# Sparse Flexible Models of Local Features

Gustavo Carneiro

Integrated Data Systems Department
Siemens Corporate Research
Princeton, NJ, USA.

David Lowe

Department of Computer Science
University of British Columbia
Vancouver, BC, Canada.

**Abstract.** In recent years there has been growing interest in recognition models using local image features for applications ranging from long range motion matching to object class recognition systems. Currently, many state-of-the-art approaches have models involving very restrictive priors in terms of the number of local features and their spatial relations. The adoption of such priors in those models are necessary for simplifying both the learning and inference tasks. Also, most of the state-of-the-art learning approaches are semi-supervised batch processes, which considerably reduce their suitability in dynamic environments, where unannotated new images are continuously presented to the learning system. In this work we propose: 1) a new model representation that has a less restrictive prior on the geometry and number of local features, where the geometry of each local feature is influenced by its $k$ closest neighbors and models may contain hundreds of features; and 2) a novel unsupervised on-line learning algorithm that is capable of estimating the model parameters efficiently and accurately. We implement a visual class recognition system using the new model and learning method proposed here, and demonstrate that our system produces competitive classification and localization results compared to state-of-the-art methods. Moreover, we show that the learning algorithm is able to model not only classes with consistent texture (e.g., faces), but also classes with shape only (e.g., leaves), classes with a common shape but with a great variability in terms of internal texture (e.g., cups), and classes of flexible objects (e.g., snake).[1]

## 1 Introduction

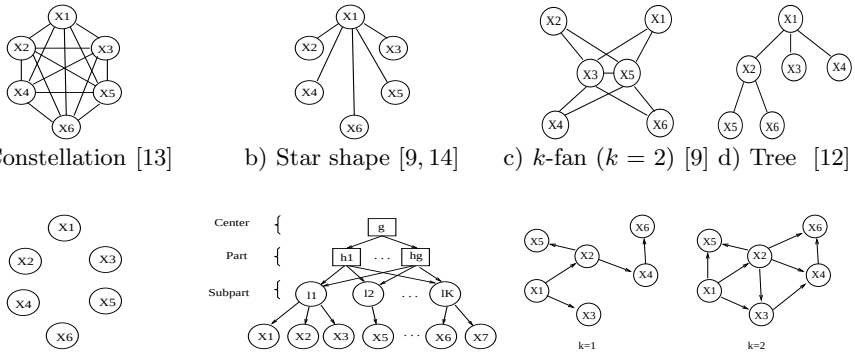
The visual recognition problem is currently one of the most difficult challenges for the computer vision community. Albeit studied for decades, we are still far from a solution that is truly generalizable to many types of visual classes. New attention has been devoted to this problem after the influential papers [2, 5], where their main contribution was a combination of principled probabilistic recognition models and (semi-)local image descriptors. The main goal is to represent a visual class with a generative model comprising both the appearance and spatial distributions of those descriptors. This problem has been aggressively tackled lately, where the objective is to provide efficient models (in terms of learning and inference) with good recognition performance [13, 14, 12, 4, 10, 17, 20, 21]. Note that learning is a method to estimate the model parameters, and inference is an approach to classify a test image as being generated by one of the learned models.

[1] This work was performed while Gustavo Carneiro was at the University of British Columbia.

In order to make the problem tractable, most of the current approaches make the following assumptions: 1) mutual independence of the appearance of parts given the model; 2) independence of appearance and geometry of parts given the model; 3) restrictive priors in terms of the geometry and number of parts. It is worth noting that we assume a model part to be represented by a local feature, and the geometry of a part to comprise position, scale, and dominant orientation. The third assumption above has two extremes. One extreme is that the geometry of parts is independent given the model [10, 21] (see the bag of features model in Fig. 1), which reduces the number of parameters to estimate during the learning stage. However, this approach leads to a poor model representation that fails to incorporate any information on the relative geometry of parts. The other extreme is to model the joint distribution of the geometry of parts [13] (see the constellation model in Fig. 1), which produces a rich representation. The main challenge with the latter model is that the number of parameters grows exponentially with the number of parts, and learning quickly becomes intractable even with a relatively small number of parts (e.g., less than 10 parts). It is unclear what types of visual classes can be effectively represented with such a small number of parts.

The middle ground between these two extremes has been intensively studied recently, where the goal is to assume restrictive priors in terms of the geometric configuration of parts in order to improve the efficiency of inference (i.e., fewer hypotheses from a test image to evaluate) and learning (i.e., fewer parameters to estimate). For example, the assumption of a star-shaped [9, 14] or a hierarchical prior configuration of local features [12, 4] (see Fig. 1) reduces the number of parameters to estimate, and inference takes advantage of the fact that all these models possess a "special" node (e.g., root in the tree, or center node in the star-shape model), which serves as a starting point for the formation of hypotheses, and consequently reduces the inference complexity. However, it is not clear what the limitations of those models are in terms of which visual classes can be represented using such restrictive priors in terms of the geometry of parts. Also, even though those methods are capable of dealing with more parts, there is still a limit of 20 to 30 parts, which clearly represents an issue if more complex classes are to be represented. A notable exception is the hierarchical model [4] that is able to deal with hundreds of parts, but it assumes an embedded hierarchical model with a small number of nodes, which might impose limits in the visual classes that can be represented with it. Finally, most of these models' parameters are learned using a (semi-)supervised off-line learning approach. This learning approach decreases the flexibility of those methods in dynamic environments where new unannotated training images are continuously presented to the learning system.

