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

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

Yuanhao Chen, *Member*, *IEEE*, Long (Leo) Zhu, *Member*, *IEEE*, Alan Yuille, *Fellow*, *IEEE*, and Hongjiang Zhang, *Fellow*, *IEEE*

Abstract—We present a method to learn probabilistic object models (POMs) with minimal supervision, which 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 elementary POMs that make use of complementary image cues. We describe a novel structure induction procedure, which uses *knowledge propagation* to enable POMs to provide information to other POMs 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 that 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 data sets for classification and segmentation with comparison to other methods. Inference takes five seconds while learning takes approximately four hours. In addition, we show that the full POM is invariant to scale and rotation of the object (for learning and inference) and can learn hybrid objects classes (i.e., when there are several objects and the identity of the object in each image is unknown). Finally, we show that POMs can be used to match between different objects of the same category, and hence, enable objects recognition.

Index Terms—Unsupervised learning, object classification, segmentation, recognition.

1 Introduction

Recent work on object classification and recognition has tended to represent objects in terms of spatial configurations of features at a small number of interest points [3], [4], [5], [6], [7], [8]. 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: 1) the sparseness of their representations restricts the set of visual tasks they can perform and 2) these models only exploit a small set of image cues. Sparseness is suboptimal for tasks such as segmentation that 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], [7], [8], [1], [2], [9]—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). In contrast, the segmentation literature—e.g., [10], [11], [12]—uses dense representations but typically requires that the precise localization/segmentation of the objects is given in the training images. But until recently—e.g., [13], [14], [15]—there have been few attempts to combine segmentation and classification or to make use of multiple visual cues.

Pattern theory [16], [17] gives a theoretical framework to address these issues—represent objects by state variables W, specify a generative model $P(\mathbf{I}|W)P(W)$ for obtaining the observed image $\mathbf{I}$, and an inference algorithm to estimate the most probable object state $W^* = \arg max_W P(W|\mathbf{I})$. The estimated state W^* determines the identity, pose, configuration, and other properties of the object (i.e., 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

- Y. Chen is with the Department of Automation, University of Science and Technology of China, Hefei, Anhui 230026, P.R. China. E-mail: yhchen4@ustc.edu.
- L. Zhu is with the Massachusetts Institute of Technology, 32-D462 CSAIL MIT, 32 Vassar Street, Cambridge, MA 02139. E-mail: leozhu@csail.mit.edu.
- A. Yuille is with the Department of Statistics, Psychology, and Computer Science, University of California, 8125 Math Science Bldg., Los Angeles, CA 90095. E-mail: yuille@stat.ucla.edu.
- H. Zhang is with the Microsoft Advanced Technology Center, 3/F, Beijing Sigma Center No. 49, Zhichun Road, Hai Dian District, Beijing 100080, China. E-mail: hjzhang@microsoft.com.

Manuscript received 31 Aug. 2008; revised 9 Jan. 2009; accepted 7 Apr. 2009; published online 30 Apr. 2009.
Recommended for acceptance by Q. Ji, A. Torralba, T. Huang, E. Sudderth, and J. Luo.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMISI-2008-08-0583.
Digital Object Identifier no. 10.1109/TPAMI.2009.95.

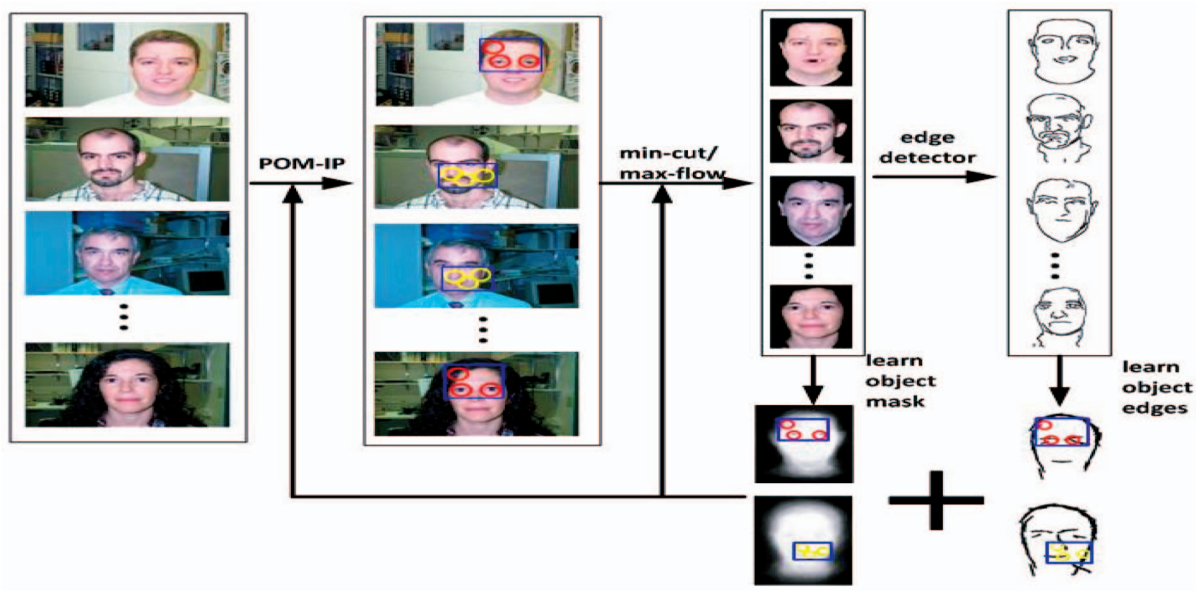

Fig. 1. The flowchart of knowledge propagation. POM-IP is learned and then trains POM-mask (using max-flow/min-cut), which includes learning a probabilistic object mask (see the feedback arrows). Then POM-IP and POM-mask help train POM-edgelets by using the object mask to provide context for the nine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

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., to 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 strategy is to first learn a simple model that only has a sparse representation of the object, and hence, only explains part of the data and performs a restricted set of tasks. Once learned, 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 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 learned 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 [18].

We formulate our approach in terms of probabilistic inference and machine learning. From this perspective, learning POMs is a structure induction problem [19], 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 4.

For concreteness, we now briefly step through the process of structure learning by KP as it occurs in this paper—see Fig. 1. First, we learn a POM defined on interest points (IPs), 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 learned from a set of images each of which contains 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 learned automatically). This POM-IP is able to detect and classify objects, detect their aspect, deal automatically with scaling and rotation changes, and give very crude estimates for segmentation. Second, we extend this model by incorporating different cues to enable accurate segmentation and improve classification. More specifically, we use the POM-IP to train a POM-mask, which uses regional image cues to perform segmentation. Intuitively, we start by using a version of grab-cut [20], [21], [22], [23], where POM-IP substitutes for human interaction to provide the initial estimate of the segmentation (as motion cues do in ObjCut [24]). This, by itself, yields a fairly poor segmentations of the objects. 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 is capable of performing classification and segmentation/localization. Third, the combination of POM-IP and POM-mask allows us to estimate the shape of the object and 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 that makes it easy to learn them and perform inference with them). After the models have been learned, KP is also used so that POM-IP provides estimates of pose (scale, position, and orientation), which helps provide initial conditions for POM-mask that, 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. Jojic et al. [25] provide alternative unsupervised learning approach, which addresses model coupling for video segmentation problem.

