Object Recognition by Combining Appearance and Geometry

David Crandall[1], Pedro Felzenszwalb[2], and Daniel Huttenlocher[1]

[1] Cornell University
Ithaca, NY 14850
{crandall, dph}@cs.cornell.edu
[2] The University of Chicago
Chicago, IL 60637
pff@cs.uchicago.edu

Abstract. We present a new class of statistical models for part-based object recognition. These models are explicitly parametrized according to the degree of spatial structure that they can represent. This provides a way of relating different spatial priors that have been used in the past such as joint Gaussian models and tree-structured models. By providing explicit control over the degree of spatial structure, our models make it possible to study questions such as the extent to which additional spatial constraints among parts are helpful in detection and localization, and the tradeoff between representational power and computational cost. We consider these questions for object classes that have substantial geometric structure, such as airplanes, faces and motorbikes, using datasets employed by other researchers to facilitate evaluation. We find that for these classes of objects, a relatively small amount of spatial structure in the model can provide statistically indistinguishable recognition performance from more powerful models, and at a substantially lower computational cost.

1 Introduction

Since the 1970's it has been observed that many objects can be represented in terms of a small number of parts arranged in a deformable configuration (e.g., [1, 2, 4, 5, 10–12, 14, 15, 17]). In such models, the appearance of each part is usually captured by a template, and the spatial relationships between parts are represented by spring-like connections between pairs of parts. Recently there has been a considerable resurgence in the use of these models for object recognition – both for detection and localization – and in learning models from example images. Particular emphasis has been on the recognition of generic *classes* of objects using models that are learned from specific examples.

The models that have been used to capture geometric relationships between the parts of an object differ substantially in their representational power and computational complexity. On one hand, joint Gaussian models (e.g., [4, 5, 11])

[1] David Crandall is supported by a NSF Graduate Research Fellowship.

have been used to explicitly capture spatial dependencies between all pairs of object parts, but the detection and localization algorithms that use these models rely on search heuristics in order to be computationally tractable. On the other hand, tree-structured graphical models (e.g., [10, 14]) have been used to efficiently detect and localize certain kinds of objects such as humans and faces, but are only able to explicitly capture a small fraction of the spatial dependencies between the parts of an object. An important goal of this paper is to improve our understanding of such tradeoffs between representational power and computational complexity for part-based recognition. We do this by introducing a family of spatial priors that provide explicit control over the degree of spatial structure that can be represented.

We use a problem formulation similar to the one in [10, 12], where for detection or localization a single overall problem is solved that takes into account both how well individual parts match the image data at each location and also the global spatial arrangement of parts. This framework is different from most other object recognition approaches (e.g. [4, 11]) that first perform feature detection to find possible locations for each part in an image and then use the detected feature locations to search for good object configurations. These methods have been popular because the explicit feature detection step reduces the number of object configurations that must be considered, but they have the disadvantage that false-negatives in the feature detection step can prevent parts from being properly localized. In [10] an efficient method was developed for tree-structured models that did not use feature detection, instead considering both part appearance and global spatial configuration at once. That method is able to provably compute the optimal object configuration in an image without explicitly searching the entire configuration space. A disadvantage to that method is that tree-structured models may not always be appropriate because of the relatively weak spatial structure that trees can capture.

In this paper we extend the implicit search techniques of [10] in order to efficiently perform object recognition without feature detection using a class of spatial priors defined by graphs that we call k-fans. Models defined by k-fans provide a natural family of priors for part-based recognition. The parameter k controls both the representational power of the models and the computational cost of doing inference with them. When $k = 0$, the locations of the object parts are independent. As k increases the spatial prior captures more information. When $k = 1$ the graphical structure of the prior is a star graph. For $k = n - 1$ (where n is the number of parts in the model) there are no conditional independencies among the part locations as in the case of a joint Gaussian model. This family of models gives us a natural way of investigating the degree to which additional spatial constraints improve recognition and affect computational cost. Using more powerful (higher-k) models does not necessarily improve classification, as it can lead to over-fitting during learning.

Besides providing an explicit balance between representational power and computational cost, k-fan models have a strong geometrical foundation. In a k-fan model the locations of k distinguished parts can be used to define the pose

of an object. With this view recognition using k-fans is related to geometric alignment [13]. From a different perspective k-fans can be used to define constraints on sets of $k+1$ parts in the model. With this view recognition using k-fans is related to geometric invariants [6]. In both cases k-fan models generalize the geometric methods by explicitly modeling uncertainty and replacing hard constraints with soft constraints based on statistical models.

As our experimental results demonstrate, for certain object classes that have been used recently in the literature, such as motorbikes, faces and airplanes, a relatively small amount of spatial structure provides almost the same recognition accuracy that is obtained using more powerful models. For small values of k, recognition with k-fans is highly practical without relying on search heuristics or feature detection.

2 Part-based Statistical Models

The central principle underlying part-based modeling is the observation that many objects can be represented by a small number of parts arranged in a characteristic configuration. The spatial relationships between parts in such a model are captured by a set of parameters S, while the appearance of each part is characterized by a set of parameters A. The model for an object is defined by the pair $M = (S, A)$.

Consider an object model with n parts $V = (v_1, \ldots, v_n)$. The location of the object in an image is given by a configuration of its parts $L = (l_1, \ldots, l_n)$, where l_i is the location of the ith part. Throughout this paper we assume that the location of a part is given by a point in the image, $l_i = (x_i, y_i)$. Using Bayes' law, the probability that the object is at a particular location given an image and a fixed set of model parameters can be written as,

$$p_M(L|I) \propto p_M(I|L)p_M(L). \tag{1}$$

Here, $p_M(I|L)$ is the likelihood of observing image I given that a particular configuration of the object occurs in the scene, and $p_M(L)$ is the prior probability that the object configuration is L. In this paper we consider three fundamental problems that can be formulated in terms of these distributions:

1. **Detection** The detection problem is to decide if the image has an instance of the object (hypothesis w_1) or if the image is background-only (hypothesis w_0). It is natural to consider the ratio of the two likelihoods,

$$q = \frac{p_M(I|w_1)}{p_M(I|w_0)}, \tag{2}$$

and compare it to a threshold to make the classification decision. The numerator is usually computed by summing over all possible configurations L as described in Section 3.4.