In this paper we propose: 1) a new model for the visual classification problem that contains a less restrictive prior on the geometry and number of local features, where the geometry of each model part depends on the geometry of its $k$ closest neighbors; and 2) an unsupervised on-line learning algorithm that is capable of identifying commonalities among input images, forming clusters of images with similar appearances, and also estimating the model parameters efficiently and accurately. As commonly assumed in the state-of-the-art works, we also assume that the appearance and the geometry of parts are independent given the model, and that the appearance of parts is mutually independent given model. The main novelty of our model is a prior based on a semi-full dependency of the geometry of parts given model (see Fig. 1-(g)). Note from the




a) Constellation [13] b) Star shape [9, 14] c) $k$-fan ($k = 2$) [9] d) Tree [12]

e) Bag of features [10, 21] f) Hierarchy [4] g) Sparse flexible model

**Fig. 1.** Graphical geometric models of priors. Note that Xi represents a model part.

graph representing our model that the geometry of each feature depends on the geometry of its $k$ neighboring features, where $k$ is a parameter that defines the degree of connectivity of each part. This prior enables an explicit control on the connectivity of the parts, and it also allows for the object being modeled to have (semi-)local rigid deformation within the area covered by the connected features, and rigid/non-rigid global deformation. Our objective with this new model is to extend the types of classes that can be represented with local image features since the model can potentially have hundreds of parts, tightly connected locally, but loosely connected globally.

We implement a new visual class recognition system using this new model and learning method described above, and demonstrate that our system produces competitive classification and localization results compared to state-of-the-art methods using standard databases. Moreover, we show that the learning algorithm is able to model not only classes with reasonable texture (e.g., faces), but also classes with shape only (e.g., leaves), classes with a common shape but with a great variability in terms of internal texture (e.g., cups), and classes of flexible objects (e.g., snakes).

## 2 Local Image Features

A local image feature represents a part in our model, and consists of an image representation of local spatial support comprising an image region at a selected scale. In this work we assume that a local feature has appearance and geometry. The appearance is the image feature extracted from the local region, while the geometry represents the image position from where it was extracted, the dominant orientation in that image position, and the filter scale used to extract the image feature. Therefore, a local feature vector $\mathbf{f}$ is described as $\mathbf{f} = [\mathbf{a}, \mathbf{g}]$, where $\mathbf{a}$ is the appearance, and $\mathbf{g} = [\mathbf{x}, \theta, \sigma]$ is the geometry consisting respectively of the position $\mathbf{x}$, orientation $\theta$, and scale $\sigma$.

### 2.1 Correspondence Set

A correspondence set represents a data association between two sets of local features. Let us say we have a set $\mathcal{F}_1 = \{\mathbf{f}_1, ..., \mathbf{f}_M\}$ and another set $\mathcal{F}_2 = \{\hat{\mathbf{f}}_1, ..., \hat{\mathbf{f}}_N\}$. An association is a mapping of the $M$ features from set $\mathcal{F}_1$ to the $N$ feature of set $\mathcal{F}_2$. In this work, a correspondence set is denoted as

$$\mathcal{E} = \{(\mathbf{f}_1, \hat{\mathbf{f}}_{c(1)}), ..., (\mathbf{f}_M, \hat{\mathbf{f}}_{c(M)})\} = \{e_1, ..., e_M\},$$

where $\mathbf{f}_i \in \mathcal{F}_1$, $\hat{\mathbf{f}}_{c(i)} \in \mathcal{F}_2$, and $c(.)$ is a mapping function that associates a feature from $\mathcal{F}_1$ to $\mathcal{F}_2$. When $\mathbf{f}_i \in \mathcal{F}_1$ is not paired with any feature from $\mathcal{F}_2$, then the correspondence is denoted as $(\mathbf{f}_i, \emptyset)$.

## 3 Probabilistic Model

Assume that there are $C$ visual classes in the database of models, where each class $\omega_i$ is represented by a set $\mathcal{F}_i$ of $M$ features, and also by appearance and geometry parameters. Also, consider the presence of a class $\omega_0$ that models general background images. A test image $I$ produces the set $\mathcal{F}_I$ of $N$ features. Then our goal is to first determine the likelihood of the presence of an instance of class $\omega_i$ in the test image, and then determine the location of each instance. Hereafter, we refer to the former problem as *classification*, and the latter as *localization*. In order to solve the data association problem, assume that $\mathcal{H}_{iI}$ is the set of all possible correspondence sets from the model features to the test image features. Thus, each correspondence set $\mathcal{E}_{iI} \in \mathcal{H}_{iI}$ has size $M$ (i.e., the number of model features).

The classification of model $\omega_i$ given the features $\mathcal{F}_I$ extracted from image $I$ involves the computation of the following ratio:

$$R = \frac{P(\omega_i|\mathcal{F}_I)}{P(\omega_0|\mathcal{F}_I)} = \frac{P(\mathcal{F}_I|\omega_i)P(\omega_i)}{P(\mathcal{F}_I|\omega_0)P(\omega_0)}. \tag{1}$$

The prior ratio $\frac{P(\omega_i)}{P(\omega_0)}$ is assumed to be one, and the likelihood term can be obtained by marginalizing out the variable $\mathcal{E}_{iI} \in \mathcal{H}_{iI}$ that denotes the correspondence set, as follows:

$$P(\mathcal{F}_I|\omega_i) = \sum_{\mathcal{E}_{iI} \in \mathcal{H}_{iI}} P(\mathcal{F}_I, \mathcal{E}_{iI}|\omega_i) = \sum_{\mathcal{E}_{iI} \in \mathcal{H}_{iI}} P(\mathcal{F}_I|\mathcal{E}_{iI}, \omega_i)P(\mathcal{E}_{iI}|\omega_i). \tag{2}$$

Hence, there can be $O(M^N)$ different correspondence sets between $\mathcal{F}_i$ and $\mathcal{F}_I$. However, recall that we aim at a rich visual class representation with hundreds of parts, and possibly thousands of features extracted from a test image, which makes (2) intractable. Therefore, we have to rely on a heuristic that quickly identifies a subset of $\tilde{\mathcal{H}}_{iI} \subset \mathcal{H}_{iI}$ which contains correspondence sets that have the potential to lead to a correct correspondence set. Finally, the likelihood ratio in (1) is then approximated with

$$\frac{P(\mathcal{F}_I|\omega_i)}{P(\mathcal{F}_I|\omega_0)} \approx \max_{\mathcal{E}_{iI} \in \tilde{\mathcal{H}}_{iI}} \frac{P(\mathcal{F}_I|\mathcal{E}_{iI}, \omega_i)P(\mathcal{E}_{iI}|\omega_i)}{P(\mathcal{F}_I|\mathcal{E}_{iI}, \omega_0)P(\mathcal{E}_{iI}|\omega_0)}. \tag{3}$$