Our experiments demonstrate the success of our approach. First, we show that the full POM—coupling POM-IP, POM-mask, and POM-edgelet—performs better for classification than POM-IP alone. Second, the segmentation obtained by coupling POM-IP with POM-mask is much better than performing segmentation with grab-cut initialized by 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], [14], [15]. 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 knowledge propagation strategy in Section 2. Next, we give detail specifications of the image cues and the representations used in this paper in Section 3. Then, we specify the details of the POMs and KP in Sections 4, 5, and 6. Finally, we report the results in Section 7.

2 LEARNING BY KNOWLEDGE PROPAGATION

We now describe our strategy for learning by knowledge propagation. Suppose that 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 modeling those parts of the data, which are the easiest. This will provide the context making it easier to learn models for the rest of the data.

To make this specific, consider learning a probability model for an object and background, see Fig. 2, which uses two types of cues: 1) sparse interest points (IPs) and 2) dense regional cues. The object can occur at a range of scales, positions, and orientations. Moreover, the object has several aspects whose appearance varies greatly and whose number is unknown. In previous work [1], [2], we have described how to learn a model POM-IP, which is capable of modeling the interest points of the object (and the background). After learning, the POM-IP is able to estimate the pose (position, scale, and orientation) and aspect of the object for new images. We now want to enhance this model by using additional regional cues and a richer representation of the object. To do this, we want to couple POM-IP with a POM-mask, which has a mask for representing the object (one mask for each aspect) and exploits the regional cues. Our strategy, *knowledge propagation*, involves learning the full POM sequentially by first learning the POM-IP and then the POM-mask. We perform sequential learning—learning POM-IP and then using it to train POM-mask (because we do not know any direct algorithm to learn both simultaneously).

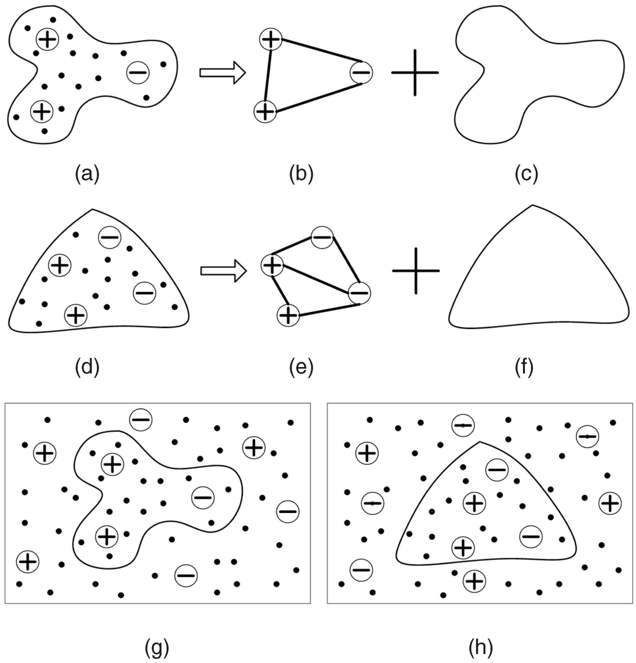

Fig. 2. The object is composed of a mask (thick closed contour) plus interest points (pluses and minuses) and has two aspects. (a) The first aspect is composed of (b) a POM-IP and (c) a POM-mask. Similarly, (d) the second aspect is composed of (e) a POM-IP and (f) a POM-mask. Panels (g) and (h) show examples, where the object is embedded in an image. Learning the POM-IP is practical, by the techniques described in [1], [2], but learning the POM-mask—or the full POM that combines IPs with the mask—is difficult because of the number of aspects (only two shown here) and the variability in scale and orientation (not shown). But the POM-IP is able to help train the POM-mask—by providing estimates of scale, orientation, and position—and facilitate learning of a full POM.

We now describe the basic ideas for a simple model and then return to the more complex models required by our vision application (which include additional models trained by both POM-IP and POM-mask).

To put this work in context, we recall the basic formulation of unsupervised learning and inference tasks. Suppose that we have data $\{d^\mu\}$ that is a set of samples from a generative model $P(d|h,\lambda)P(h|\Lambda)$ with hidden states h and model parameters λ, Λ. The two tasks are: 1) to learn the model—i.e., determine λ, Λ by MAP estimation $\lambda^*, \Lambda^* = \arg\max_{\lambda,\Lambda} P(\lambda, \Lambda|\{d^\mu\})$ using training data $\{d^\mu\}$ (which also includes learning the structure of the model) and 2) to perform inference from d to determine $h(d)$ by MAP $h^*(d) = \arg\max_h P(h|d,\lambda,\Lambda)$. But, as described in Section 1, there may not be efficient algorithms to achieve these tasks.

The basic idea of knowledge propagation can be illustrated as follows, see Fig. 3. 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 $P(d_1|h_1,\lambda_1)P(d_2|h_2,\lambda_2)P(h_1|\Lambda_1)P(h_2|h_1,\Lambda_2)$. These are essentially two models for generating different parts of the data, which are linked by the coupling term $P(h_2|h_1,\Lambda_2)$, as in Fig. 3. Knowledge propagation proceeds by first decoupling the models and learning the model by setting $\hat{\lambda}_1, \hat{\Lambda}_1 = \arg\max_{\lambda_1,\Lambda_1} \prod_\mu \sum_{h_1} P(d_1^\mu|h_1,\lambda_1)P(h_1|\Lambda_1)$ from the data $\{d_1^\mu\}$ (i.e., ignoring the $\{d_2^\mu\}$). Once this model has been learned, we can use it to make inference of the hidden state $h_1^*(d)$. This

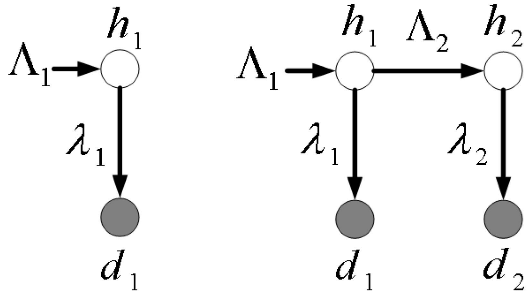

Fig. 3. Knowledge propagation. Left panel: The model for $P(d_1|h_1)P(h_1)$, where the likelihood and prior are specified by λ_1, Λ_1. Right panel: Learning the structure and parameters λ_1, Λ_1 for $P(d_1|h_1)P(h_1)$ enables us to learn a model with additional hidden states h_2, data d_2, and parameters λ_2, Λ_2. We can also perform inference on h_2 by first estimating h_1 using model $P(d_1|h_1)P(h_1)$.

