Recognition by Probabilistic Hypothesis Construction

Pierre Moreels, Michael Maire, and Pietro Perona

California Institute of Technology, Pasadena CA 91125, USA,
pmoreels@vision.caltech.edu

Abstract. We present a probabilistic framework for recognizing objects in images of cluttered scenes. Hundreds of objects may be considered and searched in parallel. Each object is learned from a single training image and modeled by the visual appearance of a set of features, and their position with respect to a common reference frame. The recognition process computes identity and position of objects in the scene by finding the best interpretation of the scene in terms of learned objects. Features detected in an input image are either paired with database features, or marked as clutters. Each hypothesis is scored using a generative model of the image which is defined using the learned objects and a model for clutter. While the space of possible hypotheses is enormously large, one may find the best hypothesis efficiently – we explore some heuristics to do so. Our algorithm compares favorably with state-of-the-art recognition systems.

1 Introduction

In the computer vision literature there is broad agreement that objects and object categories should be represented as collections of parts (or features) which appear in a given mutual position or shape (eg side-by-side eyes, a nose below them etc). Each feature contains local information describing the image content [2,3]. There is, however, disagreement as to the best tradeoff in this design space. On one hand, one may wish to represent the appearance and position of parts in a careful probabilistic framework, which allows to generate principled learning and detection algorithms. One example of this approach is the 'constellation model' [4] which has been successfully applied to unsupervised learning and recognition of object categories amongst clutter [5,6]. This approach is penalized by a large number of parameters that are needed to represent appearance and shape and by algorithmic complexity – as a result there is a practical limit to the size of the models that one can use, typically limiting the number of object parts below 10. On the other hand, one finds in the literature models containing hundreds of features. In this case the authors dramatically simplify the way appearance and position are modeled as well as the algorithms used to learn and match models to images. A representative of this approach is David Lowe's algorithm [7,8] which can recognize simultaneously and quickly multiple individual objects (as opposed to categories).

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3021, pp. 55–68, 2004.

We are interested in exploring whether probabilistically rigorous modeling may be extended to yield practical data structures and algorithms for models that contain hundreds of features. To this end, we modify the constellation model [6,9] to incorporate a number of attractive features presented by Lowe: using a KD-tree for efficiently associating features by appearance as well as computing feature positions with respect to a common reference frame rather than with respect to each other. Additionally, we pool representational parameters amongst features. As a result, it is possible to learn models quickly based on a single example; additionally, the system gains the robustness associated with using a large number of features while also offering an expressive probabilistic model for verifying object presence. One additional contribution is exploring efficient algorithms for associating models with images that are based on this probabilistic model and the A* search technique [10,11].

In section 2 we review the feature matching and constellation model approaches upon which this paper builds. Section 3 details the probabilistic framework used in our recognition system. Section 4 describes the algorithm for incrementally constructing a high probability hypothesis without exploring the entire hypothesis space. In section 5, we discuss the task of learning. In section 6, we compare our systemès performance against that of a pure feature matching approach. Finally, in section 7, we present conclusions and discuss areas for further research.

2 Related Research

A feature-based recognition approach recently developed by Lowe [7,8] consists of four stages: feature detection, extraction of feature correspondences, pose parameter estimation, and verification. Features are computed over multiple scales, at positions that are extrema of a difference-of-Gaussian function. An orientation is assigned to a feature using the histogram of local image gradients. Each feature's appearance is represented by a vector constructed from the local image region, sampled relative to the feature orientation. A k-d tree structure, modified with backtracking for search efficiency [12], is the central component of a database used to perform efficient appearance-based feature matching. Each match between scene and model features suggests a position, orientation, and scale for the model within the scene. Recognition is achieved by grouping similar model poses using a Hough transform and then explicitly solving for the transformation from model to scene coordinates.

The constellation model [4,5,6,9] also relies on matching image parts, but typically uses on the order of 5 features, whereas Lowe uses hundreds of features. Rather than restricting features to a rigid position, the constellation model uses a joint probability density on part positions. In addition, a probabilistic model for feature appearance is used, permitting the quality of matches to be measured. One drawback of the constellation model is the high number of training samples required, although recent work by Fei Fei et al [13] proved that learning can be efficiently achieved with few examples. Another disadvantage of the constellation

model lies in the large computation time required in order to learn feature configurations, limiting it to use of a relatively small number of parts for each object category. In our adaptation of the constellation approach for individual object recognition, this limitation disappears, at a slight cost to the model's generality.

