

Unsupervised Learning of Probabilistic Object Models (POMs) for Object Classification, Segmentation and Recognition using Knowledge Propagation

Yuanhao Chen¹, Long (Leo) Zhu², Alan Yuille^{2,3}, Hongjiang Zhang⁴

¹University of Science and Technology of China, Hefei, Anhui 230026 P.R.China

yhchen4@ustc.edu

²Department of Statistics, ³Psychology and Computer Science

University of California, Los Angeles, CA 90095

{lzh, yuille}@stat.ucla.edu

⁴Microsoft Advanced Technology Center, hjzhang@microsoft.com

Abstract

We present a method to learn probabilistic object models (POM's) with minimal supervision which can exploit different visual cues and perform tasks such as classification, segmentation, and recognition. We formulate this as a structure induction and learning task and our strategy is to learn and combine basic POM's that make use of complementary image cues. We describe a novel structure induction procedure which uses *knowledge propagation* to enable POM's to provide information to other POM's and "teach them" (which greatly reduces the amount of supervision required for training and speeds up the inference). In particular, we learn a POM-IP defined on Interest Points using weak supervision [1], [2] and use this to train a POM-mask, defined on regional features, which yields a combined POM which performs segmentation/localization. This combined model can be used to train POM-edgelets, defined on edgelets, which gives a full POM with improved performance on classification. We give detailed experimental analysis on large datasets which show that the full POM is invariant to scale and rotation of the object (for learning and inference) and performs inference rapidly. In addition, we show that we can apply POM's to learn objects classes (i.e. when there are several objects and the identity of the object in each image is unknown). We emphasize that these models can match between different objects from the same category and hence enable object recognition.

I. INTRODUCTION

Recent work on object recognition has tended to represent objects in terms of spatial configurations of features at a small number of interest points [3], [4], [5], [6]. Such models are computationally efficient, for both learning and inference, and can be very effective for tasks such as classification. But they have two major disadvantages: (i) the sparseness of their representations restricts the set of visual tasks they can perform, and (ii) these models only exploit a small set of image cues. Sparseness is suboptimal for tasks such as segmentation which instead require different representations and algorithms. This has led to an artificial distinction in the vision literature where detection/classification and segmentation are treated as different problems being addressed with different object representations, different image cues, and different learning and inference algorithms. One part of the literature concentrates on detection/classification – e.g. [3], [4], [5], [6], [1], [2] – uses sparse generative models, and learns them using comparatively little human supervision (e.g. the training images are known to include an object from a specific class, but the precise localization/segmentation of the object is unknown). By contrast, the segmentation literature – e.g. [7], [8], [9] – uses dense representations but typically requires that the precise localization/segmentation of the objects are given in the training images. But until recently – e.g. [10], [11], [12] – there have been few attempts to combine segmentation and classification or to make use of multiple visual cues.

Pattern theory gives a theoretical framework to address these issues – represent objects by state variables W , specify a generative model $P(I|W)P(W)$ for obtaining the observed image I , and an inference algorithm to estimate the most probable object state $W^* = \arg \max_W P(W|I)$. The estimated state W^* determines the identity, pose, configuration, and other properties of the object (i.e. is sufficient to perform all object tasks). This approach makes use of all cues available in the image and is formally optimal in the sense of Bayes decision theory. Unfortunately it currently suffers from many practical disadvantages when faced with the complexity of natural images. It is unclear how to specify the object representations, how to learn generative models from training data, and how to perform inference effectively (i.e. estimate W^*).

The goal of this paper is to describe a strategy for learning probabilistic object models (POMs) in an incremental manner with minimal supervision. The idea is to begin by learning a simple model that only has a sparse representation of the object and hence only explains a small part

of the data and performs a restricted set of tasks. Once learnt, this model can process the image to provide information that can be used to learn POMs with increasingly richer representations, which exploit more image cues, and which can perform more visual tasks. We refer to this strategy as knowledge propagation (KP) since it uses knowledge provided by the simpler models to help train the more complex models (e.g. the simple models act as teachers). Knowledge propagation is also used after the POMs have been learnt to enable rapid inference to be done (i.e. estimate W^*). To assist KP, we use techniques for growing simple models using proposals obtained by clustering [1], [2]. A short version of this work was presented in [13].

We formulate our approach in terms of probabilistic inference and machine learning. From this perspective, learning POMs is a structure induction problem where the goal is to learn the structure of the probability model describing the objects as well as the parameters of their distributions. Structure induction is a difficult and topical problem and differs from more traditional learning where the structure of the model is assumed known and only the parameters need to be estimated. Knowledge propagation is a method for doing structure learning that builds on our previous work on structure induction [1], [2] which is summarized in section (IV).

For concreteness, we now briefly step through the process of structure learning by KP as it occurs in this paper – see figure (1). Firstly, we learn a POM defined on interest points (IP's), POM-IP, using the techniques described in [1], [2]. We start with a POM-IP because the sparseness of the interest points and their different appearances makes it easy to learn it with minimal supervision. This POM-IP can be learnt from a set of images which each contain one of a small set of objects with variable pose (position, scale, and rotation) and variable background. *This is the only information provided to the system – the rest of the processing is completely automatic.* The POM-IP is a mixture model where each component represents a different aspect of the object (the number of components is learnt automatically). This POM-IP is able to detect and classify objects, to detect their aspect, deal automatically with scaling and rotation changes, and give very crude estimates for segmentation. We next extend this model by incorporating different cues to enable accurate segmentation and to improve classification. We use the POM-IP to train a POM-mask and use regional image cues to perform segmentation with a min-cut/max-flow algorithm [14]. Intuitively, we start by using a version of grab-cut [15], [16], [17] where POM-IP substitutes for human interaction to provide the initial estimate of the segmentation (as motion cues do in ObjCut [18]). This, by itself, yields a fairly poor segmentations of the objects.

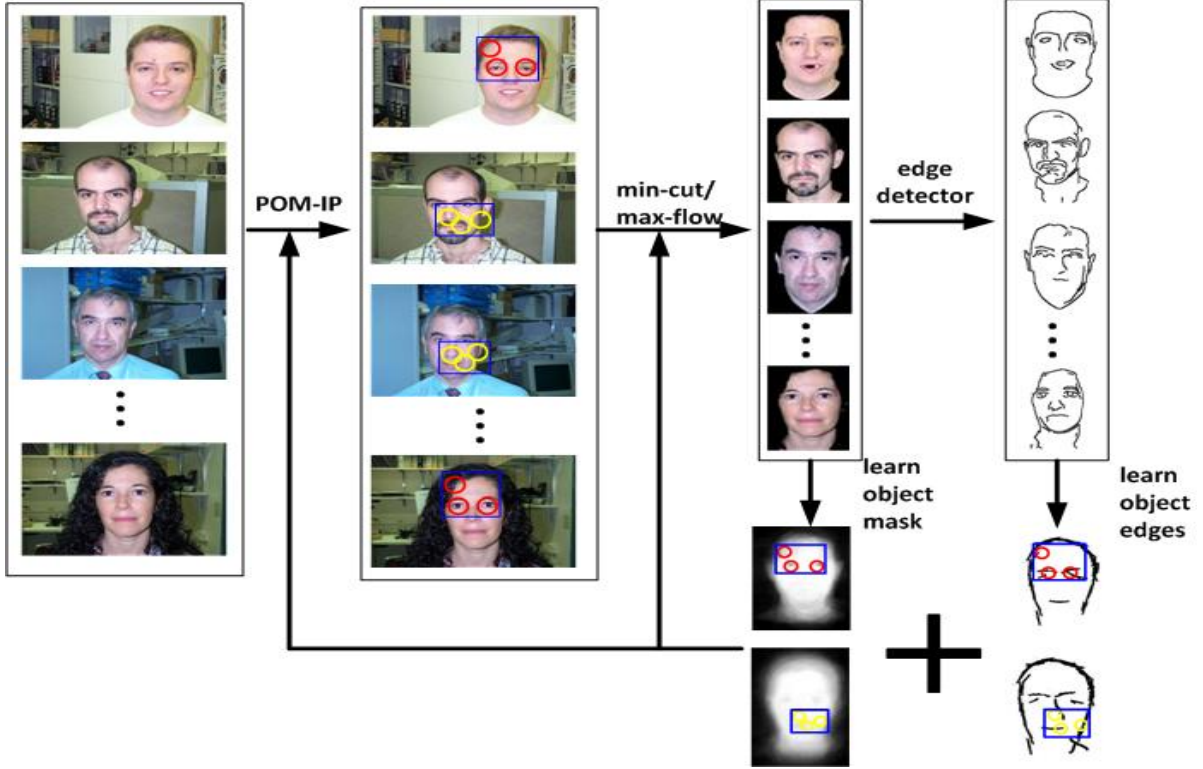


Fig. 1. The flow chart of knowledge propagation. POM-IP is learnt and then trains POM-mask (using max-flow/min-cut) which includes learning a probabilistic object mask. Then POM-IP and POM-mask help train POM-edgelets by using the object mask to provide context for the six POM-edgelets. Knowledge propagation is also used for inference (after learning) with similar flow from POM-IP to POM-mask to POM-edgelets.

