to the edge set $\mathcal{E}$.

The annotations of edges are computed using the mean and the variance of the distance $[d_{ij}, \sigma_{ji}] \in \mathcal{T}$ estimated during the preceding step. The relative position (*i.e.* distance) of the generated feature is chosen as the midpoint between the subfeatures. Thus the distance from subfeatures to the new feature is set to one half of the estimated distance between subfeatures such that $e_{n\to i} = \{d_{ij}/2, \sigma_{ij}\}$ and $e_{n\to j} = \{d_{ji}/2, \sigma_{ji}\}$. Finally, the new edges and the new feature are respectively added to the edge set, $\mathcal{E} \leftarrow \mathcal{E} \cup \{e_{n\to i}, e_{n\to j}\}$, and to the vertex set, $\mathcal{V} \leftarrow \mathcal{V} \cup f_n$, of the current graph $\mathcal{G}$.

4. Inferring High-Level Features

Inferring the probability density function of hierarchical features conditioned on the detected primitives amounts to estimating the belief of the corresponding higher-level nodes. Feature detection can thus be posed as inference in a graphical model. One way to do this is to use Nonparametric Belief Propagation (NBP) [8]. NBP is an inference algorithm for graphical models that generalizes particle filtering. This algorithm propagates information by a series of local message-passing operations. Following the NBP notation, a message m_{ij} from node i to j is written

$$m_{i,j}(x_j) = \int \psi_{i,j}(x_i, x_j)\ \phi_i(x_i, y_i) \prod_{k \in \tau_i \setminus j} m_{k,i}(x_i)\ dx_i \quad (1)$$

where x_i is the random variable associated with the node i, τ_i denotes its neighbors and $\phi_i(x_i, y_i)$ is the observation potential obtained via the detection of a low-level feature. The potential $\psi_{i,j}$ models the spatial relation between nodes i, j. Since the prior $p(x_i)$ is uniform, it is proportional to the conditional $\psi_{i,j}(x_j|x_i)$. This is defined by a function $F(x_i, d_{i,j}, \sigma_{i,j})$ that maps the distribution of a node i to a connected node j according to their spatial relation (Fig. 2).

Each random variable x is represented by a density estimate that is defined as a set of weighted samples $\{\mu^m, w^m\}_{m=1}^M$. The parameters are the means μ^m and the weights w^m of the particles. During an iteration of NBP, each vertex receives messages from connected nodes. Each message consists of a set of samples. By estimating the product of theses incoming messages $m_{k,i}(x_i)$, a node can compute the new local belief, an approximation of the marginal distribution $p(x_i|y)$:

$$\hat{p}(x_i|y) = \phi_i(x_i, y_i) \prod_{k \in \tau_i} m_{k,i}(x_i) \quad (2)$$

An exact computation of this product being intractable, we use the Gibbs sampler to draw M weighted samples $\{\hat{\mu}_i^m, \hat{w}_i^m\}_{m=1}^M$ from the product of incoming messages. The number M of samples used to represent each message is proportional to the image size.

A message sent from i to j specifies the location evidence of the destination node j according to the belief available in the source node i. Such a message is produced by computing the product of messages received by i (except the one received from j) with the observation potential,

$$\{\hat{\mu}_p^m, \hat{w}_p^m\}_{m=1}^M = \phi_i(x_i, y_i) \prod_{k \in \tau_i \setminus j} m_{k,i}(x_i) \quad (3)$$

Then we resample from the conditional $\psi_{i,j}(x_j|x_i)$. To do this, each sample $\hat{\mu}_p^m$ is moved by distance d_{ij} in a set of random directions $\theta_k \in [0, 2\pi[$ (eq. 4). The resulting particules constitute the new message $m_{i,j} = \{\mu_{ij}^m, w_p^m\}_{m=1}^M$.

$$\mu_{ij}^m = \hat{\mu}_p^m + d_{ij}[\sin\theta_k\ \cos\theta_k] \quad (4)$$

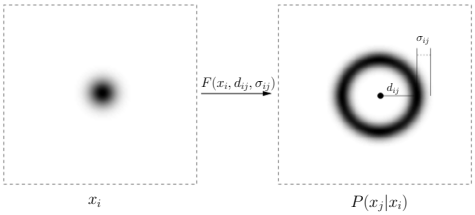

Figure 2. Function F that computes the conditional probability between two variables.

We approximate NBP by setting the variance of each sample to a fixed value.

In our representation, higher-level features are not directly observable. Evidence is incorporated into the graph exclusively via primitive classes at the leaves of the graph. The observation potentials $\phi_i(x_i, y_i)$ are defined by Gaussian mixtures. For initialization, we match detected features in an image with the most similar classes. Each component is set to a detected location and thus represents a possible location of the feature class. The weight of a component is inversely proportional to the Mahalanobis distance between the observed descriptor and the feature class. To deal with occlusions, we add a high-variance Gaussian located at the image center to the observation potential.

5. Experiments

We have investigated the behavior of hierarchies on two object recognition tasks. During the first experiment, we trained our models on the objects of the COIL-100 database [5]. As a training set, we utilized 24 of the 72 available views uniformly distributed around each object. Our primitives were random patches described by rotation-invariant descriptors of 13×13 pixel values. We deliberately used simple, relatively inexpressive features to demonstrate the functioning of our method. The number of feature classes was selected according to the BIC criterion. We limited the combination of each feature to the three best spatial relations and we ended the procedure after six levels of combination. A graph resulting from learning one object model is presented in Figure 3. Recognition was tested against the 48 remaining images of each object. We give in Figure 4 the confusion matrices concerning the first 25 object models for increasing number of levels (1 to 6) in the hierarchy. Brightness indicates the total number of detections. As we can observe, a single level of feature combinations is weakly discriminant. Thanks to our hierarchy, selectivity arises from spatial combination of high-level features.