3 Probabilistic Framework

We model individual objects as constellations of features. Features are generated by applying Lowe's feature detector [7,8] to each training image. Each feature has a position, orientation, and scale within the object model as well as a feature vector describing its appearance. We learn probabilistic models for foreground and background feature appearance. The collection of models extracted from training images, together with a k-d tree of model features searchable by appearance, forms the *database*.

Features are generated from a scene using the same procedure applied to training images. If a model is present, each of its features has a chance of appearing as a scene feature. We also expect spurious background detections in the scene. A *hypothesis* assigns each scene feature to either the background or a model feature. It also specifies the pose of each model present in the scene. A hypothesis may indicate the presence of multiple instances of the same object, each in a different pose.

The task of the recognition algorithm is to find the hypothesis that best explains the scene. The solution is the hypothesis with maximum probability conditioned on both the observed scene features and the database.

3.1 Hypothesis Valuation

Let $\mathcal{O}$ denote the set of observed scene features, $\mathcal{D}$ the database, and H a hypothesis. We define the valuation of H by $v(H) = p(H|\mathcal{O}, \mathcal{D})$. Using Bayes rule,

$$v(H) = p(H|\mathcal{O}, \mathcal{D}) = \frac{p(\mathcal{O}|H, \mathcal{D})p(H|\mathcal{D})}{p(\mathcal{O}|\mathcal{D})} \tag{1}$$

The desired output of the recognition algorithm is the hypothesis H maximizing this valuation. In particular,

$$H = \underset{H \in \mathcal{H}}{argmax} \left(\frac{p(\mathcal{O}|H, \mathcal{D})p(H|\mathcal{D})}{p(\mathcal{O}|\mathcal{D})} \right) = \underset{H \in \mathcal{H}}{argmax} \left(p(\mathcal{O}|H, \mathcal{D})p(H|\mathcal{D}) \right) \tag{2}$$

where $\mathcal{H}$ denotes the set of all hypotheses and we dropped the constant $p(\mathcal{O}|\mathcal{D})$.

In order to evaluate these probabilities, we expand a hypothesis into several components. The hypothesis states which objects are in the scene and where those objects are detected in the scene. Let m denote the number of object detections predicted by hypothesis H. Then, for $i = 1 \ldots m$, H specifies the model, $M_i \in \mathcal{D}$, of the i^{th} detected object, as well a set of parameters, Z_i, describing that model's pose in the scene.

In addition to stating the position of detected objects, hypothesis H attributes their appearance to features found in the scene. In particular, H breaks the set of scene features, $\mathcal{O}$, into $m+1$ disjoint sets, $\mathcal{O}_0 \ldots \mathcal{O}_m$, where $\mathcal{O}_0$ is the set of features attributed to the background and for $i = 1 \ldots m$, $\mathcal{O}_i$ is the set of features attributed to model M_i. To specify the exact pairing between scene features in $\mathcal{O}_i$ and model features in M_i, we introduce two auxiliary variables, $\mathbf{d}_i$ and $\mathbf{h}_i$. The binary vector $\mathbf{d}_i$ indicates which features of M_i are detected (value 1) and which features are missing (value 0). Vector $\mathbf{h}_i$ also contains an entry for each feature j of M_i. If j is detected ($\mathbf{d}_{ij} = 1$), then $\mathbf{h}_{ij}$ indicates the element of $\mathcal{O}_i$ to which j corresponds. In other words, $\mathbf{h}_i$ maps indices of detected model features to indices of their corresponding scene features.

For notational convenience, we define the single vector $\mathbf{h}$ to contain the entire correspondence map between scene features and model features (or background). $\mathbf{h}$ is simply the concatenation of all the $\mathbf{h}_i$'s. Also, n denotes the number of background features, or equivalently, the size of $\mathcal{O}_0$. Together, $\mathbf{h}$, n, $\{\mathbf{d}_1, \ldots, \mathbf{d}_m\}$, $\{Z_1, \ldots, Z_m\}$, and $\{M_1, \ldots, M_m\}$ completely specify a hypothesis. These variables contain all detection, pose, and feature correspondence information.