But this segmentation can be improved by using the training data to learn priors for the masks (different priors for each aspect). This yields an integrated model which combines POM-IP and POM-mask and which is capable of performing classification and segmentation/localization. Because this model can estimate the shape of the object it can provide sufficient context to train POM-edgelets which can localize subparts of the object and hence improve classification (the context provides strong localization for the POM-edgelets which makes it easy to learn them and perform inference with them). After the models have been learnt, KP is also used so that POM-IP provides estimates of pose (scale, position, and orientation) which helps provide initial conditions for POM-mask which, in turn, provides initial conditions for the POM-edgelets. We stress that learning and performing inference on POM-mask and POM-edgelets is very challenging without the initial conditions provided by the earlier models. The full model couples the POM-IP, POM-

mask, POM-edgelets together (as a regular, though complicated, graphical model) and performs inference on this model.

Our experiments demonstrate the success of our approach. Firstly, we show that the full POM – coupling POM-IP, POM-mask, and POM-edgelet – performs better for classification than POM-IP alone. Secondly, the segmentation obtained by coupling POM-IP with POM-mask is much better than performing segmentation with POM-IP only. In addition, we show that the performance of the system is invariant to scale, rotation, and position transformations of the objects and can be performed for hybrid object classes. We give comparisons to other methods [3], [11], [12]. Finally we show promising results for performing recognition by the POM-IP (i.e. distinguishing between different objects in the same category).

The structure of this paper is as follows. First we describe the basic ideas of knowledge propagation in section (II). Next we give details specifications of the image cues and the representations used in this paper in section (III). Then we specify the details of the POMs and KP in section (IV,V,VI). Finally we report the results in section (VII).

II. LEARNING BY KNOWLEDGE PROPAGATION

We now describe the basic idea of learning by knowledge propagation. Suppose our goal is to learn a generative model to explain some complicated data. It may be too hard to attempt a model that can explain all the data in one attempt. An alternative strategy is to build the model incrementally by first attempting to model those parts of the data which are easiest to model. This will provide context which makes it easier to learn models for the rest of the data.

To put this work in context, we recall the basic formulation of unsupervised learning and inference tasks. Suppose we have data $\{d^\mu\}$ that is a set of sample from a generative model $P(d|h, \omega)P(h|\Omega)$ with hidden states h and model parameters ω, Ω . (For example, d can correspond to the interest points in the image, h is the positions of nodes of the model, ω are parameters of the distributions specifying the appearances and spatial relations between the interest points, and Ω are the parameters on the structure of the model). The two tasks are: (i) to learn the model – i.e. determine ω, Ω by MAP estimation $\omega^*, \Omega^* = \arg \max P(\omega, \Omega | \{d^\mu\})$ using training data $\{d^\mu\}$ (which also includes learning the structure of the model), and (ii) to perform inference from d to determine $h(d)$ by MAP $h^*(d) = \arg \max P(h|d, \omega, \Omega)$. But, as described in the introduction, there may not be efficient algorithms to achieve these tasks.

The basic idea of knowledge propagation can be illustrated as follows. Assume that there is a natural decomposition of the data into $d = (d_1, d_2)$ and hidden states $h = (h_1, h_2)$ so that we can express the distributions as $P(d_1|h_1, \omega_1)P(d_2|h_2, \omega_2)P(h_1|\Omega_1)P(h_2|h_1, \Omega_2)$. This is essentially two models for generating different parts of the data which are linked by the coupling term $P(h_2|h_1, \Omega_2)$. Knowledge propagation proceeds by first decoupling the models and learning the model by setting $\hat{\omega}_1, \hat{\Omega}_1 = \arg \max \prod_{\mu} \sum_{h_1} P(d_1^{\mu}|h_1, \omega_1)P(h_1|\Omega_1)$ from the data $\{d_1^{\mu}\}$ (i.e. ignoring the $\{d_2^{\mu}\}$). Once this model has been learnt, we can use it to make inference of the hidden state $h_1^*(d)$. This provides information which can be used to learn the second part of the model – i.e. to estimate $\omega_2^*, \Omega_2^* = \arg \max \prod_{\mu} \sum_{h_2} P(d_2^{\mu}|h_2, \omega_2)P(h_2|h_1^*(d), \Omega_2)$. These estimates are only approximate, since they make approximations about the coupling between the two models. But these estimates can be improved by treating them as initial conditions for alternating iterative algorithms (e.g. fix (ω_1, Ω_1) make best estimate of (ω_2, Ω_2) , then fix (ω_2, Ω_2) and make best estimate of (ω_1, Ω_1) , and so on). This results in a coupled Bayes net for generating the data. Knowledge propagation can also be used in inference. We use the first model to estimate $h_1^*(d) = \arg \max P(d_1|h_1)P(h_1)$ and then estimate $h_2^*(d) = \arg \max P(d_2|h_2)P(h_2|h_1^*(d))$. Once again, we can improve these estimates by using them as initial conditions for an alternating iterative algorithm (i.e. fix h_1 and estimate h_2 , then fix h_2 and estimate h_1). It is straightforward to extend this approach to other data d_3, d_4, \dots and hidden states h_3, h_4, \dots .

In this paper, we will let $P(d_1|h_1)P(h_1)$ be a probabilistic object model POM-IP which uses interest points (i.e. d_1 describes the appearance of interest points and ignores the rest of the image). It has been shown – [1], [2] see section (IV) – that models of this type can be learnt in an unsupervised manner even if the pose (position, orientation, scale) and aspect (mixture component) of the object are unknown. Moreover, there are rapid inference algorithms to estimate $h_1^* = \arg \max P(d_1|h_1)P(h_1)$ [1], [2], see section (IV). Then $P(d_2|h_2)P(h_2|h_1)$ correspond to a probabilistic object model POM-mask which represents the object by a mask with models for the image appearance inside and outside the mask. Once the POM-IP has been learnt (i.e. we have estimated ω_1, Ω_1) then we can estimate $h_1^*(d)$ yielding the pose (position, orientation, scale), aspect, and a very crude estimate of the silhouette of the object (from the bounding box of the interest points). This information provided by POM-IP makes it practical to learn the POM-mask efficiently (including learning a probability distribution for the shape of each aspect of the object) and also helps with estimating $h_2^*(d)$ once POM-mask has been learnt. Note that it

is very difficult to learn a POM-mask by itself for this type of data because of the variability of pose, aspect, and even of the shape itself (essentially POM-mask provides the alignment). Next we proceed by using POM-IP and POM-mask to train POM-edgelets.

III. THE IMAGE REPRESENTATION

This section describes the different image features that we use: (i) interest points, (ii) edgelets, and (iii) regional features. These will be used to define POM-IP, POM-edgelets, and POM-mask.

The *edgelet and interest point image features* of an image I are represented by a set of triples $d_1(I) = \{(z_i, \theta_i, A_i)\}$, where z_i is the location of the feature in the image, θ_i is the orientation of the feature, and A_i is an appearance vector. For edgelets, the appearance vector is not used. The edgelets are extracted by applying the Canny edge detector and estimating the orientation. The interest point features are those reported in [1], [2] which were designed to be relatively independent of scale and photometric properties. The Kadir-Brady procedure [19] is used to detect interest regions. These are described by the SIFT descriptor [20]. Principal Component Analysis (PCA) is used to reduce the description to fifteen dimensions to give the appearance A together with the orientation θ .