The second object recognition experiment was conducted on a database of real objects [6]. This image library contains color images of eight different real objects. Each image was captured at different arbitrary poses around the object. For our experiments, we still used the same random primitives but kept the best five spatial relations of each feature to construct a hierarchy. The models were evaluated on 51 test images of the object recognition database. These

images represent cluttered scenes containing previously-learned objects along with distractors and various illumination conditions. The recognition results (Figure 5) are comparable to the state-of-the-art methods using 2D models [3] or explicit 3D matching [6]. This is remarkable since we use neither robust features nor any discriminative information to build our models. Ferrari's method considers all training images independently which essentially reduces recognition to the matching of robust features. By using statistical learning and a flexible representation, we obtain good results with simple features.

6. Conclusion

We presented a generic framework for unsupervised learning of visual feature hierarchies. Our experiments show that the system works well for both isolated objects [5] and real scenes [6] in the presence of clutter and occlusion. They demonstrated the representational power of the hierarchy; recognition rates increase with the depth of the hierarchy. In comparison with previous approaches, the proposed framework offers several improvements. First, it allows the learning and the represention of *shape flexibility* at different levels in the hierarchy by only considering the distance between features. Second, it does not rely on complex features such as SIFT features; simple *random primitives* can be used. Third, where previous methods were limited to a few high-level features (typically, less than 10), our system is able to deal with *dense graphical models* containing more than 500 nodes. Finally, the recognition of *3D real objects* is successful even in the presence of arbitrary 3D rotations, scale changes, lightning conditions and severe occlusions.

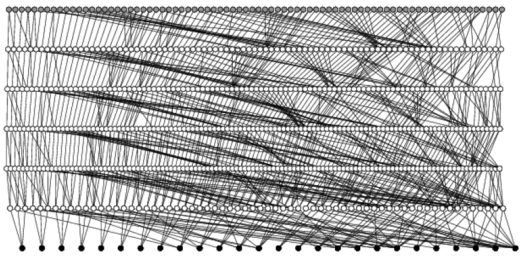

Figure 3. Graphical model learned from random patches for an object of COIL-100 [5].

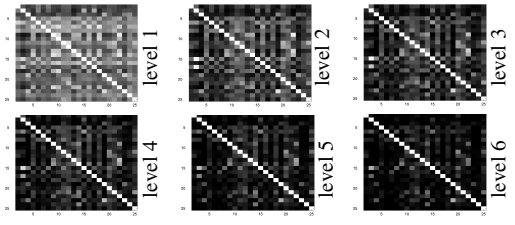

Figure 4. Confusion matrices for one- to six-level models (COIL-100 [5]).

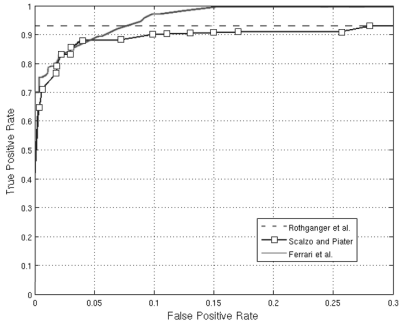

Figure 5. ROC curve obtained by our hierarchical framework and two of the best state-of-the-art methods.

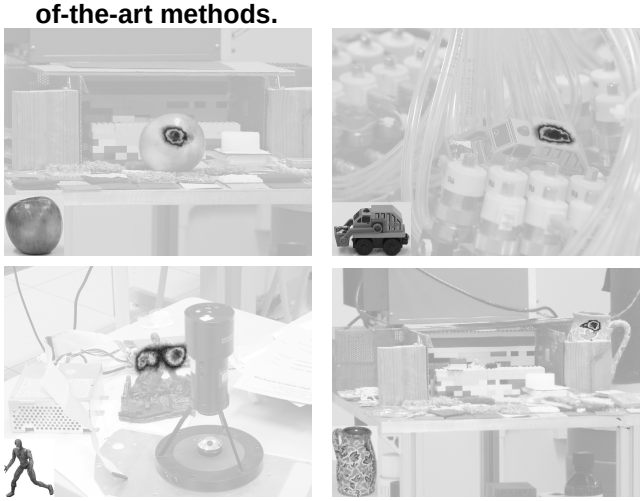

Figure 6. Object Detection, the illustrations correspond to the activation of the top level feature that had the highest response rate.

References

[1] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, pages 710–715, 2005.

[2] B. Epshtein and S. Ullman. Feature hierarchies for object classification. In *ICCV*, pages 220–227, 2005.

[3] V. Ferrari, T. Tuytelaars, and L. V. Gool. Simultaneous object recognition and segmentation by image exploration. In *ECCV*, pages 40–54, May 2004.

[4] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *BMVC*, pages 959–968, 2004.

[5] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical Report CUCS-005-96, 1996.

[6] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D Object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. In *IJCV*, volume 66, pages 231–259, March 2006.

[7] F. Scalzo and J. Piater. Statistical learning of visual feature hierarchies. In *IEEE Workshop on Learning in CVPR*, 2005.

[8] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *CVPR*, pages 605–612, 2003.

[9] H. Wersing and E. Koerner. Unsupervised learning of combination features for hierarchical recognition models. In *ICANN*, pages 1225–1230, 2002.

[0]This work was supported by the Wallonian Ministry of Research (D.G.T.R.E.) under Contract No. 03/1/5438.