First let us concentrate on the term $P(\mathcal{E}_{iI}|\omega)$ in the ratio (3) above. Given the high number of model features, we assume that the prior of having a specific match in the correspondence set is mutually independent of other matches. Therefore, we have

$$P(\mathcal{E}_{iI}|\omega) = \prod_{j=1}^{M} P(e_j|\omega). \tag{4}$$

Basically, $P(e_j|\omega)$ describes the likelihood of detecting model feature $\mathbf{f}_j$ in a test image assuming the presence of model $\omega$.

The term $P(\mathcal{F}_I|\mathcal{E}_{iI}, \omega)$ is computed as follows:

$$P(\mathcal{F}_I|\mathcal{E}_{iI}, \omega) = \left[\prod_{j=1}^{M} P(\hat{\mathbf{a}}_{c(j)}|e_j, \omega)\right] P(\{\hat{\mathbf{g}}_{c(j)}\}_{j=1..M}|\mathcal{E}_{iI}, \omega), \tag{5}$$

where $P(\{\hat{\mathbf{g}}_{c(j)}\}_{j=1..M}|\mathcal{E}_{iI}, \omega) = P(\hat{\mathbf{g}}_{c(M)}|\{\hat{\mathbf{g}}_{c(j)}\}_{j=1..(M-1)}, \mathcal{E}_{iI}, \omega)...P(\hat{\mathbf{g}}_{c(1)}|\mathcal{E}_{iI}, \omega)$, which is the decomposition of the likelihood of feature geometry using the chain rule of probability. The first term $P(\hat{\mathbf{a}}_{c(j)}|e_j, \omega)$ represents the likelihood of having the appearance matching between model feature $\mathbf{f}_j$ and test image feature $\hat{\mathbf{f}}_{c(j)}$. The second term $P(\{\hat{\mathbf{g}}_{c(j)}\}_{j=1..M}|\mathcal{E}_{iI}, \omega)$ denotes the likelihood of having a specific joint geometry of model features that were paired to features in the test image. It is important to mention that the decomposition can happen in all possible ways, which means that feature $\mathbf{f}_1$ does not represent a "special" feature that needs to be found in the test image in order to find all the other model features. As a result, another possible decomposition would be $P(\hat{\mathbf{g}}_{c(1)}|\{\hat{\mathbf{g}}_{c(j)}\}_{j=2..M}, \mathcal{E}_{iI}, \omega)...P(\hat{\mathbf{g}}_{c(M)}|\omega)$. Notice that even though we decompose this joint distribution, its computation still has a high time complexity. Moreover, this joint distribution would make the model sensitive to non-rigid deformations. Therefore, in order to solve these two issues, we approximate $P(\hat{\mathbf{g}}_{c(M)}|\{\hat{\mathbf{g}}_{c(j)}\}_{j=1..(M-1)}, \mathcal{E}_{iI}, \omega)$ to:

$$P(\hat{\mathbf{g}}_{c(M)}|\{\hat{\mathbf{g}}_{c(j)}\}_{j=\arg(\mathcal{K}_{iI}(\mathbf{f}_M,k,\mathcal{E}_{iI}))}, \mathcal{K}_{iI}(\mathbf{f}_M, k, \mathcal{E}_{iI}), \omega), \tag{6}$$

where $\mathcal{K}_{iI}(\mathbf{f}_M, k, \mathcal{E}_{iI}) \subset \mathcal{E}_{iI}$ returns the correspondences containing the $k$ closest model features to feature $\mathbf{f}_M$ in the geometric space of the model. The parameter $k$ denotes how sparsely each model feature is connected to its neighbors and is used to adjust the tradeoff between the richness of representation and the sensitivity of the model to non-rigid deformations. Also the richer the representation is (i.e., larger $k$), the higher the complexity of computing (6).

### 3.1 Probabilistic Correspondence Based on Semi-local Geometric Coherence

Equation 6 introduces the likelihood of the geometry of the observed test image feature $\hat{\mathbf{g}}_{c(l)}$ given the geometric information present in the respective $k$ closest model features to $\mathbf{g}_l$ in the space of model geometry. Following up on the idea described in [7], the geometric values of the test image feature $\hat{\mathbf{f}}_{c(l)}$ are predicted using the following pairwise relations:

$$\mathbf{n}^T_{c(l)c(o)}(\mathbf{x}_{c(l)} - \mathbf{x}_{c(o)}) = \|\mathbf{x}_l - \mathbf{x}_o\| + r_{\mathcal{D}}(\mathbf{f}_l, \mathbf{f}_o),$$

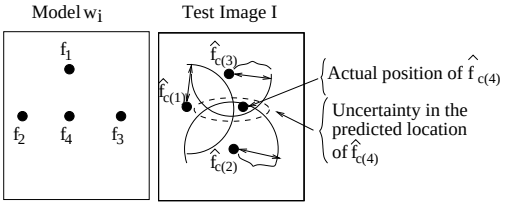



**Fig. 2.** Example of position prediction. Given the set of model features $\{\mathbf{f}_l\}_{l\in\{1,2,3,4\}}$, suppose we want to estimate the position of test image feature $\hat{\mathbf{f}}_{c(4)}$. The probable location of the feature (represented by an ellipsoid) is based on a Gaussian distribution computed using the position of the correspondences in the test and model images and the pairwise variances $\sigma^2_{\mathcal{D}}(\mathbf{f}_l, \mathbf{f}_o)$ estimated in the learning stage.