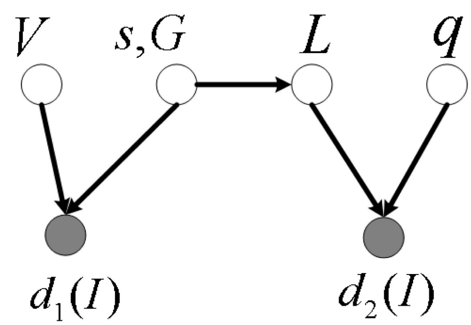

Fig. 4. The coupling between POM-IP and POM-mask is provided by the G, s variables for pose and aspect. This yields a full Bayes net containing IP-nodes and mask-nodes. Learning of the parameters of the POM-mask is facilitated by the POM-IP.

provides information that can be used to learn the second part of the model—i.e., to estimate $\lambda_2^*, \Lambda_2^* = \arg\max_{\lambda_2,\Lambda_2} \prod_\mu \sum_{h_2} P(d_2^\mu|h_2, \lambda_2)P(h_2|h_1^*(d^\mu), \Lambda_2)$. 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, such as belief propagation or Gibbs sampling (e.g., converge to a maxima of $\prod_\mu P(d_1^\mu|h_1, \lambda_1)P(d_2^\mu|h_2, \lambda_2)P(h_1|\Lambda_1)P(h_2|\Lambda_2)$, by doing maximization with respect to λ_1, Λ_1 and λ_2, Λ_2 alternatively). 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 $h_1^*(d) = \arg\max_{h_1} P(d_1|h_1)P(h_1)$ and then estimate $h_2^*(d) = \arg\max_{h_2} P(d_2|h_2)P(h_2|h_1^*(d))$. Once again, we can improve these estimates by using them as initial conditions for an algorithm that converges to a maxima of $P(d_1|h_1)P(h_1)P(d_2|h_2)P(h_2)$ by doing maximization with respect to h_1 and h_2 alternatively. It is straightforward to extend knowledge propagation— both learning and inference—to other sources of data $d_3, d_4, \ldots$ and hidden states $h_3, h_4, \ldots$.

In this paper, $d_1(\mathbf{I})$ denotes the set of IPs in the image, see Fig. 2. The variable $h_1 = (V, s, G)$ determines the correspondence V between observed IPs and IPs in the model, s respects the aspect of the model (a choice of mixture component), and G is the pose of the object (position, scale, and orientation). The model parameters λ_1, Λ_1 are described in Section 4. We refer to the probability distribution over this model $P(d_1(\mathbf{I})|s, V, G)P(s)P(V)P(G)$ as POM-IP. The form of this model means that we can do efficient inference and learning (including structure induction) without needing to know the pose G or the aspect s [1], [2]. See Section 4 for the full description.

$d_2(\mathbf{I})$ are feature vectors (e.g., color, or intensity, values) computed at each pixel in the image. The variable $h_2 = (L, \vec{q})$ denotes the labeling L (e.g., inside or outside boundary), and the distributions $\vec{q} = (q_O, q_B)$ specify the distribution of the features inside and outside the object. The POM-mask is defined by the distributions $P(d_2(\mathbf{I})|L, \vec{q})P(L|G, s)P(\vec{q})$ and specified by corresponding model parameters λ_2, Λ_2, see Section 5. Inference and learning are considerably harder for POM-mask if not intractable (without a POM-IP or other help). Attempts to learn image masks (e.g., [26]) assume very restricted transformation of the object between images (e.g., translation), a single aspect s, or make use of motion flow (with similar restrictions). But, as we show in this paper, POM-IP can provide the estimates of the pose G, the aspect s, and a very crude estimation of the object mask (given by the bounding box of the interest points), which are sufficient to teach the POM-mask and perform inference after the POM-mask has been learned.

The coupling between POM-IP and POM-mask is performed by the variables G, s, see Fig. 4, which extends Fig. 3.

Learning the POM-mask will enable us to train additional models that are specified within specific subregions of the object. Once POM-mask has been applied, we can estimate the image region corresponding to the object, and hence, identify the subregions. This provides sufficient context to enable us to learn models POM-edgelets defined on edgelets, see Section 6, which occur within specific subregions of the object. The full POM is built by combining a POM-IP with a POM-mask and POM-edgelets, see Figs. 4 and 1.

3 The Image Representation

This section describes the different image features that we use: 1) interest points (used in POM-IP), 2) regional features (in POM-mask), and 3) edgelets (in POM-edgelet).

The *interest point features* $d_1(\mathbf{I})$ of an image $\mathbf{I}$ used in POM-IP are represented by a set of attributed features $d_1(\mathbf{I}) = \{z_i\}$, where $z_i = (\vec{x}_i, \theta_i, A_i)$ with $\vec{x}_i$ the position of the feature in the image, θ_i is the feature's orientation, and A_i is an appearance vector. The procedures used to detect and represent the feature points were described in [1], [2]. Briefly, we detect interest points and determine their position $\vec{x}$ by Kadir-Brady [27] and represent them by the SIFT descriptor [28] using principal component analysis to obtain a 15-dimensional appearance vector A and an orientation θ.

The *regional image features* $d_2(\mathbf{I})$ used in POM-mask are computed by applying a filter $\rho(\cdot)$ to the image $\mathbf{I}$ yielding a set of responses $d_2(\mathbf{I}) = \{\rho(\mathbf{I}(\vec{x})) : \vec{x} \in D\}$, where D is the image domain. POM-mask will split the image into the object region $\{\vec{x} \in D \,\mathrm{s.t.} L(\vec{x}) = 1\}$ and the background region $\{\vec{x} \in D \,\mathrm{s.t.} L(\vec{x}) = 0\}$. POM-mask requires us to compute the feature histograms, $f_O(., L)$ and $f_B(., L)$, of the filter $\rho(\cdot)$ in both regions:

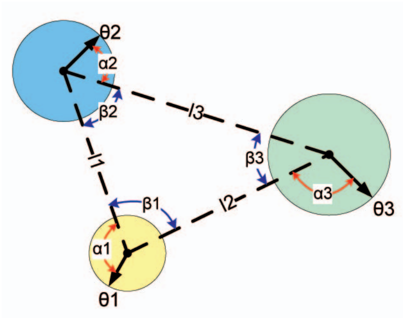

Fig. 5. The oriented triplet is specified by the internal angles β, the orientation of the vertices θ, and the relative angles α between them.

$$f_O(\alpha, L) = \frac{1}{|D_O|} \sum_{\vec{x} \in D} \delta_{L(\vec{x}),1} \delta_{\rho(\mathbf{I}(\vec{x})),\, \alpha}, \tag{1}$$

$$f_B(\alpha, L) = \frac{1}{|D_B|} \sum_{\vec{x} \in D} \delta_{L(\vec{x}),0} \delta_{\rho(\mathbf{I}(\vec{x})),\, \alpha}, \tag{2}$$

where $|D_O| = \sum_{\vec{x} \in D} \delta_{L(\vec{x}),1}, |D_B| = \sum_{\vec{x} \in D} \delta_{L(\vec{x}),0}$ are the sizes of the object and background regions, δ is the Kronecker delta function, and α indicates the histogram bin. In this paper, the filter $\rho(\mathbf{I}(\vec{x}))$ is either the color or the gray-scale intensity.