2. **Localization** Assuming the object is present in the scene, the configuration that most likely corresponds to its true position is one with maximum posterior probability,
$$L^* = \arg\max_L p_M(L|I).$$

3. **Supervised learning** The maximum-likelihood estimate of the model parameters given a set of labeled training images $\{(I_1, L_1), \ldots, (I_T, L_T)\}$ is,
$$S^* = \arg\max_S \prod_i p_M(L_i),$$
$$A^* = \arg\max_A \prod_i p_M(I_i|L_i).$$

The algorithmic complexity of solving these three problems is highly dependent on the form of the likelihood model $p_M(I|L)$ and the spatial prior $p_M(L)$. In the next section we discuss a particular likelihood model which has important structural properties, while the focus of the rest of the paper is primarily on the form of the spatial prior.

2.1 Appearance

For computational purposes, the most important property of the appearance model is that $p_M(I|L)$ factors into two parts: a term which does not depend on the object configuration, and a product of functions each of which depends on the location of a single part. Because of this factorization, any independence assumption that is present in the spatial prior will also be present in the posterior. The majority of the recent work on part-based recognition has used a similar factorization. A notable exception is the patchwork of parts model in [2] which does not make this assumption in order to better capture overlapping parts.

In our work we use a simple template-based appearance model that operates on oriented edge maps in order to be relatively invariant to changes in image intensity. Let I be the output of an oriented edge detector, so that for each pixel p, $I(p)$ is either 0 indicating that there is no edge at p or a value in $\{1, \ldots, r\}$ indicating that there is an edge in one of r possible quantized orientations at p. We assume that the values of each pixel in the image are independent given the object configuration. The appearance of the ith part is given by a template $\mathcal{T}_i$. The probability that a pixel $p \in \mathcal{T}_i$ has value u is defined by a foreground model for that part, $f_i(p)[u]$. We further assume that each pixel in the background has value u with probability $b[u]$. The model parameters $A = ((\mathcal{T}_i, f_i), ..., (\mathcal{T}_n, f_n), b)$ encode the foreground model for each part and the background model.

Let w_0 be the hypothesis that the object is not present in the image. By our independence assumption we have,
$$p_M(I|w_0) = \prod_p b[I(p)].$$

We say that parts i and j do not overlap if $(\mathcal{T}_i \oplus l_i) \cap (\mathcal{T}_j \oplus l_j) = \emptyset$. Here $\oplus$ denotes Minkowsky addition, which is used to translate the templates according the locations of the parts. For a configuration L without overlap we have,

$$p_M(I|L) = p_M(I|w_0) \prod_{v_i \in V} g_i(I, l_i), \tag{3}$$

where

$$g_i(I, l_i) = \prod_{p \in \mathcal{T}} \frac{f_i(p)[I(p + l_i)]}{b[I(p + l_i)]}. \tag{4}$$

Each term in g_i is the ratio of the foreground and background probabilities for a pixel that is covered by template $\mathcal{T}_i$. In equation (3) the denominator of g_i cancels out the contribution of $p_M(I|w_0)$ for those pixels that are under some part. As long as we only consider configurations L without overlapping parts the likelihood function defined above is a true probability distribution over images, in that it integrates to one. When parts overlap this is an approximation. Note that for many objects the spatial prior $p_M(L)$ strongly encourages parts in the model to not overlap, thus making this a reasonable appearance model.

2.2 Spatial Prior

The spatial prior $p_M(L)$ represents geometric relationships between the parts of an object. The simplest form of the prior assumes that there are no spatial dependencies between parts, so that the part locations are independent of one another (the naive Bayes assumption). Under this assumption, $p_M(L)$ can be written as:

$$p_M(L) = \prod_{v_i \in V} p_M(l_i).$$

The detection and localization problems are particularly easy with this spatial prior. For localization it is only necessary to maximize $g_i(I, l_i)p_M(l_i)$ independently for each l_i. This can be done in $O(nh)$ time for a model with n parts and h possible locations for each part. But while this model yields computationally tractable recognition and learning procedures, it is unable to accurately represent multi-part objects since it captures no relative spatial information.

Another option is to make no independence assumptions on the locations of different parts by, for example, using a joint Gaussian model for the spatial distribution $p_M(L)$ (e.g. as in [5]). Learning a maximum-likelihood distribution from labeled images in this case is easy, by simply computing the sample mean and covariance of the labeled part locations. However it is not known how to perform exact inference using this spatial prior efficiently. To make inference tractable, various heuristics have been employed to reduce the search space. For example, feature detection is normally used to constrain the possible locations of each part.

Spatial models between the two extremes just described can be defined by making certain conditional independence assumptions. These assumptions are

commonly represented using an undirected graphical model (or Markov random field). Let $G = (V, E)$ be an undirected graph. The graph is used to define a distribution for the random variables $(l_1, \ldots, l_n)$ in the following way. The value for the location of v_i is independent of the values of all other nodes, conditioned on the values of the neighbors of v_i in the graph. The independence assumptions of the naive Bayes model are represented by a graph with no edges while a model with no independence assumptions such as the joint Gaussian corresponds to a complete graph.

Efficient learning and inference procedures for models with tree-structured spatial priors are known. The detection and localization problems can be solved in $O(nh^2)$ time using dynamic programming. Moreover, in many cases one can solve these problems in $O(nh)$ time – the same asymptotic time as the naive Bayes case where there are no dependencies between part locations (see [10]).

To summarize, we can imagine a spectrum of spatial priors, arranged according to the degree of spatial independence assumptions they make. On one end of the spectrum, we assume that all parts are spatially independent, so that the location of a given part does not constrain the location of *any* other part. Inference in this case is efficient but the object model is weak. At the other end are models that make no independence assumptions. This form of spatial prior can capture arbitrarily complex spatial relationships between part locations, but even for restricted cases it is not known how to perform exact inference efficiently. Tree-structured spatial priors fall in between the two extremes. In the following section, we introduce a family of spatial priors, called k-fans, which are explicitly parametrized according to where they fall along this spectrum.