The *oriented triplets* were designed [1], [2] to give geometric properties, the *invariant triplet vector* (ITV), which are independent of scale and orientation. (Previous authors have used non-oriented triplets [21], [22]). The (ITV) $\vec{l}(z_i, \theta_i, z_j, \theta_j, z_k, \theta_k)$ is a function of the geometry of three features points $(z_i, \theta_i, z_j, \theta_j, z_k, \theta_k)$ which is invariant to scale and rotation.

The *regional image features* $d_2(I)$ are computed by applying a filter $\rho(\cdot)$ to the image I yielding a set of responses $d_2(I) = \{\rho_z(I) : z \in R, \text{ where } R \text{ is the image domain. The domain } R \text{ is split into pixels within the object denoted by } L_z = 1 \text{ and pixels outside the object with } L_z = 0 \text{ (the variable } \{L_z\} \text{ specifies the location and segmentation of the object, see section (V)). If } \{L_z\} \text{ is specified, we can compute the histograms of the image statistics inside the object } f_O(\cdot, L) \text{ and in the background } f_B(\cdot, L):$

$$f_O(\alpha, L) = \frac{1}{|R_O|} \sum_{z \in R} \delta_{L_z, 1} \delta_{\rho_z(I), \alpha}, \quad (1)$$

$$f_B(\alpha, L) = \frac{1}{|R_B|} \sum_{z \in R_O} \delta_{L_z, 0} \delta_{\rho_z(I), \alpha}, \quad (2)$$

where $|R_O| = \sum_{z \in R} \delta_{L_z, 1}$, $|R_B| = \sum_{z \in R} \delta_{L_z, O}$, δ is the Kronecker delta function, and α indicates the histogram bin. In this paper, $\rho_z(I)$ is either the colour or grey-scale image intensities. But other choices, including local texture filters are also suitable.

The *edges* $d_3(I)$ are obtained by applying a Canny edge detector and estimating their orientation. Hence $d_3(I) = \{(z_j, \theta_j)\}$ where j indexes the edges, z_j is position, and θ_j is orientation.

IV. POM-IP AND STRUCTURE INDUCTION

The full POM is built by combining a POM-IP with a POM-mask and POM-edgelets, see figure (2). In this section we introduce the POM-IP. The terminology for the hidden states of the full POM is shown in table (I).

The POM-IP is defined on sparse interest points and is almost identical to the probabilistic grammar Markov model (PGMM) described in [1], [2], see figure (2). The only difference is that we use an explicit pose variable G which is used to relate the different POMs and provides a key mechanism for knowledge propagation (G appeared in [2] but was integrated out in equation (9)). The input to the POM-IP is $d_1(I) = \{(z_i, \theta_i, A_i) : i = 1, \dots, N\}$ and is the position, orientation, and appearance of the interest points detected in the image I (see previous section). We specify the hidden states of the POM-IP by $h_1 = \{s, V, G\}$ where s, V, G are defined as follows. POM-IP is a mixture model where the *aspect* s labels the mixture component. Each aspect consists of a set of attributed nodes $a = 1, \dots, N_s$ organized into triplet cliques, see figure (2). The correspondence variable $V = \{i(a)\}$ specifies the assignment between nodes (labeled a) of the aspect and the interest points (labeled i) (or edgelets) in the image. G is the pose of the POM-IP and can be expressed as $G = (z_c, \theta_c, S_c)$ where z_c, θ_c, S_c are the center, rotation, and scale of the POM-IP. The parameters ω_1, Ω_1 of POM-IP specify the probability distributions $P(d_1(I)|h_1, \omega_1)P(h_1|\Omega_1)$ and will be specified in section (IV-A).

The *POM-IP distribution* is specified by $P(d_1(I)|V, G, s)P(V|s)P(s)P(G)$. The distribution $P(d_1(I)|V, G, s)$ specifies the probability of generating the interest points in the image and includes a probability distribution for the appearances A of the interest points and for geometric deformations of their relative positions and orientations (z, θ) . $P(V|s), P(s), P(G)$ are the prior probabilities for the correspondences V , the aspects s , and the pose G .

There are three main tasks for POM-IP: (I) structure induction, (II) parameter learning to estimate (Ω, ω) , and (III) inference to estimate (s, V, G) (from a single image).

Notation	Meaning
$\{(z, \theta, A) : i = 1, \dots, N\}$	the interest points in the image
z_i	the location of the feature
θ_i	the orientation of the feature
A_i	the appearance vector of the interest point feature
s	the aspect of the object
$a = 1, \dots, N_s$	the set of attributed nodes of the aspect s
$V = \{i(a)\}$	the correspondence variable between node a and the interest point i
G	the pose (position, orientation, and scale) of the object
$q = (q_O, q_B)$	the set of distribution on the image
q_O	the set of distribution which statistics inside the object
q_B	the set of distribution which statistics outside the object
I	the intensity image
L	a binary label field of the object

TABLE I

THE TERMINOLOGY USED TO DESCRIBE THE HIDDEN STATES h OF THE POMs

Inference requires estimating s, V, G from input $d_1(I)$ with the model parameters (Ω, ω) fixed. This requires solving

$$\begin{aligned}
 (s^*, V^* G^*) &= \arg \max_{s, V, G} P(s, V, G | d_1(I), \omega, \Omega) \\
 &= \arg \max_{s, V, G} P(d_1(I), \omega, \Omega, s, V, G)
 \end{aligned} \tag{3}$$

Parameter learning occurs when the structure of the model is known but we have to estimate the parameters of the model. Formally we specify a set W of parameters (ω, Ω) which we estimate by MAP. Hence we estimate

$$(\omega^*, \Omega^*) = \arg \max_{\omega, \Omega} P(\omega, \Omega | \{d_1(I_\mu)\}) \propto P(\{d_1(I_\mu)\} | \omega, \Omega) P(\omega, \Omega) \tag{4}$$

Structure Learning involves learning the model structure. Our strategy is to grow the structure of the PGMM by adding new aspect nodes, or by adding new cliques to existing aspect nodes. For each new structure, we evaluate the fit to the data by computing the *score*:

$$\text{score} = \max_{\omega, \Omega} P(\omega, \Omega) \prod_{\mu} P(d_1(I_\mu) | \omega, \Omega) \tag{5}$$

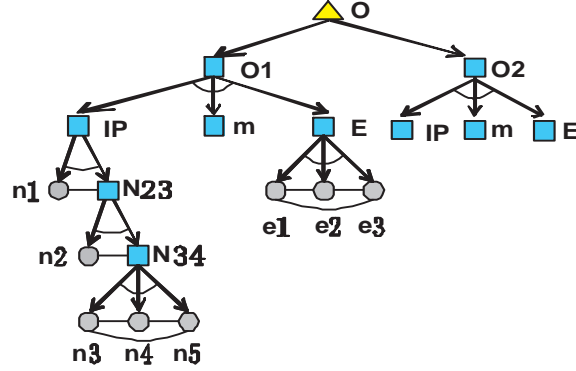


Fig. 2. The full POM is a graphical structure which couples three types of models: POM-IP (IP), POM-mask (m), and POM-edgelets (E). It represents a mixture of distributions – where the mixture components correspond to different objects (O1,O2) or different aspects of the same object. For a fixed aspect, the POM-IP consists of an ordered set of nodes – e.g. $n1, n2, n3, n4, n5$ – with cliques $(n1, n2, n3), (n2, n3, n4), (n3, n4, n5)$. A POM-edgelet (E) also consists of ordered nodes $e1, e2, e3$ and cliques $(e1, e2, e3)$.

The full description of these tasks is described in [2]. Some approximations are required (e.g. maximizing over G instead of summing it out) in order to make these computations tractable. For completeness, we sketch the main details in section (IV).

A. Details of the POM-IP model

The full POM-IP is a mixture distribution where the mixture components are labeled by s (the number of components is learnt automatically). For each aspect s , the POM-IP consists of an ordered set of nodes $L_O(s) = \{1, 2, \dots, n_s\}$ which define a set of triplet cliques $C(s) = \{(1, 2, 3), (2, 3, 4), \dots, (n_s - 2, n_s - 1, n_s)\}$, see figure (2). Each node has position and orientation – specified by (z_a, θ_a) – and a probability distribution on the appearance. There is a correspondence variable $V = \{i(a)\}$ which specifies the matching between node a of the POM-IP and an interest point (z_i, θ_i, A_i) observed in the image. We require that each POM-IP node has at most one match but can be unmatched (e.g. because of occlusion or failure of the interest point detector).