The *edgelet features* $d_3(\mathbf{I})$ are also represented by attributed features $d_3(\mathbf{I}) = \{z_j^e\}$, where $z_j^e = (\vec{x}_j, \theta_j)$ with $\vec{x}_j$ the position of the edgelet and θ_i its orientation. The edgelets are obtained by applying the Canny edge detector.

The sparse features of the models—interest points and edgelets—will be organized in terms of triplets. For each triplet, we calculate an *invariant triplet vector* (ITV), which is a function $\vec{l}(\vec{x}_i, \theta_i, \vec{x}_j, \theta_j, \vec{x}_k, \theta_k)$ of the positions $\vec{x}_i$ and orientations θ_i of the three features that form it and is invariant to the position, scale, and orientation of the triplet—see Fig. 5. We note that previous authors have used triplets defined over feature points (without using orientation) to achieve similar invariance [29], [30].

4 POM-IP

In this section, we introduce the POM-IP. The terminology for the hidden states of the full POM is shown in Table 1.

The POM-IP is defined on sparse interest points $d_1(\mathbf{I}) = \{z_i\}$ and is almost identical to the probabilistic grammar Markov model (PGMM) described in [1], [2], see Fig. 6. The only difference is that we use an explicit pose variable G that is used to relate the different POMs and provide a key mechanism for knowledge propagation (G appeared in [2] but was integrated out in (9)). But, as we will show in the experimental section, POM-IP outperforms the PGMM due to details on the reimplementation (e.g., allowing a greater number of aspects).

The POM-IP is specified as a generative model $P(\{z_i\}|s, G, V)P(G)P(s)P(V)$ for generating interest points $\{z_i\}$. It generates IPs both for the object(s) and the background. It has hidden states s (the model aspect), G (the pose), and V (the assignment variable which relates the IPs generated by the model to the IPs detected in the image). Each *aspect* s consists of an ordered set of IPs $z_1, \dots, z_{n(s)}$ and corresponds to one configuration of the object. These IPs are organized into a set of $n(s) - 2$ cliques of triplets $(z_1, z_2, z_3), \dots, (z_{n(s)-2}, z_{n(s)-1}, z_{n(s)-2})$ (see Fig. 7). The background IPs $z_{n(s)+1}, \dots, z_{n(s)+b}$ are generated by a background process. G is the pose of the POM-IP and can be expressed as $G = (\vec{x}_c, \theta_c, S_c)$, where $\vec{x}_c, \theta_c, S_c$ are the center, rotation, and scale of the POM-IP. The assignment variable $V = \{i(a)\}$ indicates the correspondence between the index a of the IPs in the model and their labels i in the image. We impose the constraint that each IP in the model can correspond to at most one IP in the image (i.e., $\sum_i i(a) \leq 1$ for all $a \in \{1, \dots, n(s)\}$). Model IPs can be unobserved—i.e., $\sum_i i(a) = 0$—because of occlusion or failure of the feature detector. (We require that all IPs generated by the background model are always observed).

The term $P(\{z_i\}|s, G, V)$ specifies how to generate the IPs for the object (with aspect s) and the background. Ignoring unobserved points for the moment, we specify this distribution in exponential form as

$$\begin{aligned} &\log P(\{z_i\}|s, G, V) = \\ &\quad \vec{\lambda}^s \cdot \vec{\phi}(\{(\vec{x}_{i(a)}, \theta_{i(a)}, G) : a = 1, \dots, n(s)\}) \\ &\quad + \vec{\lambda}^{A,s} \cdot \vec{\phi}^D(\{A_{i(a)} : a = 1, \dots, n(s)\}) \\ &\quad + \lambda|B| + \vec{\lambda}^B \cdot \vec{\phi}^B(\{z_{i(b)} : b = n(s), \dots, n(s) + |B|\}) \\ &\quad + \log J(\{z_i\}; \vec{l}, G) - \log Z[\lambda]. \end{aligned} \tag{3}$$

TABLE 1
The Terminology Used to Describe the Hidden States h of the POMs

Notation	Meaning
$\{(\vec{x}, \theta, A) : i = 1, ..., N\}$	the interest points in the image
$\vec{x}_i$	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
$a = 1, ..., N_s$	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 distribution of features inside the object
q_B	the distribution of features outside the object
$\mathbf{I}$	the intensity image
L	a binary label field of the object

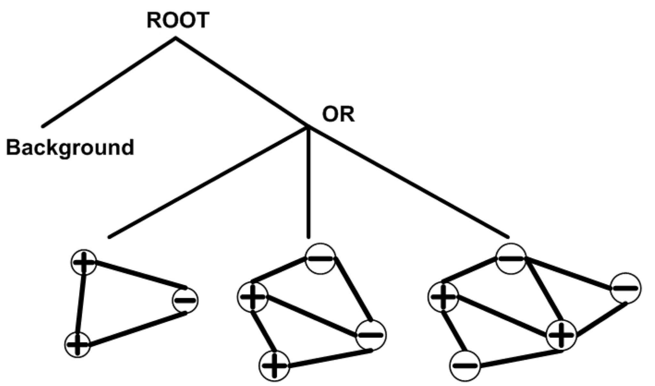

Fig. 6. Graphical illustration of POM-IP. This POM-IP has three aspects (mixture components), which are children of the OR node. Compare the first two aspects to the models in Fig. 2. Each aspect model is built out of triplets, see description in Section 4. There is also a background model to account for the interest points in the image that are not due to the object.