Using this decomposition, we now return to the computation of the valuation of a hypothesis. From equation (2) we can redefine the hypothesis valuation as

$$v'(H) = p(\mathcal{O}|H, \mathcal{D}) \cdot p(H|\mathcal{D}) \tag{3}$$

3.2 Pose and Appearance Density

The term $p(\mathcal{O}|H, \mathcal{D})$ characterizes the probability density in location, scale, orientation, and appearance for the features detected in the scene image. Conditioning on the pose of models present in hypothesis H, we can assume that features attributed by H to different model objects are mutually independent:

$$\begin{aligned} p(\mathcal{O}|H, \mathcal{D}) =& p(\mathcal{O}|\mathbf{h}, n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) \\ =& p(\mathcal{O}_0|n, \mathcal{D}) \cdot \prod_{i=1}^{m} p(\mathcal{O}_i|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D}) \end{aligned} \tag{4}$$

– $p(\mathcal{O}_0|n, \mathcal{D})$ is the probability that the n background detections would occur at the exact positions and with the exact appearances specified in $\mathcal{O}_0$. We assume each point in the (location, orientation, scale) space examined by the feature generator has an equal chance of producing a spurious detection. Assuming that clutter detections are independent from each other,

$$p(\mathcal{O}_0|n, \mathcal{D}) = \left[\frac{1}{A} \cdot \frac{1}{2\pi}\right]^n \cdot \prod_{\mathbf{x} \in \mathcal{O}_0} p_{\mathbf{bg}}(\mathbf{x}|\mathcal{D}) \tag{5}$$

where A is the number of pixels in the Gaussian pyramid used for feature detection, or equivalently, the size of the (location, scale) space, and there is a range of 2π in possible values for orientation. $p_{\mathbf{bg}}(\mathbf{x}|\mathcal{D})$ is the density describing the appearance of background features.

– $p(\mathcal{O}_i|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D})$ is the probability that the detections of the model features indicated by the hypothesis would occur with the exact pose and appearance specified in $\mathcal{O}_i$. Conditioning on model pose, we will assume independence between model features. This is a key assumption distinguishing our model from the constellation model [4,5,6]. Thus,

$$p(\mathcal{O}_i|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D}) = \prod_{\mathbf{x}\in\mathcal{O}_i} p_{\mathbf{pose}}(\mathbf{x}|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D}) \cdot p_{\mathbf{fg}}(\mathbf{x}|\mathbf{h}_i, \mathbf{d}_i, M_i, \mathcal{D}) \tag{6}$$

where $p_{\mathbf{pose}}$ and $p_{\mathbf{fg}}$ are the pose and appearance probabilities, respectively, for the foreground features.

The discussion on learning in section 5 describes the technique used for estimating probability densities $p_{\mathbf{bg}}$, $p_{\mathbf{pose}}$, and $p_{\mathbf{fg}}$.

3.3 Hypothesis Prior

The term $p(H|\mathcal{D})$ is the prior on the hypothesis. We expand this term as

$$\begin{aligned} p(H|\mathcal{D}) = &\ p(\mathbf{h}, n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}|\mathcal{D}) \\ = &\ p(\mathbf{h}|n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) \cdot p(n|\{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) \\ &\cdot \left[\prod_{i=1}^{m} p(\mathbf{d}_i|Z_i, M_i, \mathcal{D})\right] \cdot p(\{Z_i\}, \{M_i\}|\mathcal{D}) \end{aligned} \tag{7}$$

– $p(\mathbf{h}|n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D})$ is the probability of a specific set of feature assignments. As $\mathbf{h}$ is simply a vector of indices mapping model features to scene features, and we have no information on scene feature appearance or position at this stage, all mappings that predict n background features and are consistent with the detection vectors $\{\mathbf{d}_i\}$ are equally likely. Hence,

$$p(\mathbf{h}|n, \{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) = p(\mathbf{h}|n, \{\mathbf{d}_i\}) = \begin{cases} \left[\frac{N!}{(N-N_{fg})!}\right]^{-1} & \mathbf{h},\ n,\ \{\mathbf{d}_i\} \\ & \text{consistent} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where N is the total number of features in the scene image and $N_{fg} = N - n$ is the number of foreground features predicted by the hypothesis.