The model parameters ω_1, Ω_1 are specified as follows. For each node $a \in L_O(s)$ there is an appearance model $P(A_a | \omega_a^A) = \frac{1}{\sqrt{2\pi}|\Sigma_{A,a}|} \exp\{-(1/2)(A_a - \mu_a^A)^T(\Sigma_a^A)^{-1}(A_a - \mu_a^A)\}$ specified by its mean μ_a^A and covariance Σ_a^A . For each clique $c \in C(s)$ there is a quadratic potential on the spatial relations $\psi_c(\vec{l}_c, \omega_c^g) = -(1/2)(\vec{l}_c - \vec{\mu}_c^z)^T(\Sigma_c^z)^{-1}(\vec{l}_c - \vec{\mu}_c^z)$ defined on the invariant triplet vector \vec{l}_c of the clique and specified by the mean $\vec{\mu}_c^z$ and covariance Σ_c^z . This gives a

distribution over the invariant triplet vectors $P(l|\omega) = \frac{1}{Z} \exp\{-\sum_{c \in C} \psi_c(\vec{l}_c, \omega_c^g)\}$. Hence $\omega_1 = \{\mu_a^A, \Sigma_a^A, \vec{\mu}_c^z, \Sigma_c^z\}$ and Ω_1 specifies the prior $P(s), P(V|s)$, see [2] for details.

In [2], we describe how to eliminate the variable G from the generative model by integrating it out. This is a useful approximation computationally since it simplifies inference and learning. We estimate (s, V) using an approximate model that is invariant to the pose G and then we estimate G in a second stage. This approximation leads to a distribution of form:

$$P(\{z_i, A_i, \theta_i\}|V, s, \omega)P(V|s, \Omega)P(s|\Omega)P(\omega)P(\Omega) \quad (6)$$

with

$$\begin{aligned} &P(\{z_i, A_i, \theta_i\}|V, s, \omega^g, \omega^A, \Omega) = \\ &\frac{1}{Z} \prod_{a \in L_O(s): i(a) \neq 0} P(A_{i(a)}|s, \omega^A, V) \prod_{c \in C(L_O(s))} P(\vec{l}_c(\{z_{i(a)}, \theta_{i(a)}\})|s, \omega^g, V) \end{aligned} \quad (7)$$

The inference task requires estimating the aspect s , the assignments V by :

$$s^*, V^* = \arg \max_{s, V} P(\{z_i, \theta_i, A_i\}|V, s)P(V|s)P(s) \quad (8)$$

The strategy is as follows. For each s , we estimate $V^* = \arg \max_V P(\{z_i, \theta_i, A_i\}|V, s)P(V|s)P(s)$. This is performed by dynamic programming exploiting the fact that the logarithm of the distribution can be expressed as a sum of order triplets – see [2] (including our technique to make this robust to missing data). Then we estimate $s^* = \arg \max_s P(\{z_i, \theta_i, A_i\}|V^*, s)P(V^*|s)P(s)$ by exhaustive search. Finally we estimate G^* from the position, orientation, and scale of all the matched interest points.

The learning task is difficult because of two issues. Firstly, the structure of the POM-IP is unknown so we have to perform structure induction to search over the enormous number of possible structures. Secondly, parameter estimation can be performed by the expectation-maximization (EM) algorithm but requires good initial conditions to ensure good convergence.

Our strategy [2] for addressing these issues involves constructing a vocabulary of triplets cliques by performing k-means clustering on the interest points in the training images $\{I_\mu\}$. This gives a triplet vocabulary D which are representative of those triplets that frequently occur in the training images. We express D in form:

$$D = \{\mu_{abc}^g, \Sigma_{abc}^g, (\mu^{A,a}, \mu^{A,b}, \mu^{A,c}), (\Sigma^{A,a}, \Sigma^{A,b}, \Sigma^{A,c})\}, \quad (9)$$

where $\mu_{abc}^g, \Sigma_{abc}^g$ are the means and covariances of the triplet ITV, $\mu^{A,a}, \Sigma^{A,a}$ are the means and covariances of the appearances.

The triplet vocabulary has two purposes. Firstly it is used to propose structures for the POM-IP. We select a triplet from the dictionary based on its frequency in the training images. This triplet is used to propose a new structure for POM-IP either by adding it to an existing aspect model (if two of the triplet nodes are a good fit to an existing node of the aspect model) or by using it to start a new aspect model. Secondly, the parameters of the selected triplet (i.e. its means and covariances) are used as initial conditions for parameter learning, hence providing good initial conditions enabling good convergence.

Parameter learning can be formulated as estimating $\omega^*, \Omega^* = \arg \max_{\omega, \Omega} \prod_{\mu} P(\omega, \Omega | \{d_1(I_{\mu})\}) = \arg \max_{\omega, \Omega} \sum_{s_{\mu}, V_{\mu}} P(\omega, \Omega, \{s_{\mu}\}, \{V_{\mu}\} | \{d_1(I_{\mu})\})$. This can be performed by the EM algorithm by defining a free energy $F(Q, \omega, \Omega)$ where $Q(\{s_{\mu}\}, \{V_{\mu}\})$ is a distribution over the variables that are being summed out. The algorithm proceeds by alternatively minimizing $F(., .)$ with respect to $Q(., .)$ and ω, Ω . As shown in [2]: (i) minimizing with respect to $Q(., .)$ can be formed analytically (if ω, Ω is fixed), (ii) estimating ω, Ω can be performed by evaluating an analytic formula which requires summing over all possible states V, s (this is due to the simple exponential models – e.g. Gaussian distributions – used in POM-IP). The summation over V can be performed using dynamic programming (the sum rule) while the summation over s is performed analytically. The initial conditions for the parameters ω, Ω is provided by the triplet dictionary D (ensuring good convergence).

Structure learning proceeds by proposing a new structure by selecting a new triplet from the triplet dictionary D , adding it to the current structure, and then performing model selection to determine whether to keep the new structure or reject it. Model selection requires evaluating the evidence for the model $\max_{\omega, \Omega} P(\omega, \Omega) \prod_{\mu} \sum_{s_{\mu}} \sum_{V_{\mu}} P(d_1(I_{\mu}), s_{\mu}, V_{\mu} | \omega, \Omega)$. Summation over the states $\{V_{\mu}\}$ is performed by dynamic programming (sum rule) and summation over $\{s_{\mu}\}$ is done exhaustively.

As shown in [2], the POM-IP can be learnt with minimal supervision when the number of aspects is unknown and the pose (position, scale, and orientation) varies between images. Its performance on classification is comparable to other approaches evaluated on benchmarked data. Its inference is very rapid (seconds) due to the efficiency of dynamic programming. Nevertheless, the POM-IP is limited because its reliance only on interest points means that it gives poor

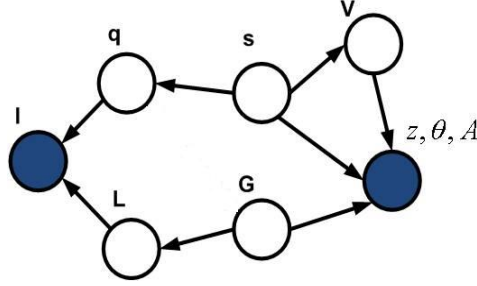


Fig. 3. The Bayes net for joining POM-IP to POM-mask.

performance on segmentation and fails to exploit all the image cues.

V. POM-MASK

The POM-mask uses regional cues to perform segmentation/localization. It is trained using knowledge from the POM-IP giving crude estimates for the segmentation (e.g. the bounding box of the IP's). This training enables POM-mask to learn a shape prior for each aspect of the object. After training, the POM-mask and POM-IP are coupled – figures (2,3). During inference, the POM-IP supplies estimates of pose and aspect to help estimate the POM-mask variables.