The first term on the right-hand side specifies the prior on the geometry of the POM-IP, which is given in terms of Gaussian distributions defined on the clique triplets. More precisely, it is expressed as $\vec{\lambda}^s \cdot \sum_{a=1}^{n(s)-2} \vec{\phi}(\vec{l}(z_a, z_{a+1}, z_{a+2}))$, where we define a Gaussian distribution over the ITV $\vec{l}(z_a, z_{a+1}, z_{a+2})$ for each clique triplet z_a, z_{a+1}, z_{a+2} and set the clique potential to be the sufficient statistics of the Gaussian (so the parameters $\vec{\lambda}^s$ specify the means and covariances of these Gaussians). The second term specifies the appearance model in terms of independent Gaussian distributions for the appearance of each IP. It is expressed as $\vec{\lambda}^{A,s} \cdot \vec{\phi}^D(\{A_{i(a)}: a = 1, \dots, n(s)\}) = \sum_{a=1}^{n(s)} \vec{\lambda}_a^{A,s} \cdot \vec{\phi}^D(A_{i(a)})$, where the potentials $\phi^D(A_{i(a)})$ are the sufficient statistics of the Gaussian distribution for the ath IP. The third and fourth terms specify the probability distribution for the number $|B|$ and appearance/positions/orientations of the background IPs, respectively. We assume that the positions and orientations of the background are uniformly distributed and the appearances are uncorrelated, so we can re-express $\vec{\lambda} \cdot \vec{\phi}^B(.)$ as $\sum_{b=n(s)}^{n(s)+|B|} \vec{\lambda}^B \cdot \vec{\phi}(z_{i(b)})$. The fifth term is a Jacobian factor $J(\{z_i\}; \vec{l}, G)$, which arises from the change of coordinates between the spatial positions and orientations of the IPs $\{\vec{x}_{i(a)}, \theta_{i(a)}\}$ in image coordinates and the ITVs $\vec{l}$ and the pose G used to specify the model. In [2], we argue that this Jacobian factor is approximately constant for the range of spatial variations of interest (alternatively, we can use the theory described in [31] to eliminate this factor by using a default model). The sixth, and final, term $Z[\lambda]$ normalizes the distribution. This term is straightforward to compute—provided we assume that the Jacobian factor is constant—since the distributions are either Gaussian (for the shape and appearance) or exponential (for the number of background IPs).

The distribution $P(s)$ is also of exponential form $P(s) = \frac{1}{Z[\lambda_s]} \exp\{\lambda_s \vec{\phi}(s)\}$. The distribution $P(G)$ is uniform. The distribution over V assumes that there is a probability ϵ that any object IP point is unobserved (i.e., $i(a) = 0$).

As described in [1], [2], there are three important computations we need do with this model: 1) *inference*, 2) *parameter learning*, and 3) *model evidence* for *model/structure induction*. The form of the model makes these computations practical by exploiting the graph structure of the model.

Inference requires estimating $(V^*, s^*, G^*) = \arg\max_{(V,s,G)} P(V, s, G|d_1(\mathbf{I}))$. To do this, for each aspect s, we perform dynamic programming to estimate V^* (exploiting the model structure) and G^*. Then we search over and maximize over s by exhaustive search (the number of aspects varies between 5 and 20). Two approximations are made during the process [1], [2]. 1) We perform an approximation, which enables us to estimate V^* by working with the ITVs $\vec{l}$ directly, and then later estimate the G^*. 2) If an IP is undetected (i.e., $i(a) = 0$), then we replace its unobserved values $z_i(a)$ by the best prediction from the observed values in its clique (observe that this will break down if two out of three IPs in a clique are unobserved, but this has not occurred in our experiments).

Parameter Learning requires estimating the model parameters λ from a set of training data $\{d_1(\mathbf{I}_\mu)\}$ by $\lambda^* = \arg\max_\lambda P(\{d_1(\mathbf{I})\}|s, G, V, \lambda)P(s|\lambda)P(V)$. This can be performed by the Expectation Maximization (EM) algorithm

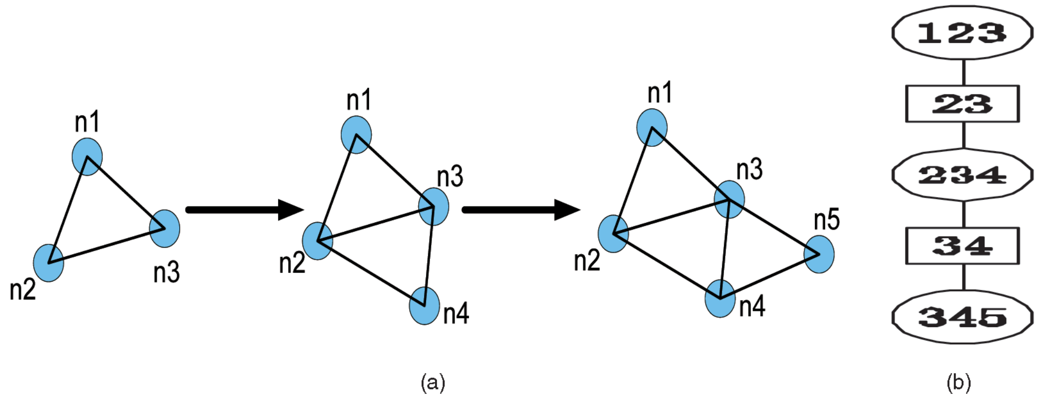

Fig. 7. The POM-IP uses triplets of nodes as building blocks. The structure is grown by adding new triangles. The POM-IP contains multiple aspects of similar form (not shown) and a default background model (not shown). Right panel shows the junction tree representation, which enables dynamic programming for inference.

in the free energy formulation [32] by introducing a probability distribution $Q(s,V)$ over the hidden states (s,V). (Good estimates for initializing EM are provided by the *dictionary*, see two paragraphs below.) The free energy is a function of $Q(.,.)$ and λ and the EM algorithm performs coordinate descent with respect to these two variables. The forms of the distribution ensure that the minimization with respect to $Q(.,.)$ can be performed analytically (with λ fixed) and the minimization with respect to λ can also be performed simply using dynamic programming (the summation form) to sum over the possible states of V and exploiting the quadratic (e.g., Gaussian) forms of the potentials. We make similar approximations to those made for inference [1], [2]: 1) work with the ITVs and eliminate G and 2) fill in the values of unobserved IPs by prediction from their clique neighbors.

Model Evidence is calculated to help model/structure induction by providing a fitness score for each model. We formulate it as calculating $\sum_{s,V,G} P(\{d_1(\mathbf{I})\}|s,G,V)P(s)$ $P(G)P(V)$ (i.e., we evaluate the performance of each model with fixed values of its model parameters λ). This requires the standard approximations: 1) work with the ITVs and eliminate G and 2) fill in unobserved IPs by the clique predictions.