– $p(n|\{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D})$ is the probability of obtaining n background features. Background features are spurious responses to the feature detector that do not match with any known object. We assume a Poisson distribution for the number of background features [9]. Since scene images may have different sizes, the expected number of background detections is proportional to the area A examined by the feature detector. If λ denotes the mean number of background features per unit area, then

$$p(n|\{\mathbf{d}_i\}, \{Z_i\}, \{M_i\}, \mathcal{D}) = p_{Poisson}(n|\lambda, A) = e^{-\lambda A}\frac{(\lambda A)^n}{n!} \tag{9}$$

– $p(\mathbf{d}_i|Z_i, M_i, \mathcal{D})$ is the probability of detecting the indicated model features from model M_i. Let p_{ij} denote the probability that feature j of model M_i is detected in the scene. The probability that it is missing is $(1 - p_{ij})$. We break $p(\mathbf{d}_i|Z_i, M_i, \mathcal{D})$ into a term for detected features and a term for missing features to obtain

$$p(\mathbf{d}_i|Z_i, M_i, \mathcal{D}) = \prod_{\substack{j \text{ detected} \\ (\mathbf{d}_{ij}=1)}} p_{ij} \cdot \prod_{\substack{j \text{ missing} \\ (\mathbf{d}_{ij}=0)}} (1 - p_{ij}) \tag{10}$$

– $p(\{Z_i\}, \{M_i\}|\mathcal{D})$ is the prior on detecting objects $\{M_i\}$ in poses $\{Z_i\}$. We model this prior by a uniform density over frame transformations and combinations of model objects in the scene. Thus, this term is dropped in the implementation presented here.

4 Hypothesis Search

The recognition process consists of finding the hypothesis H that maximizes $v'(H)$. Unfortunately, due to the size of the hypothesis space $\mathcal{H}$, it is not possible to evaluate $v'(H)$ for each $H \in \mathcal{H}$. Early work by Grimson (e.g. [14]) showed the exponential growth of the search tree and the need for hypotheses pruning. Here, we use the A* search technique [10,11] to incrementally construct a reasonable hypothesis while only examining a small fraction of the hypothesis space.

In constructing incrementally a solution, we introduce the notion of a *partial hypothesis* to refer to a partial specification of a hypothesis. In particular, a partial hypothesis specifies a set of models $\{M_i\}$ and their corresponding poses $\{Z_i\}$ as well as a pairing between scene features and model features. Unpaired scene features are either marked as background or unassigned, whereas are either missing or unassigned. The partial hypothesis does not dictate how the unassigned scene or model features are to be treated. A *completion* of a partial hypothesis is a hypothesis that makes the same assignments as the partial hypothesis, but in which there are no unassigned scene features. A completion may introduce new models, make pairings between unassigned scene and model features, mark unassigned scene features as background, or mark unassigned model features as missing.

4.1 A*

We can organize the set of all partial hypotheses into a tree. The root of the tree is the partial hypothesis containing no models and in which all scene features are unassigned. The leaves of the tree are all complete hypotheses, (i.e. $\mathcal{H}$). Descending a branch of the tree corresponds to incrementally making decisions about feature assignments in order to further specify a partial hypothesis.

We prioritize the exploration of the tree by computing a valuation for each partial hypothesis. Partial hypotheses are entered into a priority queue according to this valuation. At each step of the search procedure, the highest valuation

partial hypothesis is dequeued and split into two new partial hypotheses. In one of these new hypotheses, a certain feature assignment is made. In the other new hypothesis, that feature assignment is expressly forbidden from occuring. This binary splitting ensures that a search of the hypothesis tree visits each partial hypothesis at most once.

4.2 Partial Hypothesis Valuation

To produce an effective search strategy, the valuation of a partial hypothesis should reflect the valuation of its best possible completion. If these two quantities were equal, the search would immediately descend the tree to the best complete hypothesis. However, it is impossible to compute the valuation of the best possible completion before actually finding this completion, which is the task of the search in the first place. Therefore, we will define the valuation of a partial hypothesis using a heuristic.

The heuristic we use can be thought of as the "optimistic worst-case scenario". It is the valuation of the partial hypothesis's completion in which all unassigned scene features are marked as background and all unassigned model features are dropped from the model. Unassigned model features are counted as neither detected nor missing. They do not enter into probability computations.

Note that this choice of heuristic is coherent with the expression for the valuation of a complete hypothesis. As the algorithm makes assignments in a partial hypothesis, its valuation approaches the valuations of its possible completions. Furthermore, this valuation is likely to serve as a decent guide for the search procedure. It is a measure of the minimum performance offered by a branch under the assumption that further assignments along that branch will do no harm.

4.3 Initialization

A list of potential database feature matches is created for each scene feature based on appearance. The empty partial hypothesis is split into two based on the best appearance match. One subbranch accepts this match, the other rejects it and forbids it.