3 *k*-fans

Now we consider a class of spatial priors that lie between the two extremes of the naive Bayes assumption and a fully-connected spatial model. Our goal is to find models with recognition performance comparable to a fully-connected model but that support fast procedures for exact (discrete) inference and learning. We start by considering a restricted form of tree model, the star graph, and then extend that model. A star graph is a tree with a central node that is connected to all other nodes. Let $G = (V, E)$ be a star graph with central node v_r. Undirected graphical models with a star structure have a particularly simple interpretation in terms of conditional distributions. The values of random variables associated with nodes $v_i \neq v_r$ are independent when conditioned on the value of v_r. This leads to the following factorization of the prior distribution,

$$p_M(L) = p_M(l_r) \prod_{v_i \neq v_r} p_M(l_i|l_r).$$

We can think of the central node v_r as a *reference* part. The position of other parts in the model are evaluated relative to the position of this reference part.

k-fans extend the star graph model to include more than one reference part. Let $R \subseteq V$ be a set of reference parts, and $\overline{R} = V - R$ be the remaining parts in a

model. Then a graph can be constructed which consists of a complete subgraph over the nodes in R, while each node in $\overline{R}$ is connected to every node in R (but to no other nodes). We call this graph a k-fan for $k = |R|$. Some examples of k-fans on six nodes are shown in Figure 1.

A *clique* in an undirected graph is a set of vertices for which there is an edge connecting every pair of nodes in the set. A k-fan can be seen as a collection of cliques of size $k+1$ connected together along a common clique of size k. The k nodes in the common clique are the reference parts R.

A k-fan can be constructed by starting with a k-clique corresponding to the reference nodes and sequentially adding new nodes by connecting each of them to the reference nodes and nothing else. With this view it is clear that k-fans are a special class of k-trees [16]. In particular k-fans are decomposable (also known as triangulated or chordal) graphs. Because k-fans are k-trees there are standard algorithms that can perform inference with these models in time that is polynomial in n and exponential in k, where n is the number of nodes in the graph [3]. An important difference between k-fans and arbitrary k-trees is that k-fan models can be learned in time polynomial in n and exponential in k while learning a k-tree is NP-hard even for small k.

As k grows from 0 to $n-1$ we get a set of graphs which intuitively interpolate between the empty graph and the complete graph on n nodes. Thus k-fans define a class of graphical models of increasing expressive power.

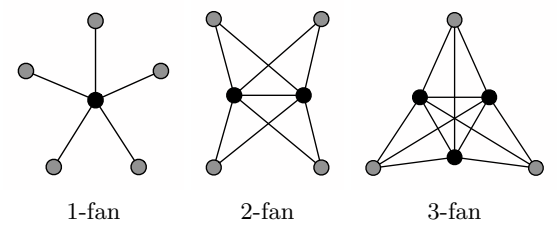


Fig. 1. Some k-fans on six nodes. The reference nodes are shown in black while the regular nodes are shown in gray.

We claim that k-fans form an important class of graphical models for part-based recognition. These are exactly the models where the locations of the non-reference parts are conditionally independent given the locations of the reference parts. Let $R = \{v_1, \ldots, v_k\}$ be the reference parts in a k-fan. We denote by $l_R = (l_1, \ldots, l_k)$ a particular configuration of the reference parts. The spatial prior defined by a k-fan can be written in conditional form as,

$$p_M(L) = p_M(l_R) \prod_{v_i \in \overline{R}} p_M(l_i | l_R). \tag{5}$$

In general both the localization and detection problems for models with spatial priors based on k-fans can be solved in $O(nh^{k+1})$ time, where n is the number

of parts in the model and h is the number of locations in the image. Thus k controls the computational complexity of inference with these models. With the additional assumption that $p_M(L)$ is Gaussian we can use distance transforms and convolutions to solve the inference problems in $O(nh^k)$, as described below. In practice the running time can be further improved using conservative pruning heuristics that eliminate low-probability configurations.

For learning k-fan models it will be useful to write the spatial prior in terms of marginal distributions,

$$p_M(L) = \frac{\prod_{v_i \in \overline{R}} p_M(l_i, l_R)}{p_M(l_R)^{n-(k+1)}}. \tag{6}$$

The numerator is the product of marginal probabilities for the $n-k$ maximal cliques and the denominator involves the marginal probability for the nodes shared by all maximal cliques (the so-called separator set which in this case is R). This is a special form of the factorization for a triangulated graph, which is the ratio of a product over maximal cliques and a product over separators [7].

3.1 Geometric Interpretation

As mentioned in the introduction there is a natural connection between k-fan models and geometric alignment [13]. In a k-fan model the locations of the reference parts can be used to compute a global transformation aligning a geometrical model and the image. This alignment defines an ideal location for each non-reference part, and deviations from these ideal locations can be measured by the conditional distributions $p_M(l_i|l_R)$.

There is a also a close connection between k-fan models and object recognition using geometric invariants. Each maximal clique in a k-fan consists of exactly $k+1$ parts, and the location of these parts can be used to define shape constraints that are invariant to certain geometric transformations (see [6]). The number of reference parts controls the type of geometric invariants that can be represented.

In a k-fan the location of a non-reference part can be described in a reference frame defined by the locations of the k reference parts. For example, when $k=1$ the location of a non-reference part can be described relative to the location of the single reference part. The values $l'_i = l_i - l_r$ are invariant under translations, so 1-fans can be used to define translation invariant models. For the case of $k=2$ the two reference parts can be used to define models that are invariant to rigid motions and global scaling. When $k=3$ we can use the three reference parts to define an affine basis in the image plane; if the location of every non-reference part is described in this basis we obtain affine invariant models. These models are important because they capture arbitrary views of planar objects under orthographic projection.