A. Overview of the POM-mask

The probability distribution of the POM-mask is defined by:

$$P(d_2(I)|L, q)P(L|M, G, u)P(M|s)P(u|s)P(q|s)P(s)P(G), \quad (10)$$

where I is the intensity image, $d_2(I)$ are the appearance features extracted – see section (III). L is a binary label field indicating which pixels belong the inside and the outside of the object, $q = (q_O, q_B)$ is a set of distributions on the image statistics inside and outside the object. L and q are the hidden states. $h_2 = \{L, q\}$ and $P(d_2(I)|L, q)$ is the model for generating the data when the labels L and distributions q are known. M, u are model parameters to be learnt (i.e. ω_2) defined as follows. The *probability mask* $M^s = \{M_i^s\}$ is defined for each aspect s so that $M_i^s \in [0, 1]$ is the probability that pixel i is inside the object with aspect s . The *displacement* u^s is the displacement between the center of the mask and the center of the interest points (specified by G) normalized for scale and orientation (also determined by G). The displacement u is required so that the mask can be scaled correctly about its center instead of the center of

the interest points. $P(L|M, G, u)$ is a prior on the labeling conditioned on the *probability mask* M , the pose G , and the displacement u . For a specific image, the probability masks undergo a geometric transformation $T(G, u)M^s$ specified by the pose variable G estimated by POM-IP and the displacement u . The probability masks are learnt automatically. The priors $P(M|s)$, $P(q|s)$ are mixture models specifying different probability masks and displacements for different aspects s . The prior $P(q|s)$ is set to be the uniform distribution because our attempts to learn it showed that it was extremely variable for most objects. $P(s)$ and $P(G)$ are the same as for POM-IP. Hence the model parameters $\omega_2 = \{M^s, u^s\}$ (i.e. the probability masks and displacements for different aspects).

The *inference* for the POM-mask estimates

$$q^*, L^* = \arg \max_{q, l} P(d_2(I)|L, q)P(L|M^{s^*}, u^{s^*}, G^*)P(u|s^*)P(M|s^*) \quad (11)$$

where G^* and s^* are the estimates of pose and aspect provided by POM-IP by knowledge propagation. Inference is performed by an alternative iterative algorithm similar to grab cut [15], [16], [17] described in detail in section (V-B). This algorithm requires initialization of L . Before learning has occurred, this estimate is provided by the bounding box of the interest points detected by POM-IP. After learning, the initialization of L is provided by the thresholded transformed probability mask $T(G^*, u^{s^*})M_i^{s^*}$.

Learning the POM-mask is also performed with knowledge propagated from the POM-IP. The main parameter to be learnt is the prior probability of the shape, which we represent by a *probability mask*. Given a set of images $\{d_2(I_\mu)\}$ we seek to find the probability masks $\{M^s\}$ and the displacements $\{u^s\}$. Ideally we should sum over the hidden states $\{L_\mu\}$ and $\{q_\mu\}$, but this is impractical so we maximize over them also. Hence we estimate $\{M^s\}, \{u^s\}, \{L_\mu\}, \{q_\mu\}$ by maximizing $\prod_\mu P(d_2(I_\mu)|L_\mu, q_\mu)P(L_\mu|M^{s_\mu^*}, u^{s_\mu^*}, G_\mu^*)P(M^s|s_\mu^*)P(u^s|s_\mu^*)$ where $\{s_\mu^*, G_\mu^*\}$ are estimated by POM-IP for image I_μ . This is performed alternative maximization with respect to $\{L_\mu\}, \{q_\mu\}$ and $\{M^s\}, \{u^s\}$ which combines grab-cut with steps to estimate $\{M_\mu\}, \{u^s\}$, see section (V-C).

B. POM-mask model details

The distribution $P(d_2(I)|L, q)$ is of form:

$$\frac{1}{Z} \exp\left\{\sum_i \phi_1(I_i|L_i, q) + \sum_{i,j \in Nbh(i)} \phi_2(I_i, I_j|L_i, L_j)\right\} \quad (12)$$

where i is the index of image pixel, j is a neighboring pixel of i and Z is the normalizing constant. This model gives a tradeoff between local (pixel) appearance specified by the unary terms and a binary terms which biases neighbouring pixels to have the same labels unless they are separated by a large intensity gradient.

The unary potential terms generate the appearance of the object and are given by:

$$\phi_1(I_i|L_i, h) = \begin{cases} \log q_O(I_i) & \text{if } L_i = 1 \\ \log q_B(I_i) & \text{if } L_i = 0 \end{cases}. \quad (13)$$

The binary potentials $\phi_2(I_i, I_j|L_i, L_j)$ is the edge contrast term [18] and makes edges more likely at places where there is a big intensity gradient:

$$\phi_2(I_i, I_j|L_i, L_j) = \begin{cases} \gamma(i, j) & \text{if } L_i \neq L_j, \\ 0 & \text{if } L_i = L_j \end{cases}. \quad (14)$$

where $\gamma(i, j) = \lambda \exp\{-\frac{g^2(i, j)}{2\gamma^2}\} \frac{1}{\text{dist}(i, j)}$, $g(\cdot, \cdot)$ is a distance measure on the colors I_i, I_j and $\text{dist}(i, j)$ measures the spatial distance between i and j . For more details, see [15], [16].

The prior probability distribution $P(L|M, G, u)$ for the labels L is defined as follows:

$$P(L|M, G, u) = \frac{1}{Z} \exp\{\sum_i \psi_1(L_i; G, u) + \sum_{i,j} \psi_2(L_i, L_j|\zeta)\} \quad (15)$$

The unary potentials use the probability mask as a prior for the labeling while the binary term is a homogeneity term which encourages neighbouring pixels to have similar labels unless there is a large edge gradient between them. The binary terms are particularly useful at the start of the learning because the probability mask is very inaccurate. As learning proceeds, the unary term becomes more important.

The unary potential $\psi_1(L_i; G, u)$ encodes a shape prior (probability mask) given by:

$$\psi_1(L_i; G, u) = L_i \log(T(G, u)M_i) + (1 - L_i) \log(1 - T(G, u)M_i), \quad (16)$$

where M is a probability mask which depends on the aspect (i.e. there are different probability masks for different aspects). Here $M_i \in [0, 1]$ is the probability that pixel i is inside the object. We denote $T(G, u)M_i$ to be the probability that pixel i is inside the object after transforming the mask by position, scale, and orientation (indexed by G and allowing for the displacement u).

The binary potential is of Ising form and encourages homogeneous regions:

$$\psi_2(L_i, L_j | \zeta) = \begin{cases} 0, & \text{if } L_i \neq L_j \\ \zeta, & \text{if } L_i = L_j \end{cases}. \quad (17)$$

where ζ is a parameter of the generic prior.

C. POM-mask inference and learning details:

The inference requires estimating

$$q^*, L^* = \arg \max_{q, L} P(d_2(I) | L, q) P(L | M^{s^*}, u^{s^*}, G^*) P(u | s^*) P(M | s^*) \quad (18)$$

where G^* and s^* are provided by POM-IP. Initialization of L is provided by the thresholded transformed probability mask $T(G^*, u^{s^*})M_i^{s^*}$, or by the bounding box of the interest points provided by POM-IP (used before the probability masks are learnt).

We perform inference by alternative maximization with respect to q, L . Formally,

$$\begin{aligned} q^{t+1} &= \arg \max_q P(d_2(I) | L^t, q) : \text{ which gives } q_O^{t+1}(\alpha) = f_O(\alpha, L^t), \quad q_B^{t+1}(\alpha) = f_B(\alpha, L^t) \\ L^{t+1} &= \arg \max L P(d_2(I) | L^t, q) P(L | M^{s^*}, u^{s^*}, G^*). \end{aligned} \quad (19)$$

The estimation of L^{t+1} is performed by max-flow [16]. This is similar to grab-cut [15], [16], [17] except that: (i) our initialization is performed automatically, (ii) our probability distribution differs by containing the probability mask. In practice we only performed a single iteration of each step since more iterations failed to give significant improvements.

The learning requires estimating the probability masks $\{M^s\}$, the displacement $\{u^s\}$, the labels $\{L_\mu\}$, and the distributions $\{q_\mu\}$ which maximize $\prod_\mu P(d_2(I_\mu) | L_\mu, q_\mu) P(L_\mu | M_\mu^{s^*}, u^{s^*}, G_\mu^*, u)$, where $\{s_\mu^*, G_\mu^*\}$ are estimated by POM-IP.