4.4 Search Step

The partial hypothesis H with the highest valuation is dequeued. If H contains a model in which there are unassigned features, the algorithm picks one of these unassigned model features. A similar splitting to that in the initialization step is performed: one subbranch adds the match to the hypothesis, and the other forbids it as far as this hypothesis is concerned. In order to save computation time, we greedily follow only the branch that results in a better valuation. This is reasonable for rigid models in which the pose constraints should allow very few possibilities for a correct match in the scene.

If there are no unassigned model features in H, we pick the unassigned scene feature with the best appearance based match and split the hypothesis on this

assignment: one subbranch accepts the match and adds the corresponding model to the hypothesis, the other rejects the match, and adds no information regarding this model.

In both of the above cases, the resulting partial hypothesis or hypotheses are enqueued and the process is repeated.

4.5 Termination

The search process corresponding to one object terminates when no more unassigned features are available in this object. The scene features paired with this object are removed, and the search iterates with the remaining scene features. If all model objects have been considered without fully explaining the scene, the unassigned scene features are considered as background detections.

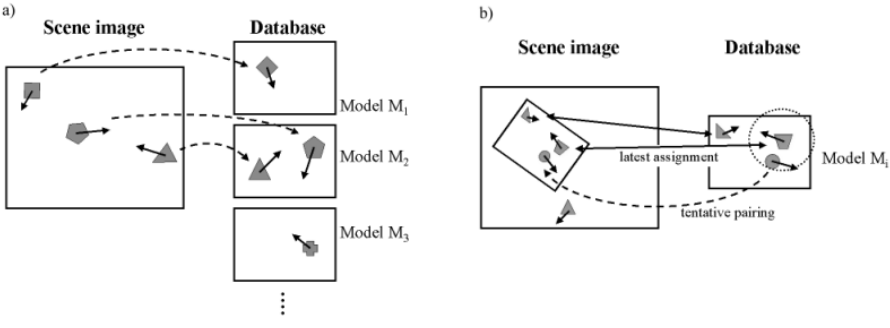

Fig. 1. Sketch of hypothesis build: a) Initialization: The best appearance match in the database is identified for each scene feature. Each such match is entered in the queue as a partial hypothesis. b) Search for a new match in the partial hypothesis which has highest valuation: we look for an unassigned feature in the same model image M_i. This feature is mapped to its best appearance match in the scene, if this new pairing is coherent with the pose predicted by the hypothesis - otherwise, the match is rejected. The pose is then updated based on the new match.

5 Learning

Several components of the probabilistic framework given above must be inferred from training examples. Since our system requires only a single training image per object, we cannot estimate separate appearance and pose densities for each feature in an object model. We therefore utilize the entire feature database in estimating global probability densities which can be applied to all features. Note that only training images are used here, not the test set.

5.1 Background Features

To estimate the background appearance density, we assume that a typical background feature looks like a feature found in the database. The background, like the database, is composed of objects. It just happens that these objects are not in the database. A probability density for the appearance of features in the database describes the appearance of background detections in a scene. We model this density with a full covariance gaussian density. Letting $\mu_{\mathbf{bg}}$ and $\Sigma_{\mathbf{bg}}$ denote the mean and covariance of the database feature appearance vectors,

$$p_{\mathbf{bg}}(\mathbf{x}|\mathcal{D}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_{\mathbf{bg}}|} e^{-\frac{1}{2}(\mathbf{x_{app}}-\mu_{\mathbf{bg}})^T \Sigma_{\mathbf{bg}}^{-1}(\mathbf{x_{app}}-\mu_{\mathbf{bg}})} \tag{11}$$

where $\mathbf{x_{app}}$ is the appearance vector of feature $\mathbf{x}$ and d the dimension of appearance vectors.

A typical model generates 500 to 1000 features, resulting for a database with 100 objects in a total of 50,000 to 100,000 training examples for the background appearance density. As our experiments used 128-dimensional appearance vectors, this was a sufficient number of examples for estimating the gaussian density.

The mean number of background detections per unit area, λ, is programmer specified in our current implementation. When running Lowe's detection method on our training and test sets, 80% of the detections were assigned to the background, therefore we chose this same fraction for λ. This parameter has only weakly effects on the total probability as the terms for pose and appearance dominate.

5.2 Foreground Features

The foreground appearance density must describe how closely a scene feature resembles the model feature to which it is matched. This density is difficult to estimate as in principle, it involves establishing hundreds of thousands of ground truth matches by hand. A possible shortcut is looking at statistics coming from planar scenes seen from different viewpoints [15], or synthetic deformations of an image [3].