$$\begin{aligned}(\theta_{c(l)} - \theta_{c(o)})_{2\pi} &= (\theta_l - \theta_o)_{2\pi} + r_{\mathcal{O}}(\mathbf{f}_l, \mathbf{f}_o),\\ \tfrac{\sigma_{c(l)} - \sigma_{c(o)}}{\sigma_{c(o)}} &= \tfrac{\sigma_l - \sigma_o}{\sigma_o} + r_{\mathcal{S}}(\mathbf{f}_l, \mathbf{f}_o),\end{aligned} \tag{7}$$

where $\mathbf{n}_{c(l)c(o)} = \frac{\mathbf{x}_{c(l)} - \mathbf{x}_{c(o)}}{\|\mathbf{x}_{c(l)} - \mathbf{x}_{c(o)}\|}$, $(.)_{2\pi} \in [0, 2\pi)$, and $r_i(\mathbf{f}_l, \mathbf{f}_o)$ is a Gaussian noise with zero mean and variance $\sigma^2_i(\mathbf{f}_l, \mathbf{f}_o)$ for $i = \mathcal{D}, \mathcal{O}, \mathcal{S}$. The predicted geometry for $\hat{\mathbf{f}}_{c(l)}$, namely $[\hat{\mathbf{x}}^*_{c(l)}, \hat{\theta}^*_{c(l)}, \hat{\sigma}^*_{c(l)}]$ (see Fig. 2), is computed by combining the prediction produced by each one of the $k$ model features assuming that: 1) the variances $\sigma^2_i(\mathbf{f}_l, \mathbf{f}_o)$ are pairwise independent, and 2) the prediction produced by each correspondence is weighted by 1) the distance between these two features in the model space.

Therefore, the likelihood in Eq. 6 can be written as:

$$g([\mathbf{x}_{c(M)}, \theta_{c(M)}, \sigma_{c(M)}]^T - [\mathbf{x}^*_{c(M)}, \theta^*_{c(M)}, \sigma^*_{c(M)}]^T; \Sigma_t), \tag{8}$$

where $g(.)$ is the Gaussian function with zero mean, and $\Sigma_t$ is the weighted covariance computed with the $k$ pairwise variances.

There are two important issues to mention in the computation above. The first issue is the computation of the likelihood of the first match in the correspondence set, which is calculated as $P(\mathbf{g}_1|\mathcal{K}_{iI}(\mathbf{f}_1, k, \mathcal{E}_{iI}), \omega) = \frac{1}{2\pi}\frac{1}{A}\frac{1}{(\sigma_{\mathrm{MAX}} - \sigma_{\mathrm{MIN}})}$, where $2\pi$ represents the range of orientation, $A$ is the area of the image in the original image resolution, and $(\sigma_{\mathrm{MAX}} - \sigma_{\mathrm{MIN}})$ denotes the range of scales that the image has been processed. The second issue is the computation of the geometry likelihood assuming the model $\omega_0$. Here we assume that, conditioned on the model $\omega_0$, the likelihood of finding a feature with some specific geometry is independent and uniformly distributed, as follows $P(\{\mathbf{g}_j\}_{j=1..M}|\mathcal{E}_{iI}, \omega_0) = M\frac{1}{2\pi}\frac{1}{A}\frac{1}{(\sigma_{\mathrm{MAX}} - \sigma_{\mathrm{MIN}})}$.

### 3.2 Probabilistic Correspondences Based on Feature Appearance

The probability of the appearance match between model feature $\mathbf{f}_j$ and test feature $\hat{\mathbf{f}}_{c(j)}$ is denoted in (5) by $P(\hat{\mathbf{a}}_{c(j)}|e_j, \omega)$. According to [8], the distribution

of feature similarities between $\mathbf{f}_j$ and $\hat{\mathbf{f}}_{c(j)}$ can be adequately approximated with a *beta distribution* for the cases where this correspondence represents either a correct or a false matching. The beta distribution, denoted as $P_\beta(x; a, b)$, is defined in terms of two parameters $a$ and $b$. The parameters $a_{\text{on}}$ and $b_{\text{on}}$ will be learned for each feature $\mathbf{f}_j$ belonging to the model $\omega_i$ to explain the observed distribution of feature similarity values given a correct correspondence, and the parameters $a_{\text{off}}$ and $b_{\text{off}}$ will be learned for the distribution of similarities given a false correspondence. Hence, given the features $\mathbf{f}_j$ and $\hat{\mathbf{f}}_{c(j)}$, and their similarity denoted by $s(\mathbf{f}_j, \hat{\mathbf{f}}_{c(j)}) \in [0, 1)$, the likelihood of having correct and false appearance correspondences are respectively computed with:

$$\begin{aligned} P(\hat{\mathbf{a}}_{c(j)}|e_j, \omega_i) &= P_\beta(s(\mathbf{f}_j, \hat{\mathbf{f}}_{c(j)}); a_{\text{on}}(\mathbf{f}_j), b_{\text{on}}(\mathbf{f}_j)), \\ P(\hat{\mathbf{a}}_{c(j)}|e_j, \omega_0) &= P_\beta(s(\mathbf{f}_j, \hat{\mathbf{f}}_{c(j)}); a_{\text{off}}(\mathbf{f}_j), b_{\text{off}}(\mathbf{f}_j)). \end{aligned} \tag{9}$$

Finally, recall from Sec. 2.1 that a model feature can remain unmatched. In this case, the term $P(e_j|\omega)$ in (4), which denotes the probability of detecting model feature $\mathbf{f}_j$, works as a penalizing factor. That is, when $e_j = (\mathbf{f}_j, \emptyset)$, then $P((\mathbf{f}_j, \emptyset)|\omega)$ equals one minus the probability of detecting $\mathbf{f}_j$ [8].

## 4 Matching

The basic matching process consists of finding an initial correspondence set, and iteratively searching for additional correspondences assuming that the previous matches are correct. This process iterates as long as there are still model features available to match test image features. This matching process is not restricted to work with a single type of local feature. As exemplified in [14], this helps in the representation of different types of visual classes. Here, our model uses the following two different types of local image features: SIFT [18], and the multi-scale phase feature [6].