Model/Structure Induction is performed by specifying a set of rules for how to construct the model out of elementary components. In PGMM [1], [2], the elementary components are triplets of IPs. To help the search over models/structures, we create a dictionary of triplets by clustering. More specifically, recall that for each triplet (z_1,z_2,z_3) of IPs, we can compute its spatial and appearance potentials $\phi(z_1,z_2,z_3)$ and $\phi^A(z_1,z_2,z_3)$. We scan over the images, compute these potentials for all neighboring triplets, and cluster them. For each cluster τ, we determine estimates of the parameters $\{\lambda_\tau, \lambda_\tau^A\}$. This specifies a *dictionary* of probabilistic triplets $\mathcal{D} = \{\lambda_c, \lambda_c^A\}$ (since the distributions are Gaussians, this will determine the mean state of the triplet and the covariances). The members of the dictionary are given a score to rank how well they explain the data. This dictionary is used in the following way. For model induction at each step, we have a default model (which is initialized to be pure background). Then we propose to grow the model by selecting a triplet from the dictionary (elements with high scores are chosen first) and either adding it to an existing aspect or by starting a new aspect. In both cases, we estimate the model parameters by the EM algorithm using initialization provided by the parameters of the default model and the parameters of the selected triplet. We adopt the new model if its model evidence is better than that of the default model. Then we proceed to select new triplets from the dictionary.

As shown in [2], the structure and parameters of the POM-IP can be learned with minimal supervision when the number of aspects is unknown and the pose (position, scale, and orientation) varies between images. Its performance on classification was comparable to other approaches evaluated on benchmarked data. Its inference was 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 performance on segmentation and fails to exploit all the image cues, as our experiments show in Section 7.

5 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 IPs). 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—Fig. 4. During inference, the POM-IP supplies estimates of pose and aspect to help estimate the POM-mask variables.

5.1 Overview of the POM-Mask

The probability distribution of the POM-mask is defined by

$$P(d_2(\mathbf{I})|L,\vec{q})P(L|G,s)P(\vec{q})P(s)P(G), \tag{4}$$

where $\mathbf{I}$ is the intensity image, $d_2(\mathbf{I})$ are the regional features—see Section 3. L is a binary valued labeling field $\{L(\vec{x})\}$ indicating which pixels $\vec{x}$ belong inside $L(\vec{x})=1$ and outside $L(\vec{x})=0$ the object, $\vec{q}=(q_O,q_B)$ are distributions on the image statistics inside and outside the object. $P(d_2(I)|L,\vec{q})$ is the model for generating the data when the labels L and distributions $\vec{q}$ are known.

The distribution $P(L|G,s)$ defines a prior probability on the shape L of the object, which is conditioned on the aspect s and pose G of the object. It is specified in terms of model parameters $\lambda_2 = \{M(s)(\vec{x})\}, \vec{u}(s)$, where $M(s)(\vec{x}) \in [0,1]$ is a *probability mask* (the probability that pixel $\vec{x}$ is inside the object) and $\vec{u}(s)$ is the vector between the center of the mask and center of the interest points (as specified by G). Intuitively, the probability mask is scaled, rotated, and translated by a transform $T(G,\vec{u}(s),s)$, which depends on $G,\vec{u}(s)$ and s. Estimates of G,s are provided to the POM-mask by POM-IP for both inference and learning—otherwise, we would be faced with the challenge of searching over G,s in addition to $L,\vec{q}$ and the model parameters $M(s),\vec{u}(s)$.

The prior $P(\vec{q})$ is set to be the uniform distribution (i.e., an improper prior) 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.

The *inference* for the POM-mask estimates

$$\vec{q}^*, L^* = \arg\max_{\vec{q},L} P(d_2(\mathbf{I})|L,\vec{q})P(L|G^*,s^*), \tag{5}$$

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 [20], [21], [23] described in detail in Section 5.2. 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 $T(G^*,\vec{u}(s^*),s^*)M^{s^*}$.

Learning the POM-mask is also performed with knowledge propagated from the POM-IP. The main parameter to be learned is the prior probability of the shape, which we represent by a *probability mask*. Given a set of images $\{d_2(\mathbf{I}_\mu)\}$, we seek to find the probability masks $\{M(s)\}$ and the displacements $\{\vec{u}(s)\}$. Ideally, we should sum over the hidden states $\{L_\mu\}$ and $\{\vec{q}_\mu\}$, but this is impractical, so we maximize over them. Hence, we estimate $\{M(s)\}, \{\vec{u}(s)\}$, $\{L_\mu\}, \{\vec{q}_\mu\}$ by maximizing $\prod_\mu P(d_2(\mathbf{I}_\mu)|L_\mu, \vec{q}_\mu) P(L_\mu|G^*, u(s^*_\mu))$, where $\{s^*_\mu, G^*_\mu\}$ are estimated by POM-IP for image I_μ. This is performed by maximizing with respect to $\{L_\mu\}, \{q_\mu\}$ and $\{M(s)\}, \{\vec{u}(s)\}$ alternatively, which combines grab-cut with steps to estimate $\{M_\mu(s)\}, \{\vec{u}(s)\}$, see Section 5.3.

5.2 POM-Mask Model Details

The distribution $P(d_2(\mathbf{I})|L, \vec{q})$ is of form

$$\frac{1}{Z[L,\vec{q}]} \exp\left\{ \sum_{\vec{x}\in D} \phi_1(\rho(\mathbf{I}(\vec{x}))|L(\vec{x}), \vec{q}) + \sum_{\vec{x},\vec{y}\in Nbh(\vec{x})} \phi_2(\mathbf{I}(\vec{x}), \mathbf{I}(\vec{y})|L(\vec{x}), L(\vec{y})) \right\}, \tag{6}$$

where $\vec{x}$ is the index of image pixel, $\vec{y}$ is a neighboring pixel of $\vec{x}$, and $Z[L, q]$ is the normalizing constant. This model gives a trade-off between local (pixel) appearance specified by the unary terms and binary terms which bias neighboring pixels to have the same labels unless they are separated by a large intensity gradient. The terms are described as follows.

The unary potential terms generate the appearance of the object as specified by the regional features, see Section 3, and are given by

$$\phi_1(\rho(\mathbf{I}(\vec{x}))|L(\vec{x}), \vec{q}) = \begin{cases} \log q_O(\rho(\mathbf{I}(\vec{x}))), & \text{if } L(\vec{x}) = 1, \\ \log q_B(\rho(\mathbf{I}(\vec{x}))), & \text{if } L(\vec{x}) = 0. \end{cases} \tag{7}$$

The binary potential $\phi_2(I(\vec{x}), I(\vec{y})|L(\vec{x}), L(\vec{y}))$ is an edge contrast term [24] and makes edges more likely at places, where there is a big intensity gradient:

$$\phi_2(I(\vec{x}), I(\vec{y})|L(\vec{x}), L(\vec{y})) = \begin{cases} \gamma(\mathbf{I}(\vec{x}), \mathbf{I}(\vec{y}), \vec{x}, \vec{y}), & \text{if } L(\vec{x}) \neq L(\vec{y}), \\ 0, & \text{if } L(\vec{x}) = L(\vec{y}), \end{cases} \tag{8}$$

where $\gamma(\mathbf{I}(\vec{x}), \mathbf{I}(\vec{y}), \vec{x}, \vec{y}) = \lambda \exp\{-\frac{g^2(\mathbf{I}(\vec{x}),\mathbf{I}(\vec{y}))}{2\gamma^2}\} \frac{1}{dist(\vec{x},\vec{y})}$, $g(.,.)$ is a distance measure on the intensities/colors $\mathbf{I}(\vec{x}), \mathbf{I}(\vec{y})$, γ is a constant, and $dist(\vec{x}, \vec{y})$ measures the spatial distance between $\vec{x}$ and $\vec{y}$. For more details, see [20], [21].