Here we followed a different approach: we approximate a good match for a feature by its closest match in appearance in the database. The difference in appearance between correctly matched foreground features is modeled with a gaussian density with full covariance matrix, and the covariance matrix $\Sigma_{\mathbf{fg}}$ is estimated from the difference in appearance between database features paired in such a manner. This yields

$$p_{\mathbf{fg}}(\mathbf{x}|\mathbf{h}_i, \mathbf{d}_i, M_i, \mathcal{D}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_{\mathbf{fg}}|} e^{-\frac{1}{2}(\mathbf{x_{app}}-\mathbf{y_{app}})^T \Sigma_{\mathbf{fg}}^{-1}(\mathbf{x_{app}}-\mathbf{y_{app}})} \tag{12}$$

where $\mathbf{y} = \mathbf{h}_i^{-1}(\mathbf{x})$ is the model feature paired with scene feature $\mathbf{x}$.

Unlike background feature pose which are modeled with a uniform distribution in equation (5), foreground features are expected to lie in a pose consistent

with that of their corresponding model. In particular, model pose Z_i predicts a scene location, orientation, and scale for each feature of M_i. If the hypothesis matches scene feature $\mathbf{x}$ to model feature $\mathbf{y}$, and Z_i maps $\mathbf{y}$ to $\mathbf{z}$, we write

$$p_{\mathbf{pose}}(\mathbf{x}|\mathbf{h}_i, \mathbf{d}_i, Z_i, M_i, \mathcal{D}) = G_{\mathbf{loc}}(\mathbf{x}|\mathbf{z}) \cdot G_\theta(\mathbf{x}|\mathbf{z}) \cdot G_{\mathbf{s}}(\mathbf{x}|\mathbf{z}) \cdot \tag{13}$$

where $G_{\mathbf{loc}}$, G_θ, and $G_{\mathbf{s}}$ are Gaussian densities for location, orientation, and log scale, respectively, with means given by the pose of $\mathbf{z}$. The covariance parameters of these densities are currently specified by hand, with values of 20 pixels for location, half an octave for log-scale and 60 degrees for orientation (orientation was quite unreliable).

We determine the model pose Z_i by solving for the similarity transform that minimizes, in the least-squares sense, the distance between observed and predicted locations of foreground model features. Z_i is updated whenever a previously unassigned feature of M_i is matched.

The probability p_{ij} of detecting individual features is set to the same value across features and models. A reasonable choice is the fraction of features that are typically needed to produce a reliable pose estimate. This value was obtained by running Lowe's detection method on our training and test sets: in average 20% of a model features were found in a test image containing this model. This value of 20% was used for p_{ij}.

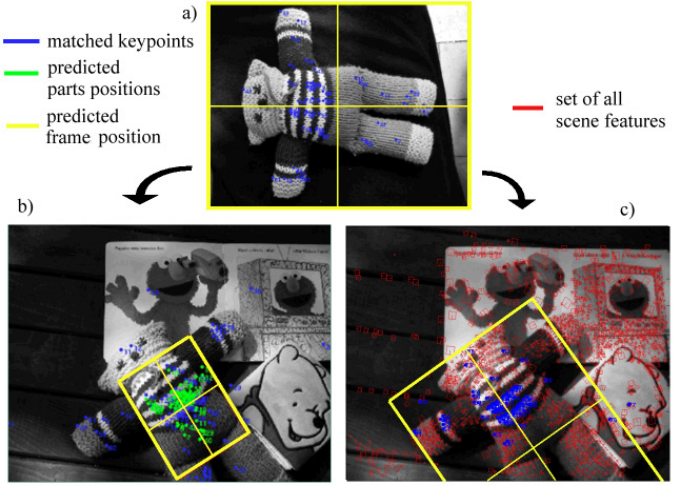

Fig. 2. Example of result for a textured object included in a complex scene (only one detection shown here). According to this hypothesis, the box displayed in the model image is transformed into the box shown in the scene image. a) Initial object b) Result of Lowe's algorithm. Since the stuffed bear is a textured object, detection of similar features can occur in many locations, leading to incorrect pairings. As a result, the frame transformation, estimated only from the features positions, is inaccurate. c) Result of the probabilistic search.