Assuming that the parameters of the distributions above have been learned (see Sec. 5), the matching process selects correspondence sets that produce a ratio $R > \tau_R$, where $\tau_R$ is an arbitrary constant (note that we can have more than one correct correspondence set, which means that several classes can be detected in the same test image and also multiple instances of the same class can also be detected in one test image). As explained in Sec. 3, the exhaustive search of correspondence sets is intractable, so we rely on certain heuristics for the matching process. We start the matching process with a nearest neighbor search, which builds the following correspondence set: $\mathcal{E}_{iI} = \{(\mathbf{f}_j, \hat{\mathbf{f}}_{c(j)}) | \mathbf{f}_j \in \mathcal{F}_i, \hat{\mathbf{f}}_{c(j)} \in \mathcal{F}_I, s(\mathbf{f}_j, \hat{\mathbf{f}}_{c(j)}) > \tau_s, \neg\exists \mathbf{f}_k \in \mathcal{F}_i \text{ s.t. } s(\mathbf{f}_k, \hat{\mathbf{f}}_{c(j)}) > s(\mathbf{f}_j, \hat{\mathbf{f}}_{c(j)})\}$, where $s(.) \in [0, 1)$ represents the similarity between two features, and $\tau_s$ is an arbitrary threshold (here $\tau_s = 0.6$ for the phase feature and $\tau_s = 0.55$ for SIFT, where the similarity measure for SIFT is normalized to be between 0 and 1). The next step comprises a feature clustering step, which assumes that the model suffered a specific type of spatial distortion and groups correspondences that move coherently according to that distortion type. This clustering process can assume rigid distortions (e.g., [18]) or non-rigid ones (e.g., [7, 16]). Similarly to [15, 19, 10], our method does not rely heavily on this initial set of matches produced by the grouping algorithm. In fact, these initial groups are useful as initial guesses for the matching algorithm. Moreover, it does not matter whether

this initial grouping is robust to non-rigid deformations since the model, in the process of expanding its correspondence set, is robust to non-rigid deformation because it depends more on nearby features than on far away features for the semi-local coherence presented in Sec. 3.1. Therefore, we adopt a simple Hough clustering approach with a restrictive rigid model (i.e., the bins in the Hough transform space are relatively small) that makes it extremely robust to outliers in the group, but sensitive to non-rigid deformations (see [7]). Specifically, for Hough clustering we used the following bin sizes: $5^o$ for rotation, factor of 2 for scale, and 0.05 times the maximum model diameter for translation. This restrictiveness results in a high number of groups, with each one having just a few correspondences.

### 4.1 Expanding the Correspondence Set

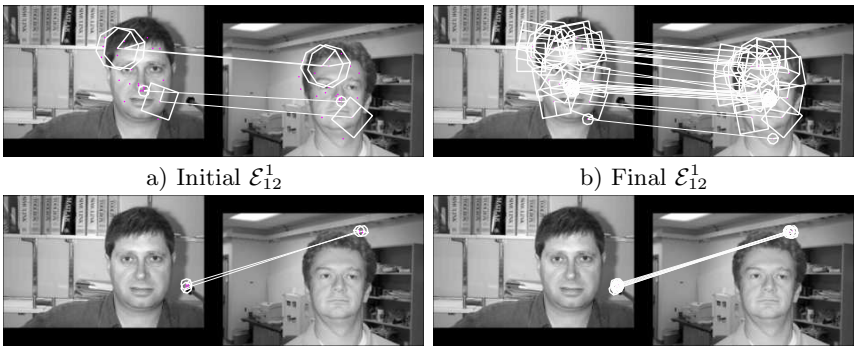
Given the groups built by the nearest neighbor search and clustering scheme, the expansion of each group is based on the following algorithm:

**Algorithm 1** *(***Matching***) Assuming that $G$ groups have been formed by the clustering process, where each group is denoted as $\mathcal{E}_{iI}^g$, the process of expanding this initial correspondence set is based on the following steps:*

1. *For each set $g \in \{1, ..., G\}$, do*
   (a) *Select the closest model feature $\mathbf{f}_j$ to any of the model features in $\mathcal{E}_{iI}^g$,*
   $$j = \arg \min_{(\mathbf{f}_j \in \mathcal{F}_i), (e_j \notin \mathcal{E}_{iI}^g)} \{\|\mathbf{x}_j - \mathbf{x}_l\|\}_{e_l \in \mathcal{E}_{iI}^g}$$
   (b) *Select the the next correspondence to include in $\mathcal{E}_{iI}^g$ according to $c(j) = \arg\max_{\hat{\mathbf{f}}_{c(j)} \in \mathcal{F}_I} P(\hat{\mathbf{f}}_{c(j)} | \mathcal{E}_{iI}^g, \omega_i)$ (see Eq. 5). Note that this computation does not have to be run over all test image features, since only a very small percentage of test image features lie sufficiently close to the predicted position, orientation, and scale of model feature $\mathbf{f}_j$;*
   (c) *If $P(\hat{\mathbf{a}}_{c(j)} | e_j, \omega_i) P(\hat{\mathbf{g}}_{c(j)} | \{\mathbf{g}\}_{j=1..,(j-1)}, \mathcal{E}_{iI}^g, \omega_i) P(e_j | \omega_i) > \tau_P$ (here, $\tau_P$ is dynamically determined based on the appearance parameters of the feature in 9 and the pairwise variances in (7), then include the correspondence $(\hat{\mathbf{f}}_{c(j)}, \mathbf{f}_j)$ in $\mathcal{E}_{iI}^g$, else include $(\emptyset, \mathbf{f}_j)$ in $\mathcal{E}_{iI}^g$;*
   (d) *Return to step 1 above until all model features are included in $\mathcal{E}_{iI}^g$.*

An example of the matching between two images containing faces (of different people) is shown in Fig. 3. Note that the matching algorithm tends to expand significantly the initial set $g \in \{1, ..., G\}$ when it contains correct correspondences.