To enforce geometric invariants over $k+1$ parts one could define $p_M(l_i|l_R)$ to be one if the $k+1$ locations satisfy a geometric constraint and zero otherwise. In general our models capture soft geometric constraints, giving preference to

configurations that satisfy relationships on $k+1$ features as much as possible. The distribution over the reference part locations $p_M(l_R)$ could be uniform in the case where all geometric constrains are defined in terms of $k+1$ parts. Non-uniform distributions can be used to represent interesting classes of non-rigid objects.

3.2 Gaussian *k*-fans

We now consider k-fan models with the additional constraint that $p_M(L)$ is a Gaussian distribution. For a Gaussian model the marginal distribution of any subset of variables is itself Gaussian. Let μ_R and Σ_R be the mean and covariance for the locations of the reference parts. The marginal distribution of the reference parts together with one non-reference part is given by a Gaussian with mean and covariance,

$$\mu_{i,R} = \begin{bmatrix} \mu_i \\ \mu_R \end{bmatrix}, \quad \Sigma_{i,R} = \begin{bmatrix} \Sigma_i & \Sigma_{iR} \\ \Sigma_{Ri} & \Sigma_R \end{bmatrix}. \tag{7}$$

These can be used to define the spatial prior in terms of equation (6). We will use this for learning Gaussian k-fans. For inference we use the conditional form of the prior in equation (5). For a Gaussian distribution, conditioning on a set of variables preserves the Gaussian property. In particular, the conditional distribution of a non-reference part location given particular locations for the reference parts $p_M(l_i|l_R)$ has mean and covariance,

$$\mu_{i|R}(l_R) = \mu_i + \Sigma_{iR}{\Sigma_R}^{-1}(l_R - \mu_R), \tag{8}$$

$$\Sigma_{i|R} = \Sigma_i - \Sigma_{iR}{\Sigma_R}^{-1}\Sigma_{Ri}, \tag{9}$$

Note how the covariance $\Sigma_{i|R}$ is independent of the location of the reference parts. This is a non-trivial property that enables the use of distance transforms and convolutions to obtain faster inference algorithms than is possible with non-Gaussian models, as we will show in Sections 3.4 and 3.5.

3.3 Learning

We can learn the spatial prior for Gaussian k-fan models from labeled images using a maximum likelihood criterion. For a fixed set of reference parts, estimating the maximum likelihood parameters S^* involves estimating the mean and covariances in (7). These can be obtained from the sample mean and covariance of the labeled configurations.

The more interesting case is when the reference parts are not fixed. In this situation all possible reference sets of size k can be considered to find the set R that yields the best possible model. There are $\binom{n}{k}$ possible reference sets, which is not very large for small values of k. For each reference set we compute the maximum likelihood model parameters using the sample mean and covariance, as described above. We select the best reference set by choosing the set R that maximizes the likelihood of observing the training data given the model.

Learning the appearance parameters A^* for the models described in Section 2.1 using labeled training data is also simple. To estimate f_i, the position of the ith part in each training example is used to align the training images. The maximum likelihood estimate for $f_i(p)[v]$ is simply the frequency that pixel p has value v on the aligned data. The only parameter that is not learned from the data is the size and shape of the template $\mathcal{T}_i$. For the experiments shown in this paper we used square windows of a fixed size.

3.4 Detection

For detection we consider the likelihood ratio in (2). The numerator of this ratio, which is the probability of an image given that it contains the object, can be expressed as a sum over all possible object configurations,

$$p_M(I|w_1) = \sum_L p_M(L) p_M(I|L).$$

Using the likelihood function (3) we see that

$$\frac{p_M(I|w_1)}{p_M(I|w_0)} = \sum_L p_M(L) \prod_{v_i \in V} g_i(I, l_i).$$

For a k-fan model the sum over all configurations L can be factored using the conditional form of the spatial prior in (5). For each $v_i \in \overline{R}$ we define

$$\alpha_i(l_R) = \sum_{l_i} p_M(l_i|l_R) g_i(I, l_i).$$

Now the likelihood ratio can be computed as,

$$\frac{p_M(I|w_1)}{p_M(I|w_0)} = \sum_{l_R} p_M(l_R) \prod_{v_i \in R} g_i(I, l_i) \prod_{v_i \in \overline{R}} \alpha_i(l_R).$$

Note that each α_i can be computed by brute force in $O(h^{k+1})$ time, while the likelihood ratio can be computed using the α_i in $O(nh^k)$ time. This procedure gives an $O(nh^{k+1})$ algorithm for computing the likelihood ratio.

For the case of a Gaussian k-fan we can compute the likelihood ratio even faster, using convolutions. For each non-reference part v_i we have,

$$p_M(l_i|l_R) = \mathcal{N}(l_i, \mu_{i|R}(l_R), \Sigma_{i|R}),$$

a Gaussian distribution with mean and covariance given by equations (8) and (9). Let $\alpha'_i(l_i)$ be the convolution of $g_i(I, l_i)$ with a Gaussian kernel of covariance $\Sigma_{i|R}$. It is not hard to see that,

$$\alpha_i(l_R) = \alpha'_i(\mu_{i|R}(l_R)).$$

So each α_i can be implicitly computed by a convolution in the space of possible part locations. This can be done in $O(h \log h)$ time instead of $O(h^{k+1})$.

The overall running time of the likelihood ratio computation for the case of a Gaussian k-fan model is $O(nh^k + nh \log h)$. Note that for a 1-fan model this is almost the same as $O(nh)$, the time that it would take to compute the likelihood ratio if the locations of the parts were completely independent. The $\log h$ dependency can be removed by using linear time methods that approximate Gaussian convolutions, such as the box-filter technique in [18].