6 Experimental Results

In the absence of "standard" training and test sets of images containing both objects and clutter, we compare the performance of our probabilistic search method to that of Lowe's algorithm on a training set consisting of 100 images of toys and common kitchen items, with a single image per object. The test set contained images of single objects, as well as complicated scenes that include several objects, ranging from 1 to 9 objects. It included 80 test images, with a total of 254 objects to be detected (each object was considered as one detection). Some test images didn't contain any learned object. In that case all feature detections are expected to be assigned to the background. We used a resolution of 480×320 for training images and test images of single objects, and 800×533 for complex scenes. All images were taken in a kitchen, with an off-the-shelf digital camera, and no precautions were taken with respect to lighting conditions, viewpoint angle and background. In particular, the lighting conditions varied significantly between training and test images, and viewing angles varied between 0 and 180 degrees (picture of the back of an object taken as test while the corresponding model was a picture of the front of the object). No image was manually segmented, and the proportion of features generated by an object in a model or test image, ranged from 10%to 80% (80% for a single object occupying most of the image). The database is available online from http://www.vision.caltech.edu/html-files/archive.html.

Our algorithm achieved a detection performance similar to Lowe's system, with a detection rate of 85%. Figure 3 shows ROC curves for both methods. The threshold used is the accuracy of the best hypothesis at the end of the search. Since our method verifies the coherence of each match by scoring partial hypothesis, our false alarm rate was lower than that of Lowe's method.

In order to measure the accuracy of the pose transformations estimated by each method, the training and test images were manually marked with ground truth information. An ellipse was fitted, and a canonical orientation was chosen, for each object. We measured the accuracy of the transformation with the distance in pixels, between the predicted positions of the ellipses in a scene, and the ground truth previously recorded. The error was averaged across points regularly spaced on the ellipse and across test images. We obtained a mean error of 45 pixels for our method, and 56 pixels for Lowe's algorithm.

Our approach requires to examine and evaluate a number of partial and complete hypotheses that is much higher than with Lowe's method. As a result, the probabilistic algorithm is the slower of the two methods. Our unoptimized code for Lowe's method takes in average 2 seconds on a Pentium 4 running at 2.4GHz to identify objects in a 800×533 image, while our probabilistic algorithm requires on average 10 seconds for the same image.

In practice, the A* search achieves only little pruning, typically 10-20% of the branches are eliminated. Therefore, the valuation heuristic was coupled with a stopping criterion (depth-first completion of the partial hypothesis that performs best after 4000 iterations). The main computational benefit of the A* method in this paper, is to introduce a framework for evaluating partial hypotheses in a

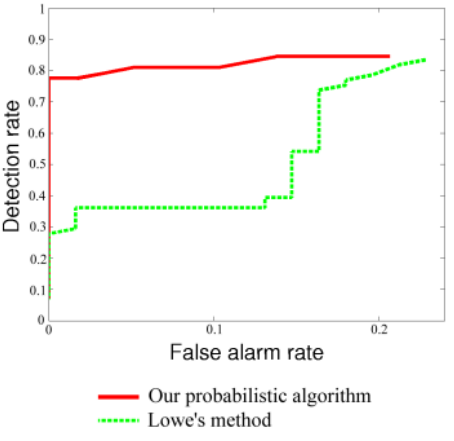

Fig. 3. ROC based on the accuracy of the pose estimated by the best hypothesis. It measures how much the hypothesis' prediction of the object position, differs from the ground truth. This quantity can be measured for both recognition systems.

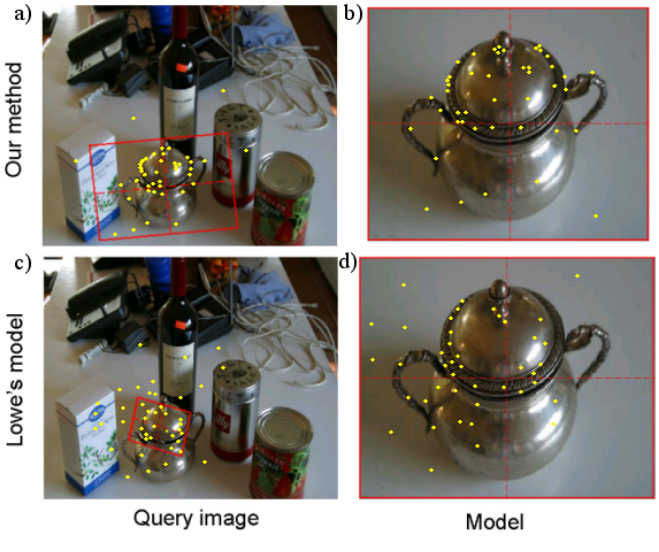

Fig. 4. Other example of recognition in a complex environment. a) and b) present one match obtained by our probabilistic search, c) and d) are the best result from Lowe's voting approach. Since Lowe's method does not evaluate geometric and appearance quality of hypotheses, numerous incorrect correspondences are accepted. As a result, the estimated frame position is inaccurate. The probabilistic search accepts only matches that are geometrically coherent, and leads to accurate pose parameters.