Step 1(a) has complexity $O(M)$ if performed with linear search, where $M$ is the number of model features. However, approximate nearest-neighbor search algorithms [3] can find the nearest neighbor with high probability (which is sufficient for our purposes) in $O(\log(M))$ time. Both the number of groups to try, $G$, and the number of test features to consider in step 1(b), $K$, are bounded by constants. Therefore, the complexity of the Alg. 1 is $O(M \log(M))$. Recall that the models leading to the most efficient matching procedures in the literature are the k-fans [9] and the star shape [14]. The former method has complexity $O(MH^K)$, where H is the total number of places in the image, where $H >> M$, and $K >= 1$. The latter method has complexity $O(NM)$, where N is the number of parts detected in an image, so $N > M$. Hence, both methods would be intractable for large values of $M$ such as those used in our experiments.



a) Initial $\mathcal{E}_{12}^1$ b) Final $\mathcal{E}_{12}^1$

c) Initial $\mathcal{E}_{12}^2$ d) Final $\mathcal{E}_{12}^2$

**Fig. 3.** Matching a pair of images using Algorithm 1. The first column shows the initial group from the heuristic based on nearest neighbor and Hough clustering. The next column illustrates the final group after the process of expanding this initial group. The group in the first row is a correct match that can be considerably expanded, while the second row shows a false initial match. The octagonal shaped features represent the multi-scale phase feature [6], and the square shaped features represent SIFT [18]. The white line connecting features from the left to the right image shows the correspondence.

## 5 Learning

In this section we describe the process of learning the following model parameters:

- For each model feature $\mathbf{f}_j \in \mathcal{F}_i$ it is necessary to learn
  - the parameters of the feature conditional similarity distribution given $\omega_i$ (i.e., $a_{\text{on}}(\mathbf{f}_j)$ and $b_{\text{on}}(\mathbf{f}_j)$) and $\omega_0$ (i.e., $a_{\text{off}}(\mathbf{f}_j)$ and $b_{\text{off}}(\mathbf{f}_j)$),
  - the probability of feature detection given $\omega_i$ and $\omega_0$: $P(e_j|\omega_i)$, and $P(e_j|\omega_0)$, respectively.
- For each pair of model features $\mathbf{f}_l$ and $\mathbf{f}_o$, it is necessary to learn
  - the variance of the Gaussian noise affecting the distance, main orientation, and scale between $\mathbf{f}_l$ and $\mathbf{f}_o$ (see Eq. 7): $\sigma_{\mathcal{D}}^2(\mathbf{f}_l, \mathbf{f}_o)$, $\sigma_{\mathcal{O}}^2(\mathbf{f}_l, \mathbf{f}_o)$, and $\sigma_{\mathcal{S}}^2(\mathbf{f}_l, \mathbf{f}_o)$, respectively.

In the literature, the process of learning model parameters similar to the above consists of, first, clustering features in the feature space (either manually [12], or automatically [13]), and then, estimating the local feature and spatial parameters based on maximum likelihood estimation. The main issue involved in those learning methods is that the parameter estimation relies on gradient descent algorithms that are fragile in the presence of a high number of parameters since it can easily get stuck in local minima, which imposes very restrictive limits in the number of parts present in a model. Also, the time and size of training data required for this estimation grows quickly (e.g., exponential in [13]) in terms of the number of parameters. Therefore, weakly connected models (e.g., the star-shaped, or the hierarchical model) have been proposed in order to allow for faster and more reliable learning methods with fewer degrees of freedom. Nevertheless, if the number of parts exceeds say 20 parts, learning is usually intractable.

In this work, we propose the following unsupervised learning algorithm, where the main idea is to build correspondence sets between pairs of images and to cluster images that have strong correspondences.

**Algorithm 2** *(***Learning***) Consider a database of models $\Omega$ that is initially empty, and for each new training image $I$ that is presented to the system, we have the following steps:*

1. *For each $\omega_i \in \Omega$,*
   (a) *Run the matching Algorithm 1 to find an instance of $\omega_i$ in $I$, and select the correspondence set that maximizes the following ratio:*

$$\mathcal{E}_{iI}^{*} = \arg \max_{\mathcal{E}_{iI}^{g} \in \tilde{\mathcal{H}}_{iI}} \frac{P(\mathcal{F}_I|\mathcal{E}_{iI}^{g}, \omega_i) P(\mathcal{E}_{iI}^{g}|\omega_i)}{P(\mathcal{F}_I|\mathcal{E}_{iI}^{g}, \omega_0) P(\mathcal{E}_{iI}^{g}|\omega_0)}$$

   (b) *If the number of matched features in $\mathcal{E}_{iI}^{*}$ exceeds $\tau_{\mathcal{E}}$ (i.e., correspondences $(\mathbf{f}_j, \hat{\mathbf{f}}_{c(j)}) \in \mathcal{E}_{iI}^{*}$ , such that $\hat{\mathbf{f}}_{c(j)} \neq \emptyset$; here $\tau_{\mathcal{E}} = 30$) then update model $\omega_i$ using the correspondence set $\mathcal{E}_{iI}^{*}$ as the initial guess for matching the image $I$ to each image included in model $\omega_i$ using the matching Algorithm 1.*
2. *If the image $I$ failed to match any model $\omega_i \in \Omega$, then form a new model containing all image features and default values for the model parameters.*
3. *For every model $\omega_i \in \Omega$, build a graph, where each node represents an image present in $\omega_i$, and the edges between nodes have weights proportional to the number of non-empty correspondences found between these two images, and then run a connected component analysis so that the initial model can be split into tightly connected groups of images.*
4. *Search for common images present in two distinct models, say $\omega_i$ and $\omega_j \in \Omega$. If a common image is found between a pair of models, then check for common features in this image that is present both models, and based on that, join the two models into one single model.*