3.5 Localization

For localization we look for an object configuration L^* with maximum posterior probability. Using Bayes law the posterior distribution for a k-fan model can be written in terms of the likelihood function (3) and the spatial prior (5). By manipulating the terms we get,

$$p_M(L|I) \propto p_M(l_R) \prod_{v_i \in R} g_i(I, l_i) \prod_{v_i \in \overline{R}} p_M(l_i|l_R) g_i(I, l_i).$$

For any $v_i \in \overline{R}$ the quality of an optimal location for the ith part can be expressed as a function of the reference locations,

$$\alpha_i^*(l_R) = \max_{l_i} p_M(l_i|l_R) g_i(I, l_i). \tag{10}$$

Using the α_i^* we can express the posterior probability of an optimal configuration for the object with particular reference locations l_R as,

$$\beta^*(l_R) = p_M(l_R) \prod_{v_i \in R} g_i(I, l_i) \prod_{v_i \in \overline{R}} \alpha_i^*(l_R). \tag{11}$$

These functions can be used to compute an optimal configuration for the object in time polynomial in the number of parts n and the number of locations for each part h (but exponential in k). Each α_i^* can be computed by brute force in $O(h^{k+1})$ time, while β^* can be computed in $O(nh^k)$ time. An optimal configuration for the reference parts l_R^* is one maximizing β^*. Finally, for each non-reference part we select l_i^* maximizing $p_M(l_i|l_R^*) g_i(I, l_i)$. This can be done in $O(h)$ time. The overall running time of this procedure is $O(nh^{k+1})$, which is reasonable for very small k.

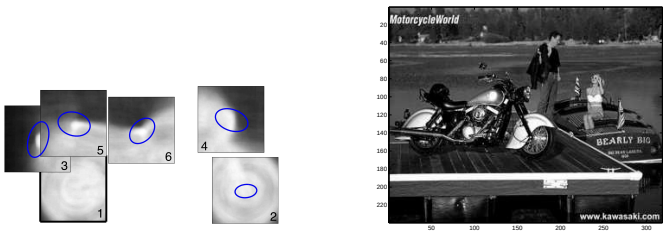
As in the case of detection we can speed up the localization procedure for Gaussian k-fans. For localization the role of convolutions is played by generalized distance transforms [9]. In this case the running time is reduced to $O(nh^k)$.

4 Inference with Gaussian *k*-fans

We have shown that in theory it is possible to perform exact inference (detection and localization) with Gaussian k-fan models efficiently without relying on feature detection. It turns out that the inference algorithms are also intuitive and

straightforward to implement. In this section we describe how the localization algorithm works using generalized distance transforms, with a running example to illustrate each step of the process.

Figure 2(a) shows a diagram of a 1-fan model with six parts for detecting motorbikes. A simplified representation of the appearance model template of each part is shown, giving the probability of an edge at each location (disregarding orientation). Bright spots in the templates correspond to locations with higher edge probabilities. In this model the reference part is the back wheel and each non-reference part is positioned according to its mean location $\mu_{i|R}$ with respect to the reference. The figure also shows the conditional covariance $\Sigma_{i|R}$ of each non-reference part location, represented by an ellipse plotted at two standard deviations away from the mean. We will describe how the localization procedure works using this motorbike model on the sample input image shown in Figure 2(b). There are three steps to the procedure which are outlined below.



(a) (b)

Fig. 2. A six part 1-fan model for motorbikes, with the back wheel as the reference part and a sample input image.

4.1 Step 1: Apply part appearance operators

The first step in performing localization is to evaluate $g_i(I, l_i)$ as defined in equation (4) for each part at each possible location. This produces a quality map for each part, indicating how well the part appearance model matches the local image information at each location. In practice we compute

$$C_i(l_i) = -\log g_i(I, l_i)$$

and think of $C_i(l_i)$ as the cost of placing part i at location l_i. While these costs have a particular form defined by the statistical model one can think of this step as essentially doing template matching with an edge template for each part. We can use the fact that edge images are sparse to compute the quality maps quickly.

Figure 3(a) shows the quality maps that were generated by the motorbike model on the sample input image, with good locations (low costs) represented

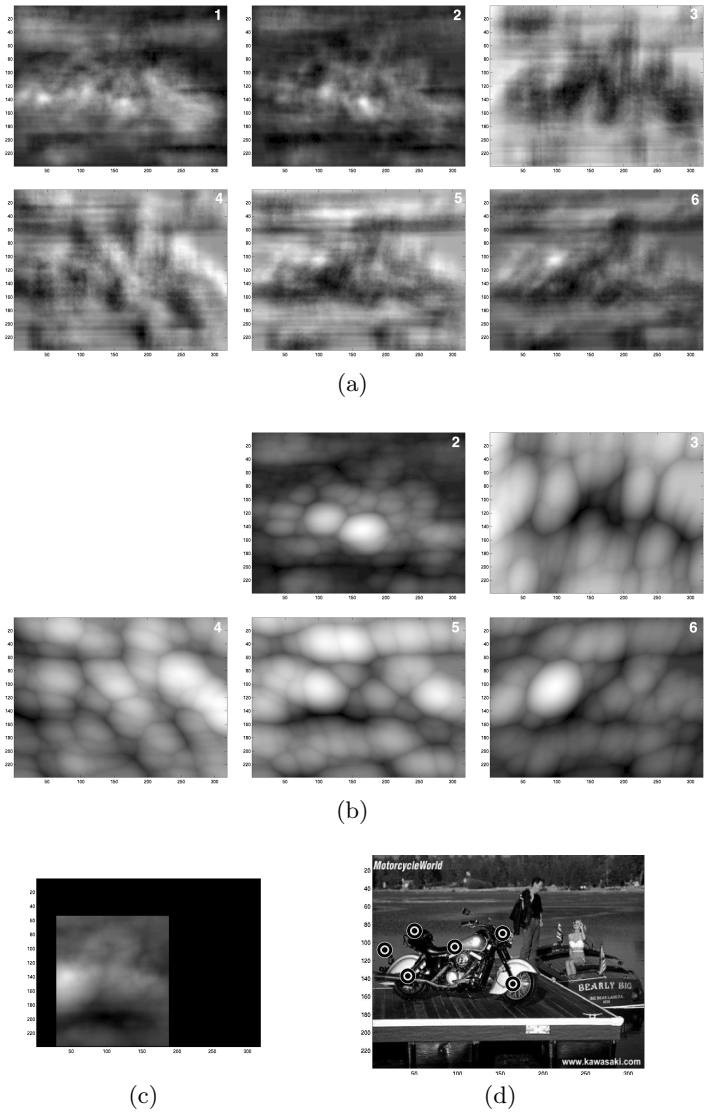


Fig. 3. Illustration of the localization procedure: (a) quality maps indicating the cost of placing each part at each location, with brighter intensity indicating better locations, (b) result of the distance transform applied to the quality maps of the non-reference parts, (c) final quality map showing the cost of placing the reference part at each location, and (d) final result, showing the localized part locations.