This is performed by alternative maximization with respect to $\{M^s\}$, $\{u^s\}$, $\{L_\mu\}$ and $\{q_\mu\}$. The methods for maximization with respect to $\{L_\mu\}$ and $\{q_\mu\}$ are those given in equation (19) performed for every image $\{I_\mu\}$ in the training dataset using the current value $\{M^{s,t}\}$ for the probability mask.

The maximization with respect to $\{M^s\}$ corresponding to estimating:

$$\{M^{s,t}\}^* = \arg \max \prod_\mu P(d_2(I_\mu) | L_\mu^t, q_\mu^t) P(L_\mu^t | M_\mu^{s^*}, u^{s^*}, G_\mu^*), \quad (20)$$

which reduces to setting:

$$M^{s,t} = \frac{\sum_{\mu} Id(s_{\mu}, s) T(G_{\mu}^*, u^{s_{\mu}^*})^{-1} L_{\mu}^t}{\sum_{\mu} Id(s_{\mu}, s)}, \quad (21)$$

where $Id(s_{\mu}, s)$ is an indicator variable that takes value 1 if $s_{\mu} = s$ and is 0 otherwise. Hence the estimate for $M^{s,t}$ is simply the average of the estimated labels L_{μ}^t for those images μ which are assigned (by POM-IP) to aspect s , where the pose of these labels has been transformed $T(G_{\mu}^*, u^{s_{\mu}^*})^{-1} L_{\mu}^t$ by the estimated pose L_{μ}^t . (Note we use $T(G, u)$ to transform the probability mask M to the label L , so $T(G, u)^{-1}$ is used to transform L to M).

The maximization with respect to u^s can be approximate by $u^{s,t+1} = k(L^t, G^*, s^*)$ where $k(L^t, G^*, s^*)$ is the displacement between the center of the label L^t and the pose center adjusted by the scale and orientation (all obtained from G^*) for aspect s^* .

In summary, the POM-mask gives significantly better segmentation than the POM-IP alone (see results section). In addition, it provides context for the POM-edgelets.

VI. THE POM-EDGELET MODELS

The *POM-edgelet distribution* is of the same form as POM-IP but does not include A attributes (i.e. the edgelets are specified only by their position and orientation). The data $d_3(I)$ is the set of edges in the image. The hidden states h_3 are the correspondence V between the nodes of the models and the edgelets. The pose and aspect are determined by the pose and aspect of the POM-IP.

Once the POM-mask model has been learnt we can use it to teach POM-edgelets which are defined on sub-regions of the shape (adjusted for our estimates of pose and aspect). Formally the POM-mask provides a mask L^* which is decomposed into non-overlapping subregions $L^* = \bigcup_{i=1}^6 L_i^*$ where $L_i^* \cap L_j^* = 0$ for $i \neq j$. There are 6 POM-edgelets which are constrained to lie within these different subregions during learning and inference. (Note that training a POM-edgelet model on the entire image is impractical because the numbers of edgelets in the image is orders of magnitude larger than the number of interest points, and all edgelets have similar appearances). The method to *learn* the POM-edgelets is exactly the same as the one for learning the POM-IP except we do not have appearance attributes and the sub-region where the edgelets appear is fixed to a small part of the image (i.e. the estimate of the shape of the sub-region).

The *inference* for the POM-edgelets requires an estimate for the pose G and aspect s which is supplied by the POM-IP (the POM-mask is only used in the learning of the POM-edgelets).

VII. RESULTS

We now give results for a variety of different tasks and scenarios. We compare performance of the POM-IP [1] and the full POM. We collect 14 classes (see figure 4) from Caltech 101 [23]. In all experiments, we learnt the full POM on a *training set* consisting of half the set of images (randomly selected) and evaluated the full POM on the remaining images, or *testing set*. The images in the dataset were required to have at least fifty images to ensure that there was sufficient data in the training set to learn the POM's. Some of the images had complex and varied image backgrounds while others had comparatively simple backgrounds (we observed no changes in performance based on the complexity of the backgrounds, but this is a complex issue which deserves more investigation).

The speed for inference is less than 5 seconds on a 450×450 image. This breaks down into 1 second for interest-point detector and SIFT descriptor, 1 second for edge detection, 1 second for the graph cut algorithm, and 1 second for the image parsing. The training time for 250 images is approximately 4 hours.

Overall our experiments show the following three effects demonstrating the advantages of the full POM compared to POM-IP. Firstly, the performance of the full POM for classification is better than POM-IP (because of the extra information provided by the POM-edgelets). Secondly, the full POM provides significantly better segmentation than the POM-IP (due to POM-mask). Thirdly, the full POM enables denser matching between different objects of the same category (due to the edgelets in the POM-edgelets). Moreover, as for POM-IP [2], the inference and learning is invariant to scale, position, orientation, and aspect of the object.

A. The Tasks

We tested on three tasks: (I) The *classification* task is to determine whether the image contains the object or is simply background. This is measured by the classification accuracy. (II) The *segmentation* task is evaluated by *precision and recall*. The precision $|R \cap GT|/|R|$ is the proportion of pixels in the estimated shape region R that are in the ground-truth shape region GT . The recall $|R \cap GT|/|GT|$ is the proportion of pixels in the ground-truth shape region that are in the estimated shape region. (III) The *recognition* task which we illustrate by showing matches.

TABLE II
COMPARISONS OF CLASSIFICATION WITH RESULTS REPORTED IN [3], [11], [1].

Dataset	full POM	PGMM[1]	Constellation[3]	Sivic et al.[11]
Faces	98.0	98.0	96.4	96.7
Airplane	91.8	90.9	90.2	98.4
Motorbikes	94.6	92.6	92.5	92.0

We performed these tests for three scenarios: (I) *Single object category* when the training and testing images containing an instance of the object with unknown background. Due to the nature of the datasets we used there is little variation in orientation and scaling of the object, so the invariance of our learning and inference was not tested. (II) *Single object category with variation* where we had manipulated the training and testing data to ensure significant variations in object orientation and scale. (III) *Hybrid object category* where the training and testing images contain an instance of one of three objects (face, motorbike, or airplane).

B. Scenario 1: Classification for Single object category

In this experiment, the training and testing images come from a single object class. The experimental results, see table (III), show improvement in *classification* when we use the full POM (compared to the POM-IP/PGMM). These improvements are due entirely to the edgelets in the full POM. We note that the regional features from POM-mask supply no information for object classification because the appearance model is very weak (i.e. the q_O distribution has uniform prior). The improvements are biggest for those objects where the edgelets give more information compared to the interest points (e.g. the football, motorbike, and grand piano). Compare with the results reported in [3], [11], [1] shown in table (II).

C. Scenario 2: Segmentation for Single object category

Observe that *segmentation* (see table (IV)) is extremely improved by using the full POM compared to the POM-IP. To evaluate these comparisons we show improvements between using the PGMM model, the POM-IP model (with grab-cut), the POM-IP combined with the POM-mask, and the full POM.. The main observation is that the bounding box round the interest-points is only partially successful. There is a bigger improvement when we use the interest-points to

TABLE III
CLASSIFICATION RESULTS ON 14 CLASSES

Dataset	POM-IP	full POM
Faces	95.8	98.0
Airplane	91.3	91.8
Motorbikes	86.1	94.6
Accordion	97.5	97.5
Buddha	88.8	91.3
Car	89.2	90.3
Football	68.3	78.3
Ketch	85.0	87.0
Menorah	71.3	73.8
Grand Piano	87.0	93.0
Starfish	78.5	78.5
Sunflower	86.3	88.8
Watch	86.5	90.5
Windsor Chair	97.5	97.5
14-class Average	86.4	89.4

initialize a grab-cut algorithm. But the best performance occurs when we use the edgelets. We also compare our method with [12] for segmentation. See the comparisons in table V.