The output of this learning algorithm is a database of models, where each model consists of the images clustered together, the correspondence sets formed between pairs of model images, the features found in those sets, and the appearance and geometric parameters. In order to learn the parameters of the feature conditional similarity distribution given $\omega_i$ (i.e., $a_{\text{on}}(\mathbf{f}_j)$ and $b_{\text{on}}(\mathbf{f}_j)$), we build the histogram of feature similarities of each model feature and, assuming a beta distribution (Sec. 3.2), estimate its parameters [8]. The distribution given $\omega_0$ (i.e., $a_{\text{off}}(\mathbf{f}_j)$ and $b_{\text{off}}(\mathbf{f}_j)$) is then estimated computing the similarities between the model feature and the closest 20 background features (in the feature space)[8]. Note that the background features are extracted from 100 random images (see [8] for more details). The probability of feature detection given $\omega_i$ is computed with the detection rate of each model feature in $\omega_i$, and the detection given $\omega_0$ is the probability of detecting a feature in any image (this is done by computing the detection rate of any feature in the database of random images). The variance of the Gaussian noise affecting the distance, main orientation, and scale between pairs of model feature is computed using the correspondence sets in the model $\omega_i$. Finally, it is important to mention that the user has to specify the *upper bound* of the total number of features to be included in the model. Defining this upper bound on the number of model features is important in order to limit the

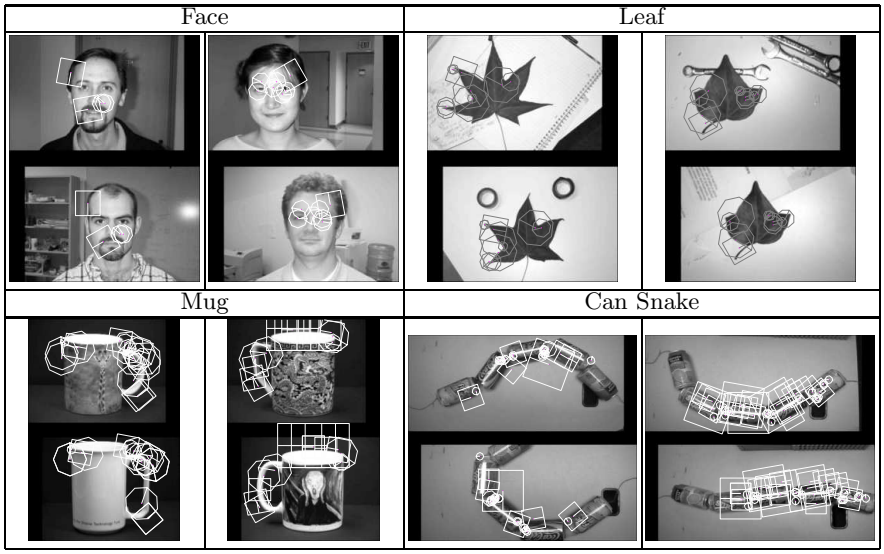



**Fig. 4.** Illustration of the correspondence sets between two pairs of images for each model. Note that each correspondence set between two images of the same model is shown in a single cell, where the arrangement of the features in the top image must find a similar structure in the bottom image.

computational complexity of the matching as defined in Sec. 4.1. Note that the model can have any number of features as long as this number is smaller than this user defined upper-bound. Whenever the learner has to eliminate features, it resorts to the classification based on the appearance statistics of the feature [8].

Our learning algorithm is used to build the models of the following databases: a) faces [13] (526 images), b) leaves [1] (186 images), c) mugs (74 images), and d) snake of cans [7] (40 images). For each database, we randomly selected half of the images for training, and the remaining images are used for testing. Fig. 4 shows two examples of matchings between pair of images present in each model.

## 6 Experimental Results

In this section we show the performance or our recognition system for the classification and localization problems.

### 6.1 Classification

Following [11], for each of the four object classes we use our recognition system to predict the presence/absence of at least one object of that class in a test image. The output of the classifier is the ratio (1) that represents the confidence of the object's presence so that a receiver operating curve (ROC) curve can be drawn. Note that we use the database of background images from [1] to draw the ROC curve.

In our first experiment, we show the ROC curves for each of the models in the database, and some examples of matchings (see Fig. 5). The database of faces is used in order to compare with the state-of-the-art methods in the literature. In this database, under similar experimental conditions, we get an equal error rate (EER) of 98.2% (recall that EER is the point at which the true positive rate equals one minus the false positive rate). The Face model in this experiment contains 3000 features and connectivity $k = 20$. This represents a competitive result compared to the EER=96.4% in [13] and of 98.2% in [9]. The EER is a function of the following two things (see Fig. 6): a) number of features present in the model, and b) connectivity $k$. The number of features in the model can be reduced by selecting a subset of the model features that are robust and detectable under model deformations, and distinctive (for details see [8]). Usually, the EER improves with the number of model features until it reaches a point of saturation, where more features do not improve the performance, but worsen the efficiency of the system. Moreover, higher $k$ also improves the richness of the representation (i.e., better EER), but reduces the system's efficiency. Finally, EER was 92.1% for the Leaf database, and 100% for the Mug and Can Snake databases.

### 6.2 Localization

We also use the experimental conditions described in [11] to illustrate the localization results. For each class, the task of our classifier is to predict the bounding box of each object in a test image. Each bounding box produced by our system is associated with a detection ratio (1) so that a precision/recall curve can be drawn. To be considered a correct localization, the area of overlap between the predicted bounding box $B_p$ and ground truth bounding box $B_{gt}$ must exceed $P$% by the formula: $\frac{area(B_b \cap B_{gt})}{area(B_p \cup B_{gt})}$. We show the precision recall curves for each of the four classes in Fig. 7 for $P = 50\%$ and $P = 25\%$. The main conclusion from these graphs is that our system is able to correctly localize the object in the image, but the bounding box formed by position of the local features present in the correspondence set tends to occupy a relatively small portion of the ground truth.