by brighter intensities. Note that the individual quality maps are quite noisy, so that simply choosing the best location for each part without taking into account their relative positions (as in the naive Bayes method) would generate poor localization results. For example, the front and back wheel appearance models are similar and there are peaks at the location of the front wheel in the back wheel quality map, and vice-versa.

4.2 Step 2: Apply distance transforms

The next step takes into account the spatial dependencies in the model as encoded by the conditional covariances of each non-reference part with respect to the references. This is done by computing the generalized distance transform of the quality map for each non-reference part to allow for variations in its position relative to the references. The output is a new quality map $\mathcal{D}_i(l_i)$ for each non-reference part. The results of this step on the running example are shown in Figure 3(b). The transformation "spreads" the quality maps produced by the appearance models. Intuitively the resulting cost $\mathcal{D}_i(l_i)$ is low near locations where the original cost is low. The size and shape of this spreading operation is controlled by the conditional covariances $\Sigma_{i|R}$ for each part.

The new costs are defined by,

$$\mathcal{D}_i(x) = \min_y \; C_i(y) + \frac{(x-y)\Sigma_{i|R}^{-1}(x-y)}{2}.$$

The algorithm in [9] can be used to compute these distance transforms in time linear in the number of possible locations for each part.

4.3 Step 3: Combine evidence

The last step in the localization procedure is to combine the distance transformed quality maps for the non-reference parts with the quality maps of the reference parts. The result is a cost for every configuration of the reference parts that takes into account the placement of the whole model. More specifically the cost for each placement of the reference parts encorporates the cost of the best placements of all the other parts. This is preciselly the negative logarithms of $\beta^*(l_R)$ in equation (11), up to an additive constant.

The procedure is particularly simple for the case of a translation invariant 1-fan model. In this case the computation of $-\log(\beta^*(l_R))$ up to an additive constant can be done as follows. We shift the distance transformed quality maps $\mathcal{D}_i(l_i)$ by the ideal position of part i relative to the reference part and sum these shifted quality maps together with the quality map for the reference part $C_r(l_r)$. The resulting map for the sample input image is shown in Figure 3(c). An optimal location for the reference part (the back wheel) l_r^* is determined by picking a lowest cost location in this map. After that the locations of the other parts can be found by selecting l_i^* for each non-reference part so as to maximize

$p_M(l_i|l_r^*)g_i(I, l_i)$. The final localization results in the sample image are shown in Figure 3(f).

Performing localization using a k-fan model with $k > 1$ can be done in a similar way. In general equation (11) can be rewritten as

$$-\log(\beta^*(l_R)) = -\log(p_M(l_R)) + \sum_{v_i \in R} C_i(l_i) + \sum_{v_i \in \overline{R}} \mathcal{D}_i(\mu_{i|R}(l_R)) + Z.$$

For a 1-fan $-\log(\beta^*(l_R))$ is a two-dimensional quality map but for general k it is a $2k$ dimensional map. To compute $-\log(\beta^*(l_R))$ we iterate over all possible reference locations and evaluate the sum above.

5 Experiments

This section presents results from experiments we have conducted to characterize the detection and localization performance of k-fans as k is varied. Since the running time varies exponentially with k, it is clear that in practice it is best to choose the lowest value of k that still provides adequate detection and localization performance. We also compare our results to those of Fergus et al [11] who used full multivariate Gaussians (i.e. $n-1$-fans, where n is the number of parts) as the spatial priors. However, since inference with this spatial model is intractable, they performed approximate inference using feature detection and various search heuristics. One of the goals of our experiments was to compare the performance of the *exact* (discrete) inference method for k-fans with small k to their *approximate* inference method for full Gaussian prior models.

To facilitate comparison of results with previous work we used some of the datasets from [11]: airplanes (800 images), faces (435 images), motorbikes (800 images), and background scenes (800 images). To further facilitate evaluation, we considered only the case of Gaussian k-fans (that is, we did not use the reference parts to define a geometric basis as described in Section 3.1). We tried to reproduce the experimental protocol of [11] as closely as possible, including using the same partitioning of the data into training and test images. We also pre-scaled all images so that object width was roughly uniform, using the same ground truth bounding boxes used in their experiments. To prevent biases related to image size, we padded out all images to a large, uniform size.

5.1 Learning the Models

As in [11], six parts were used to model each object. For airplanes we used the front and back landing gear, nose, wing tip, tail, and vertical stabilizer. For faces we used the two eyes, nose, two corners of the mouth, and chin. For motorbikes, the front and back wheel, headlight and tail light, and the front and back of the seat were used. Ground truth was collected by hand-labeling the training images. Note that [11] used an unsupervised training method but we should not expect supervised learning to necessarily give better results than unsupervised

learning – a supervised approach is limited by the quality of the parts chosen and the accuracy of the hand-labeled ground truth.

The models were learned from labeled examples using the procedure described in Section 3.3. To learn the appearance model for a given part, a fixed-size patch surrounding the labeled part location was extracted from each training image. Canny edge detection was used to generate edge maps. Edge orientation was quantized into four directions (north/south, east/west, northeast/southwest, northwest/southeast) and represented as four separate binary edge maps. Note that opposing directions were quantized into the same bin. This prevents edge directions from changing when an object is moved from a light background to a dark background or vice-versa. Morphological dilation was applied on each map independently. Finally, foreground model probabilities were estimated by computing the frequency of each of the 16 possible combinations of edge orientations at each position in the template across all training images. The background model probabilities were estimated from the observed density of edges in background images.