D. Performance for different object categories

To get better understanding of segmentation and classification results, and the relative importance of the different components of the full POM, consider figure (4) where we show examples for each object category and see table (III, IV). The first column shows the input image and the second column gives the bounding box of the interest points of POM-IP. Observe that this bounding box only gives a crude segmentation and can lie entirely inside the object (e.g. face, football), or encompass the object (e.g. car, starfish), or only capture a part of the object (e.g. accordion, airplane, grand piano, windsor chair). The third column shows the results of using grab-cut initialized by the POM-IP. This gives reasonable segmentations for some objects (e.g. accordion, football) but has significant errors for others (e.g. car, face, clock, windsor chair) sometimes capturing large parts of the background while missing significant parts of the object (e.g. windsor chair). The fourth column shows the POM-mask learns good shape priors

TABLE IV

THE COMPARISONS OF SEGMENTATION BY DIFFERENT POMs. THE PRECISION AND RECALL MEASURE IS REPORTED. “AVERAGE”: THE AVERAGE PERFORMANCE ON 14 CLASSES.

Dataset	PGMM[1]	POM-IP	POM-IP + POM-Mask	full POM
Accordion	77 / 48.1	80.6 / 43.0	88.3 / 43.8	88.2 / 44.0
Airplane	44 / 62.5	61.4 / 75.9	73.9 / 75.1	75.2 / 75.4
Buddha	70 / 64.5	76.0 / 85.4	78.4 / 84.2	80.9 / 83.4
Car	31 / 89.6	28.0 / 61.6	52.0 / 50.7	50.0 / 54.3
Face	86 / 64.4	72.6 / 87.0	72.2 / 89.3	73.5 / 89.6
Football	84 / 47.8	96.8 / 50.5	94.9 / 51.1	93.0 / 62.6
Ketch	63 / 63.9	67.9 / 69.7	67.1 / 72.5	69.8 / 71.0
Menorah	62 / 43.6	73.2 / 35.4	77.4 / 31.6	74.2 / 38.3
Motorbike	65.6 / 84.2	80.9 / 71.8	88.2 / 69.6	82.8 / 86.3
Grand Piano	73.1 / 54.5	86.2 / 61.5	88.0 / 76.8	87.8 / 81.3
Starfish	42.8 / 74.2	71.5 / 77.5	74.5 / 73.1	77.1 / 78.5
Sunflower	82.8 / 66.7	87.9 / 79.4	86.9 / 81.7	87.9 / 81.8
Watch	82.2 / 64.4	94.0 / 63.4	94.9 / 63.9	95.4 / 69.2
Windsor Chair	84.2 / 52.0	84.7 / 55.8	85.3 / 54.0	94.4 / 56.0
Average	67 / 62	75 / 65	80 / 65	80 / 69

TABLE V

SEGMENTATION COMPARISON WITH CAO AND FEIFEI [12]. THE MEASURE OF SEGMENTATION ACCURACY IN PIXELS IS USED.

	POM	Cao and Feifei[12]
Faces easy	86.0%	78.0%
Motorbikes	79.0%	77.0%
Grand Piano	84.8%	78.0%
Starfish	85.9%	69.0%
Sunflower	86.2%	86.0%
Watch	75.5%	60.0%

TABLE VI
CLASSIFICATION RESULTS WITH VARIABLE SCALE AND ORIENTATION.

	POM	[1]
Faces	98.0	98.0
Faces(Scaled)	96.5	-
Faces(Rotated)	96.7	94.8
Faces(Scale+Rotated)	94.6	92.3

(probability masks) for all objects despite the poorness of some of the initial segmentation results. This column also shows the positions of the edgelet features learn by the POM-edgelets. The thresholded probability mask is shown in the fifth column and we see that it takes reasonable forms even for the windsor chair. The sixth column show the results of using the full POM model to segment these objects (i.e. using the probability mask as a shape prior) and we observe that the segmentations are good and significantly better than those obtained using grab-cut only. Observe that the background is almost entirely removed and we now recover the missing parts, such as the legs of the chair and the rest of the grand piano. Finally, the seventh column illustrates the locations of the feature points (interest points and edgelets) and shows that the few errors occur for the edgelets at the boundaries of the objects.

E. Scenario 3: Varying the scale and orientation of the objects

The full POM is designed so that it is invariant to scale and rotation for both learning and inference. This advantage was not exploited in scenario 1, since the objects tended to have similar orientations and sizes. To emphasize and test this invariance, we learnt the full POM for a data-set of faces where we scaled, translated, and rotated the objects, see figure (5). The scaling was from 0.6 to 1.5 (i.e. by a factor of 2.5) and the rotation was uniformly sampled from 0 to 360 degrees. We considered three cases where we varied the scale only, the rotation only, and scale and rotation. The results, see table (VI,VII), show only slight degradation in performance for the tasks.



Fig. 4. The columns show the fourteen objects that we used. The seven columns are labelled left to right as follows: (1) Original Image, (2) the Bounding Box specified by POM-IP , (3) the GraphCut segmentation with the features estimating using the Bounding Box, (4) the probability object-mask with the edgelets (green means features within the object, red means on the boundary), (5) the thresholded probability mask,(6) the new segmentation using the probability object-mask (i.e. POM-IP + POM-mask), (7) the parsed result.

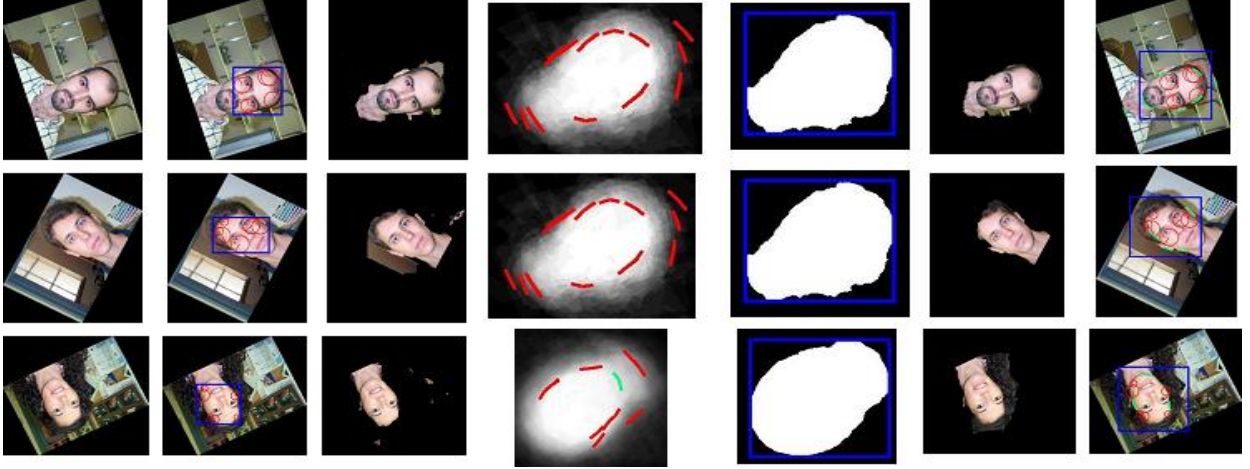


Fig. 5. The full POM can be learnt even when the training images are randomly translated, scaled and rotated.

TABLE VII

COMPARISONS OF SEGMENTATION BY DIFFERENT POMs WHEN SCALE AND ORIENTATION ARE VARIABLE. THE PRECISION AND RECALL MEASURE IS REPORTED.

Dataset	PGMM[1]	POM-IP	POM-IP + POM-Mask	full POM
Faces	86 / 64.4	72.6 / 87.0	72.2 / 89.3	73.5 / 89.6
Faces(Scaled)	83 / 63	71 / 90	76 / 87	76 / 89
Faces(Rotated)	80 / 61	62 / 90	70 / 88	70 / 90
Faces(Sscaled+Rotated)	81 / 57	63 / 84	68 / 85	68 / 87

F. Scenario 4: Hybrid Object Models

We now make the learning and inference tasks even harder by allowing the training images to contain several different types of objects (extending work in [1] for the PGMM). More specifically, each image will contain either a face, a motorbike, or an airplane (but we do not know which). The full POM will be able to successfully learn a hybrid model because the different objects will correspond to different aspects. It is important to realize that we can identify the individual objects as different aspects of the full POM, see figure (6). In other words, the POM does not only learn the hybrid class, it also learns the individual object classes in an unsupervised way.

The performance of learning this hybrid class is shown in table (VIII,IX). We see that the

TABLE VIII

THE CLASSIFICATION RESULTS FOR HYBRID MODELS

Dataset	full POM	PGMM[1]
Hybrid	87.8	84.6

TABLE IX

THE SEGMENTATION RESULTS FOR HYBRID MODELS USING DIFFERENT POMs. THE PRECISION AND RECALL MEASURE IS REPORTED.