## 7 Conclusions

We have shown that it is possible to efficiently derive object class models containing hundreds of features by allowing each feature to depend on only its $k$ closest neighbors. This has the additional advantage that it can represent flexible objects in a natural way because their local geometry is often more tightly constrained than their global geometry. Our novel on-line learning algorithm is able to cluster images with similar appearance, identify consistent subsets of features, and efficiently estimate their model parameters. Experimental results show that this approach can be applied across a variety of object classes, even if they are defined by only a small subset of shared features.

**Acknowledgements** The authors would like to thank Kevin Murphy for useful discussions during the progress of this work and to thank Allan Jepson and Sven Dickinson for sharing the Mug database. The authors also wish to acknowledge funding received from NSERC (Canada) to support this research.

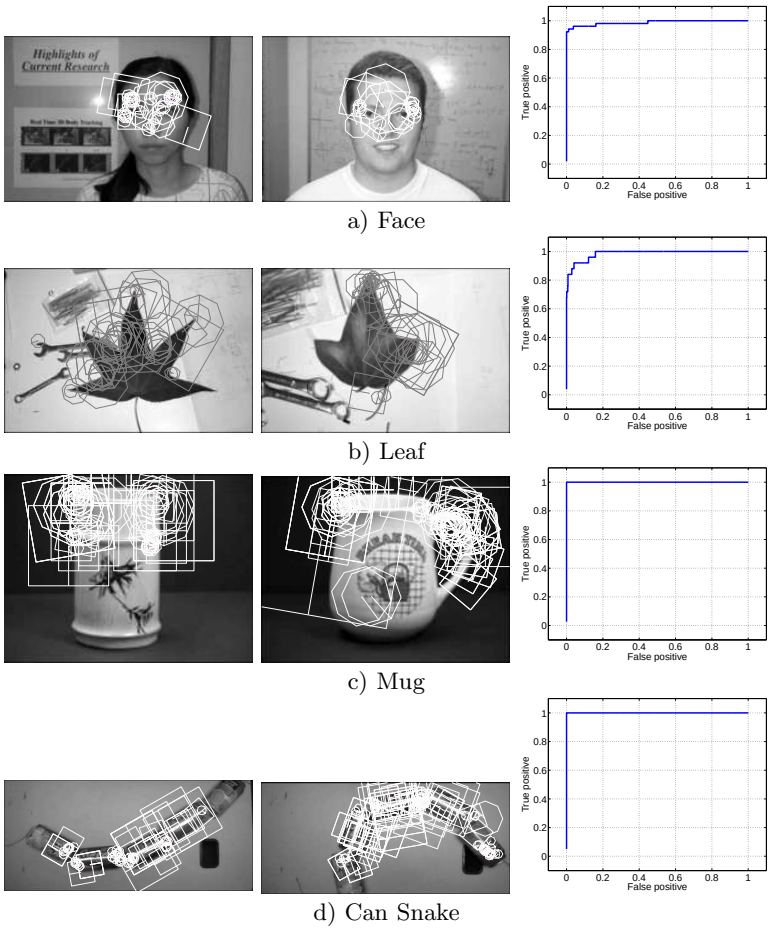



**Fig. 5.** Two examples of correspondence sets found in test images and the ROC curve for each model.

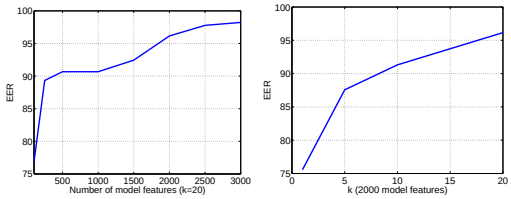
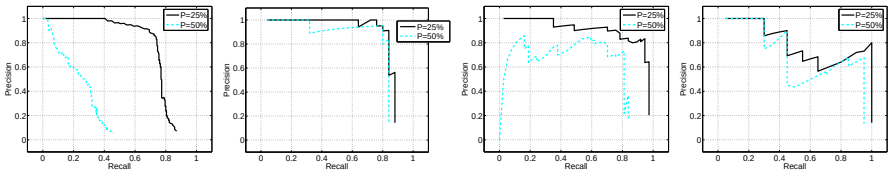



**Fig. 6.** EER versus number of training features and $k$ for the Face database.

## References

1. http://www.vision.caltech.edu/html-files/archive.html.




a) Face b) Leaf c) Mug d) Can snake

**Fig. 7.** The localization performance is assessed using the precision vs. recall curves for the Face, Leaf, Mug, and Can snake databases.


2. Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11:1691–1715, 1999.
3. S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45:891–923, 1998.
4. G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *CVPR*, 2005.
5. M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV*, 1998.
6. G. Carneiro and A. Jepson. Multi-scale local phase features. In *CVPR*, 2003.
7. G. Carneiro and A. Jepson. Flexible spatial models for grouping local image features. In *CVPR*, 2004.
8. G. Carneiro and A.Jepson. The distinctiveness, detectability, and robustness of local image features. In *CVPR*, 2005.
9. D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, 2005.
10. G. Csurka, C. Bray, and C. Dance L. Fan. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.
11. M. Everingham, L. Van Gool, C. Williams, and A. Zisserman. Pascal Visual Object Classes Challenge Results. 2005. (http://www.pascal-network.org/challenges/VOC/voc/results_050405.pdf).
12. P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.
13. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.
14. R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*, 2005.
15. V. Ferrari, T. Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation by image exploration. In *ECCV*, 2004.
16. S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *BMVC*, 2004.
17. B. Liebe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, 2003.
18. D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.
19. P.Moreels, M.Maire, and P. Perona. Recognition by probabilistic hypothesis construction. In *ECCV*, 2004.
20. D. Ramanan, D. Forsyth, and K. Barnard. Detecting, localizing, and recovering kinematics of textured animals. In *CVPR*, 2004.
21. N. Vasconcelos. *Bayesian models for visual information retrieval*. PhD thesis, Massachusetts Institute of Technology, 2000.