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

$$P(L|G,s) = \frac{1}{Z[G,s]} \exp\left\{ \sum_{\vec{x}\in D} \psi_1(L(\vec{x}); G, s) + \sum_{\vec{x}\in D \vec{y}\in Nbh(\vec{x})} \psi_2(L(\vec{x}), L(\vec{y})|\zeta) \right\}. \tag{9}$$

The unary potentials correspond to a shape prior, or probabilistic mask, for the presence of the object while the binary term encourages neighboring pixels to have similar labels. The binary terms are particularly useful at the start of the learning process because the probability mask is very inaccurate at first. As learning proceeds, the unary term becomes more important.

The unary potential $\psi_1(L(\vec{x}); G, s)$ encodes a shape prior of form

$$\psi_1(L(\vec{x}); G, s) = L(\vec{x}) \log(T(G, \vec{u}, s) M(\vec{x}, s)) + (1 - L(\vec{x})) \log(1 - T(G, u, s) M(\vec{x}, s)), \tag{10}$$

which is a function of parameters $M(\vec{x}, s), \vec{u}(s), T(G, \vec{u}, s)$, which need to be learned. Here, $M(\vec{x}, s) \in [0, 1]$ is a probabilistic mask for the shape of the object for each aspect $s.T(G, \vec{u}, s)$ transforms the probabilistic mask—translating, rotating, and scaling it—by an amount that depends on the pose G with a displacement $\vec{u}(s)$ (to adjust between the center of the mask and the center of the interest points). In summary, $T(G, \vec{u}(s), s)M(\vec{u}(s), s)(\vec{x})$ is the approximate prior probability that pixel $\vec{x}$ is inside the object (with aspect s) if the object has pose G. The approximation becomes exact if the binary potential vanishes.

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

$$\psi_2(L(\vec{x}), L(\vec{y})|\zeta) = \begin{cases} 0, & \text{if } L(\vec{x}) \neq L(\vec{y}), \\ \zeta, & \text{if } L(\vec{x}) = L(\vec{y}), \end{cases} \tag{11}$$

where ζ is a fixed parameter.

5.3 POM-Mask Inference and Learning Details

Inference for the POM-mask requires estimating

$$\vec{q}^*, L^* = \arg\max_{\vec{q}, L} P(d_2(\mathbf{I})|L, \vec{q}) P(L|G^*, s^*), \tag{12}$$

where G^* and s^* are provided by POM-IP.

Initialization of L is provided by the thresholded transformed probability mask $T(G^*, \vec{u}(s^*), s^*) M(\vec{x}, s^*)$ (after the probabilistic mask $M(.,.)$ has been learned) and by the bounding box of the interest points provided by POM-IP (before the probabilistic mask has been learned).

We perform inference by maximizing with respect to $\vec{q}$ and L alternatively. Formally,

$$\begin{aligned} \vec{q}^{t+1} &= \arg\max_q P(d_2(\mathbf{I})|L^t, \vec{q}^t) : \\ &\quad \text{which gives } q_O^{t+1}(\alpha) = f_O(\alpha, L^t), \\ &\quad q_B^{t+1}(\alpha) = f_B(\alpha, L^t) \\ L^{t+1} &= \arg\max_L P(d_2(\mathbf{I})|L^t, \vec{q}^t) P(L|G^*, s^*). \end{aligned} \tag{13}$$

The estimation of $\vec{q}^{t+1}$ only requires computing the histograms of the regional features inside and outside the current estimated position of the object (specified by $L^t(\vec{x})$). The estimation of L^{t+1} is performed by max-flow [21]. This is similar to grab-cut [20], [21], [23] except that: 1) Our initialization is performed automatically and 2) 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 $\{M(\vec{x}, s)\}$ and the displacement $\vec{u}(s)$. In principle, we should integrate out the hidden variables $\{L_\mu(\vec{x})\}$ and the distributions $\{\vec{q}_\mu\}$. But this is computationally impractical, so we estimate them also. This reduces to maximizing the following quantity with respect to $\{M(\vec{x}, s)\}, \vec{u}(s)$, $\{L_\mu(\vec{x})\}, \{\vec{q}_\mu\}$:

$$\prod_\mu P(d_2(\mathbf{I}_\mu)|L_\mu, \vec{q}_\mu)P(L_\mu|G^*_\mu, s^*_\mu), \tag{14}$$

where $\{s^*_\mu, G^*_\mu\}$ are estimated by POM-IP.

This is performed by maximizing with respect to $\{M(\vec{x}, s)\}$,$\vec{u}(s)$, $\{L_\mu(\vec{x})\}$, and $\{\vec{q}_\mu\}$ alternatively. The maximization with respect to $\{L_\mu(\vec{x})\}$ and $\{q_\mu\}$ is given in (13) and performed for every image $\{I_\mu\}$ in the training data set using the current values $\{M^t(\vec{x}, s)\}, \vec{u}^t(s)$ for the probability masks and the displacement vectors.

The maximization with respect to $\{M(\vec{x}, s)\}$ corresponds to estimating:

$$\{M^t(\vec{x}, s^*)\} = \arg\max \prod_\mu P(d_2(\mathbf{I}_\mu)|L^t_\mu, \vec{q}^t_\mu)P(L^t_\mu|G^*_\mu, s^*_\mu), \tag{15}$$

where $P(L^t_\mu|G^*_\mu, s^*_\mu)$ is computed from (13) using the current estimates of $\{M(\vec{x}, s^*)\}$ and $\vec{u}(s^*)$.

This can be approximated (this is exact if the binary potentials vanish) by

$$M^t(\vec{x}, s) = \frac{\sum_\mu \delta_{s^*_\mu, s} T(G^*_\mu, \vec{u}(s^*_\mu), s^*_\mu)^{-1} L^t_\mu(\vec{x})}{\sum_\mu \delta_{s^*_\mu, s}}, \tag{16}$$

where δ is the Kronecker delta function. Hence, the estimate for $M^t(\vec{x}, s)$ is simply the average of the estimated labels $L^t_\mu(\vec{x})$ for those images μ that are assigned (by POM-IP) to aspect s, where the pose of these labels has been transformed $T(G^*_\mu, \vec{u}(s^*_\mu), s^*_\mu)^{-1} L^t_\mu(\vec{x})$ by the estimated pose $L^t_\mu(\vec{x})$. Note that we use $T(G, \vec{u}(s), s)$ to transform the probability mask M to the label L, so $T(G, u(s), s)^{-1}$ is used to transform L to M.