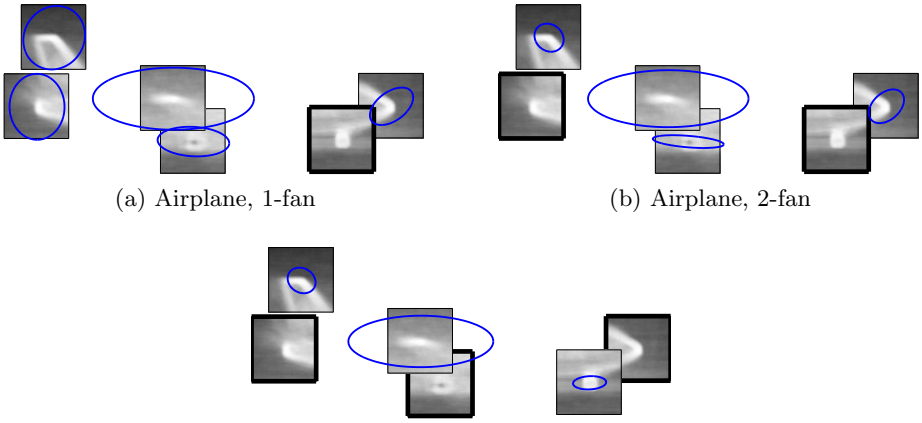
Figure 4 illustrates some of the models we learned. Note that in each case the configuration of parts is readily recognizable as a prototype of the object. It is particularly interesting to compare the 1-fan and 2-fan models for the airplanes. Note that as k increases, the variability in the non-reference part locations (as shown by the ellipses) decreases substantially. Figure 5 illustrates the appearance model for the front wheel of the motorbike model in detail.

5.2 Detection Results

For detection we found an optimal configuration for the object in each test image, using the procedure described in Sections 3.5 and 4, and then used that location to approximate the likelihood ratio in equation (2). With this approach each positive detection comes with a particular localization. We kept all parameters exactly the same across the different object classes (template size = 50×50, dilation radius = 2.5 pixels).

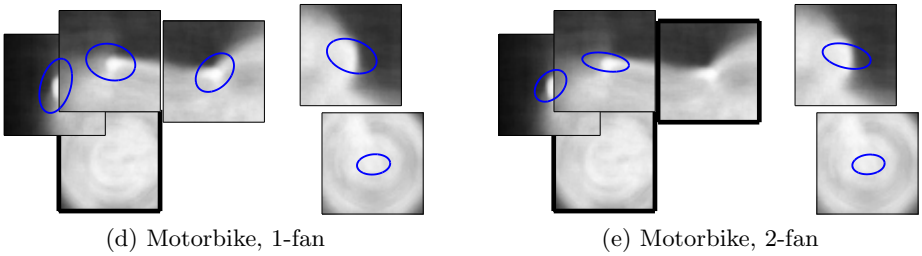
Figure 6 shows ROC curves generated from these experiments. For each object class, the figure compares ROC curves for k-fans with k ranging from 0 (no structure) to 2. We observe that for motorbikes, high accuracy is achieved using 0-fans, and adding spatial constraints gives little improvement. On the other hand, for airplanes, 1-fans perform significantly better than 0-fans, and 2-fans perform significantly better than 1-fans, indicating that increasing degrees of spatial constraints give better performance. We conclude that the appropriate amount of spatial structure in the model varies from object to object.

Table 1 summarizes the recognition accuracy at the equal ROC points (point at which the true positive rate equals one minus the false positive rate). We note that our equal ROC results compare favorably with those obtained using full multivariate Gaussian structural models (with heuristics that make inference sub-optimal but computationally tractable) in [11]. They report 90.2%, 92.5% and 96.4% for airplanes, motorbikes and faces respectively, under the same experimental conditions.

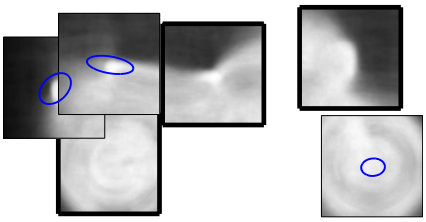


(a) Airplane, 1-fan (b) Airplane, 2-fan

(c) Airplane, 3-fan

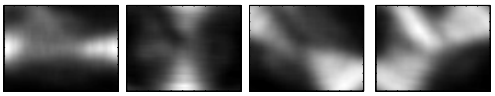


(d) Motorbike, 1-fan (e) Motorbike, 2-fan



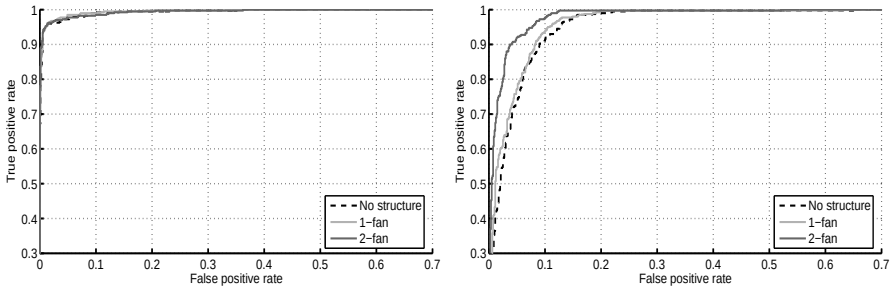
(f) Motorbike, 3-fan

Fig. 4. Illustration of some of the learned models. Images (a) through (f) show part appearance models positioned at their mean configuration. The reference parts have a black border around them. The ellipses illustrate the conditional covariances for a non-reference part given the locations of the references. High intensity pixels represent high edge probabilities. For clarity, just the probability of an edge is shown, although the actual models capture probabilities of each individual edge orientation. Note how the locations of parts are more constrained as k increases.



Vertical Horizontal 45 degrees 135 degrees
Probability of edge at each orientation

Fig. 5. The appearance model for the front wheel of the motorbike model.