Fig. 5. Samples from our training and test sets. The red boxes show locations where models were identified

way that is coherent with the valuation of complete hypotheses, and a ranking of hypotheses that leads to efficient search.

7 Discussion and Conclusion

We have presented a new probabilistic model and efficient search strategy for recognizing multiple objects in images. Our model provides a unified view of two previous lines of research: it may be thought as a probabilistic interpretation of David Lowe's work [7,8] or, conversely, as a special case of the constellation model [4] where many of the parameters are pooled amongst models, rather than learned individually.

Our experiments indicate that the system we propose achieves the same detection rate as Lowe's algorithm with significantly lower false alarm rates. The localization error of detected objects is also smaller. The price to be paid is a slower processing time, although this may not be a significant issue since our code is currently not optimized for speed. The front-end of both systems was identical (feature detection, feature representation, feature matching) and therefore all measurable differences are to be ascribed to the probabilistic model and to the matching algorithm.

It is clear that the heuristic we chose for ranking partial hypotheses is susceptible of improvement. In choosing it we followed intuition rather than a principled

approach. This is obviously an area for further investigation. Developing better techniques for estimating the probability density function of appearance and pose error of both foreground and background features is another issue deserving of further attention.

Acknowledgments. The authors thank Prof. D.Lowe for kindly providing the code of his feature detector and for useful advice. They also acknowledge funding from the NSF Engineering Research Center on Neuromorphic Systems Engineering at Caltech.

References

1. Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. Arch. Rat. Mech. Anal. **78** (1982) 315–333
2. B. Schiele: Object Recognition Using Multidimensional Receptive Field Histograms PhD thesis, I.N.P. de Grenoble, 1997.
3. C. Schmid and R. Mohr: Local greyvalue invariants for image retrieval IEEE Trans. on Patt. Anal. Mach. Int., 19(5):530–535, 1997.
4. M.C. Burl and P. Perona Recognition of planar object classes IEEE Comp. on Comp. Vision and Patt. Recog., CVPR 96, San Francisco, CA, June 1996.
5. R. Fergus, P. Perona, A. Zisserman: Object class recognition by unsupervised scale-invariant learning. IEEE Conf. on Comp. Vision and Patt. Recog., 2003
6. M. Weber, M. Welling and P. Perona: Unsupervised learning of models for recognition. Proo. 6th Europ. Conf. Comp. Vis., ECCV2000, 2000.
7. D.G. Lowe: Object recognition from local scale-invariant features. Proc. Int. Conf. Comp. Vision, Corfu, Greece, pp. 1150–1157, 1999.
8. D.G. Lowe: Distinctive image features from scale-invariant keypoints accepted paper, Int. J. of Comp. Vision, 2004.
9. M. Weber: Unsupervised Learning of Models for Object Recognition, Ph.D thesis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA, 2000.
10. J.M. Coughlan and A.L.Yuille: Bayesian A* tree search with expected O(N) node expansions: applications to road tracking. Draft submitted to Neural Computation, Dec. 2002.
11. J. Pearl: Heuristics, Addison-Wesley, 1984.
12. J.S. Beis and D.G. Lowe: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. Proc. IEEE Conf. on Comp. Vision and Patt. Recog., Puerto Rico, pp. 1000-1006, 1997.
13. L. Fei-Fei, R. Fergus and P. Perona: A bayesian approach to unsupervised one-shot learning of object categories, Proc. Int. Conf. on Comp. Vision, Nice, France, 2003.
14. W.E.L. Grimson: Model-based recognition and localization from sparse range or tactile data, AI Memo 738, Massachusetts Institute of Technology, Aug 1983.
15. K. Mikolajczyk and C. Schmid, A performance evaluation of local descriptors, Proc. IEEE Conf. on Comp. Vision and Patt. Recog., Madison, Wisconsin, p. 257.