Dataset	PGMM[1]	POM-IP	POM-IP + POM-Mask	full POM
Hybrid	60 / 61	69 / 72	77 / 65	73 / 73

performance degrades very little, despite the fact that we are giving the system even less supervision. The confusion matrix between faces, motorbikes and airplanes is shown in table X. Our result is slightly worse than [11].

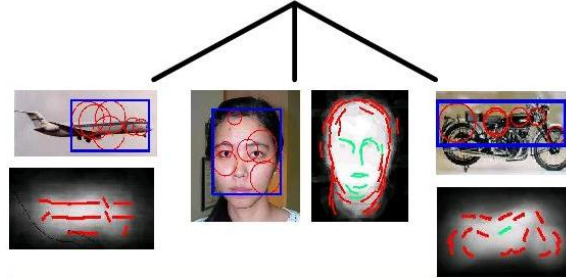


Fig. 6. Hybrid Model. The training images consist of faces, motorbikes and airplanes but we do not know which type of object is in the image.

G. Scenario 5: Matching and Recognition

This experiment was designed as a preliminary experiment to test the ability of the POM-IP to perform recognition (i.e. to distinguish between different objects in the same object category). These experiments show that the POM-IP is capable of performing matching and recognition. Figure 7 shows an example of correspondence between two images. This correspondence is obtained by first performing inference to estimate the configuration of POM-IP and then to

TABLE X

THE CONFUSION MATRIX FOR THE HYBRID MODEL. THE MEAN OF THE DIAGONAL IS 89.8% (I.E. CLASSIFICATION ACCURACY) WHICH IS COMPARABLE WITH THE 92.9% REPORTED IN [11].

	Face	Motorbikes	Airplanes
Face	96.0%	0.0%	4.0%
Motorbikes	2.2%	85.4%	10.4%
Airplanes	2.0%	10.0%	88.0%

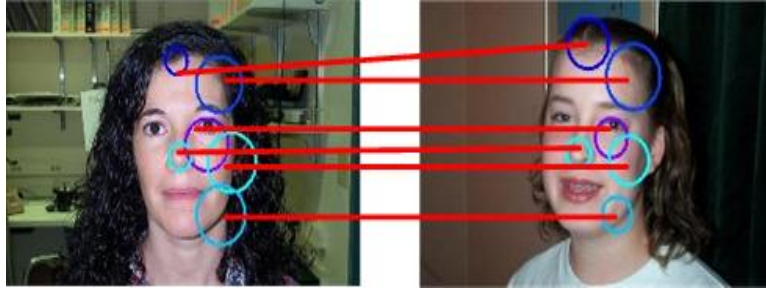


Fig. 7. An example of correspondence obtained by POM.

match corresponding nodes). For recognition, we use 200 images containing 23 persons. Given a query of a image containing a face, we output the top three candidates from the 200 images. The similarity between two images is measured by the differences of intensity of the corresponding interest points. The recognition results are illustrated in figure 8. The results are promising.

VIII. DISCUSSION

This paper is part of a research program where the goal is to learn object models for all object-related visual tasks. In this paper we built on previous work [1], [2] which used weak supervision to learn a probabilistic grammar Markov model (PGMM) which used interest point features and performed classification. Our extension is based on combining elementary probabilistic object models (POM's) which use different visual cues and can combine to perform a variety of visual tasks. The POM's cooperate to learn and do inference by *knowledge propagation*. In this paper, the POM-IP (or PGMM) was able to train a POM-mask model so that the combination could perform localization/segmentation. In turn, the POM-mask was able to train a set of POM-edgelets which when combined into a full POM can use edgelet features to improve the

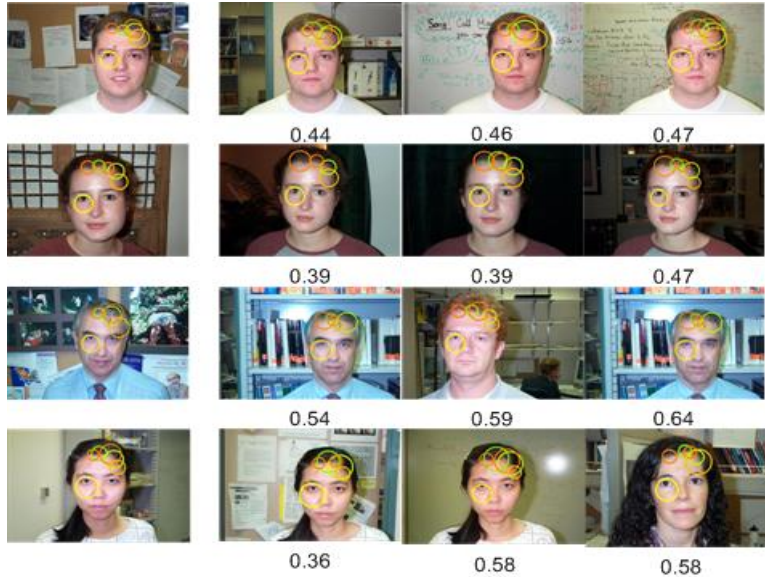


Fig. 8. Recognition Examples. The first column is the prototype. The next three columns show the top three rankings. A distance to the prototype is shown under each image

classification. We demonstrated this approach on large numbers of images of different objects. We also showed the ability of our approach to learn and perform inference when the scale and rotation of objects is unknown. We showed its ability to learn a hybrid model containing several different objects. The inference is performed in seconds, and the learning in hours.

IX. ACKNOWLEDGMENTS

This research was supported by NSF grant 0413214 and the W.M. Keck Foundation. We thank Iasonas Kokkinos, Zhuowen Tu, and YingNian Wu for helpful feedback.

REFERENCES

- [1] L. Zhu, Y. Chen, and A. L. Yuille, “Unsupervised learning of a probabilistic grammar for object detection and parsing,” in *NIPS*, 2006, pp. 1617–1624.
- [2] —, “Unsupervised learning of probabilistic grammar-markov models for object categories,” in *To appear in TPAMI*, 2008.
- [3] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *CVPR (2)*, 2003, pp. 264–271.
- [4] —, “A sparse object category model for efficient learning and exhaustive recognition,” in *CVPR (1)*, 2005, pp. 380–387.
- [5] D. J. Crandall and D. P. Huttenlocher, “Weakly supervised learning of part-based spatial models for visual object recognition,” in *ECCV (1)*, 2006, pp. 16–29.

- [6] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *ECCV'04 Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004, pp. 17–32.
- [7] E. Borenstein and S. Ullman, "Learning to segment," in *ECCV (3)*, 2004, pp. 315–328.
- [8] A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation," in *ECCV (4)*, 2006, pp. 581–594.
- [9] X. Ren, C. Fowlkes, and J. Malik, "Cue integration for figure/ground labeling," in *NIPS*, 2005.
- [10] J. M. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *ICCV*, 2005, pp. 756–763.
- [11] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their localization in images," in *ICCV*, 2005, pp. 370–377.
- [12] L. Cao and L. Fei-Fei, "Spatially coherent latent topic model for concurrent object segmentation and classification," in *ICCV*, 2007.
- [13] Y. Chen, L. Zhu, A. L. Yuille, and H. Zhang, "Unsupervised learning of probabilistic object models (poms) for object classification, segmentation and recognition," in *CVPR*, 2008.
- [14] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," in *EMMCVPR*, 2001, pp. 359–374.
- [15] A. Blake, C. Rother, M. Brown, P. Pérez, and P. H. S. Torr, "Interactive image segmentation using an adaptive gmmrf model," in *ECCV (1)*, 2004, pp. 428–441.
- [16] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV*, 2001, pp. 105–112.
- [17] C. Rother, V. Kolmogorov, and A. Blake, "'grabcut': interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.
- [18] M. P. Kumar, P. H. S. Torr, and A. Zisserman, "Obj cut," in *CVPR (1)*, 2005, pp. 18–25.
- [19] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.
- [20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [21] Y. Amit and D. Geman, "A computational model for visual selection," *Neural Computation*, vol. 11, no. 7, pp. 1691–1715, 1999.
- [22] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, 2005.
- [23] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Comput. Vis. Image Underst.*, vol. 106, no. 1, pp. 59–70, 2007.