(a) (b)

Fig. 6. Detection results for (a) motorbiks and (b) airplanes. Note that the ROC curves are truncated at a false positive rate of 0.7 and a true positive rate of 0.3.

	Planes	Bikes	Faces
0-fans	90.5%	96.5%	98.2%
1-fans	**91.3%**	**97.0%**	98.2%
2-fans	**93.3%**	97.0%	98.2%

Table 1. Equal ROC performance for the detection experiments. A boldface number for a k-fan indicates a statistically significant difference between the areas under the ROC curves of the $k-1$ and k-fan models (with 95% confidence).

We applied the statistical test in [8] to judge the differences in areas under the ROC curves of the various models. These results are also shown in Table 1. For each object class we computed the probability that the area under the ROC curve for the k-fan model is significantly different from the area under the ROC curve for the model with one less reference part. Differences significant at a greater than 95% confidence level are shown in boldface.

We also conducted multi-class detection experiments to test the ability of the models to differentiate between the three different object classes and the background images. For each test image, the three object detectors were applied, and the object class with the highest likelihood was chosen. That likelihood was compared to the threshold at the equal ROC point to decide between that object class and the background class. The results are shown in Table 2. The performance of multi-class recognition is similar to the single class case. The use

	0-fan				1-fan				2-fan			
	Planes	Bikes	Faces	BG	Planes	Bikes	Faces	BG	Planes	Bikes	Faces	BG
Planes	357	10	0	33	362	5	0	33	370	8	0	22
Bikes	4	382	0	14	4	384	0	12	4	384	0	12
Faces	3	9	205	0	3	8	206	0	1	9	207	0
Background	72	28	0	700	68	24	0	708	53	23	0	724

Table 2. Confusion matrices for the multi-class detection experiments. Rows correspond to actual classes, while columns correspond to predicted classes.

of relatively accurate probabilistic models allows for direct comparison between the scores of each object class without tuning weighting parameters.

As in [11], we also tested the detectors in a setting where the object scale was not known (i.e. images were not pre-scaled to a uniform object width). The object widths varied between about 200 and 700 pixels for the motorbike and plane categories, while the face dataset had very little scale variation. We applied the detectors at four different scales to each image and chose the scale having the highest-likelihood detection. Recognition performance in this experiment was comparable to the case of pre-scaled images.

The average running time per image of the detection algorithm on these datasets on a 3GHz Pentium 4 is approximately 0.1 seconds for a 1-fan model, 3.3 seconds for a 2-fan model, and 37.6 seconds for a 3-fan model.

5.3 Localization Accuracy

Figure 7 illustrates some localization results produced by our system on the motorbike dataset, showing precise localization of the parts despite substantial variability in their appearances and configurations. Recent work has generally focused on evaluating detection performance but we believe it is also important to evaluate the accuracy of localization. For example, some applications may benefit from knowing the exact locations of each part individually. Also, examining localization performance helps to reveal the evidence that the detection algorithm is using to perform its classification decisions, and to ensure that it is not exploiting "unfair" biases in the image data, such as image size or patterns in the image backgrounds. For each object class, for the subset of images that were correctly classified during the detection task at the equal ROC point, the part locations produced by our system were compared to hand-labeled ground truth. We computed the trimmed means (at 75% and 90%) of the Euclidean distances (in pixels) between estimated locations and the ground truth. For the motorbike models the localization errors are reasonably small (less than 10 pixels) for most parts when $k > 0$, while the errors for faces are less than 2 pixels. Table 3 summarizes the results for the motorbikes models. In this case the localization accuracy is high for most parts when using a model without spatial structure. The accuracy increases as we add spatial constraints even when recognition performance does not increase.

Fig. 7. Some localization results. In each of these cases all parts were localized correctly.

Model	rear wheel		front wheel		headlight		tail light		back of seat		front of seat	
	75%	90%	75%	90%	75%	90%	75%	90%	75%	90%	75%	90%
No structure	15.6	34.4	1.9	2.3	10.9	18.8	12.0	19.3	21.6	33.9	6.3	12.2
1-fan	2.1	12.5	1.9	2.3	10.9	18.6	11.4	18.7	20.6	32.9	6.3	12.0
2-fan	1.9	2.4	1.9	2.3	10.1	16.6	11.0	18.3	17.2	28.5	5.4	9.3

Table 3. Part localization errors for the correctly detected motorbike images, showing 75% and 90% trimmed means of Euclidean distance between estimated part locations and ground truth.

References

1. Y. Amit. *2D Object Detection and Recognition, Models, Algorithms, and Networks.* MIT Press, 2002.
2. Y. Amit and A. Trouvé. Pop: Patchwork of parts models for object recognition. 2005.
3. U. Bertele and F. Brioschi. *Nonserial Dynamic Programming.* Academic Press, 1972.
4. M.C. Burl and P. Perona. Recognition of planar object classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.
5. M.C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *European Conference on Computer Vision*, 1998.
6. S. Carlsson. Geometric structure and view invariant recognition. *Phil. Trans. R. Soc. Lond. A*, 359(1740), 1998.

7. R. F. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.
8. DeLong E.R., DeLong D.M., and Clarke-Pearson D.L. Comparing the areas under two or more correlated roc curves: a non-parametric approach. *Biometrics*, 44(3), 1998.
9. P.F. Felzenszwalb and D.P. Huttenlocher. Distance transforms of sampled functions. September 2004. Cornell Computing and Information Science Technical Report TR2004-1963.
10. P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1), 2005.
11. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
12. M.A. Fischler and R.A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22(1), 1973.
13. D.P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, November 1990.
14. S. Ioffe and D.A. Forsyth. Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1), 2001.
15. P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification and image indexing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
16. D. J. Rose. On simple characterizations of k-trees. *Discrete Mathematics*, 7(3-4):317–322, 1974.
17. H. Schneiderman and T. Kanade. Probabilistic formulation for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.
18. W.M. Wells, III. Efficient synthesis of Gaussian filters by cascaded uniform filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2), 1986.