The maximization with respect to $\vec{u}(s)$ can be approximated by $\vec{u}(s)^{t+1} = \vec{k}(L^t, G^*, s^*)$, where $\vec{k}(L^t, G^*, s^*)$ 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 Section 7). In addition, it provides context for the POM-edgelets. But note that the POM-mask needs the POM-IP to initialize it and provide estimates of the aspect s and pose G.

6 The POM-Edgelet Models

The *POM-edgelet distribution* is of the same form as POM-IP but does not include attributes A (i.e., the edgelets are specified only by their position and orientation). The data $d_3(\mathbf{I})$ are 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 learned, we can use it to teach POM-edgelets that are defined on subregions of the shape (adjusted for our estimates of pose and aspect). Formally, the POM-mask provides a mask L^*, which is decomposed into nonoverlapping subregions (3 by 3) $L^* = \bigcup_{i=1}^{9} L^*_i$, where $L^*_i \bigcap L^*_j = 0$ for $i \neq j$. There are nine 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 number 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 that we do not have appearance attributes and the subregion, where the edgelets appear, is fixed to a small part of the image (i.e., the estimate of the shape of the subregion).

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).

7 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 the 26 classes from Caltech 101 [33], which have at least 80 examples (the POMs require sufficient data to enable us to learn them). In all experiments, we learned 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*. 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 that 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 matching the IPs and edgelets. 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. First, the performance of the full POM for classification is better than POM-IP (because of the extra information provided by the POM-edgelets). Second, the full POM provides significantly better segmentation than the POM-IP (due to POM-mask). Third, 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 are invariant to scale, position, orientation, and aspect of the object. Finally, we also show that POM-IP—our reimplementation of the original PGMM [2]—performs better than PGMM due to slight changes in the reimplementation and a different stopping criterion, which enables the POM-IPs to have more aspects.

7.1 The Tasks

We tested on three tasks. 1) The *classification* task is to determine whether the image contains the object or is simply background. This is measured by the classification accuracy. 2) The *segmentation* task is evaluated by *precision and recall*. The precision $|\mathrm{R} \cap \mathrm{GT}|/|\mathrm{R}|$ is the proportion of pixels in the estimated shape region R that are in the

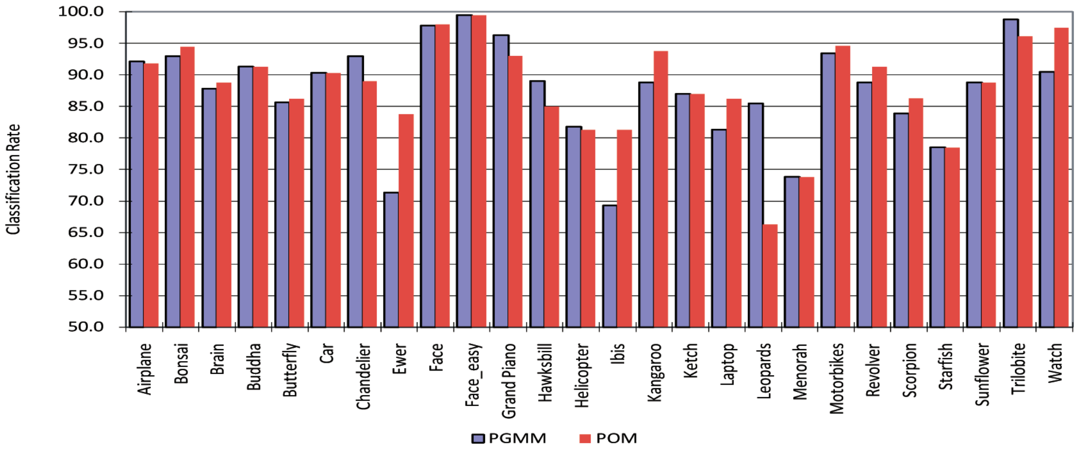

Fig. 8. We report the classification performance for the 26 object classes, which have at least 80 images. The average classification rate of POM-IP (PGMM) is 86.2 percent. The average classification rate of POMs is 88.6 percent.

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. 3) The *recognition* task which we illustrate by showing matches.

We performed these tests for three scenarios. 1) The *single object category* is when the training and testing images contain an instance of the object with unknown background. Due to the nature of the data sets we used, there is little variation in orientation and scaling of the object, so the invariance of our learning and inference was not tested. 2) The *single object category with variation* is where we had manipulated the training and testing data to ensure significant variations in object orientation and scale. 3) The *hybrid object category* is where the training and testing images contain an instance of one of the three objects (face, motorbike, or airplane).

7.2 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 Fig. 8, 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 because the regional features from POM-mask supply no information for object classification due to the weakness of the appearance model (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). We give comparisons to the results reported in [3], [14], [1] in Table 2.

7.3 Scenario 2: Segmentation for Single Object Category

Observe that *segmentation* (see Table 3) 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 initialize a grab-cut algorithm. But the best performance occurs when we use the edgelets. We also compare our method with that of [15] for segmentation. See the comparison in Table 4.

7.4 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 Fig. 9, where we show examples for each object category (see Fig. 8 and Table 3). 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, and 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, watch, and windsor chair) sometimes capturing large parts of the background while missing significant parts of the object (e.g., windsor chair). The fourth column shows that the POM-mask learns good shape priors (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 learned 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 shows the results of using the full POM model to segment these objects

TABLE 2
Comparisons of Classification with Results Reported in [3], [14], [1]

Dataset	full POM	[1]	[3]	[14]
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

TABLE 3
The Segmentation Performance Precision/Recall for 26 Objects Classes Which Contain At Least 80 Images

Dataset	PGMM[1]	POM-IP	POM-IP + POM-Mask	full POM
Airplane	44.0 / 62.5	61.4 / 75.9	73.9 / 75.1	75.2 / 75.4
Bonsai	71.2 / 37.5	77.5 / 54.0	78.3 / 53.6	78.6 / 53.4
Brain	84.0 / 39.1	94.1 / 60.9	97.7 / 68.9	97.7 / 69.0
Buddha	70.2 / 64.5	76.0 / 85.4	78.4 / 84.2	80.9 / 83.4
Butterfly	72.1 / 45.7	85.9 / 72.2	85.2 / 74.0	85.5 / 74.7
Car	31.1 / 89.6	28.0 / 61.6	52.0 / 50.7	50.0 / 54.3
Chandelier	73.3 / 48.5	82.4 / 54.6	83.4 / 50.8	83.4 / 50.9
Ewer	77.4 / 49.0	91.2 / 62.0	94.1 / 58.1	94.2 / 58.4
Face	86.8 / 64.4	72.6 / 87.0	72.2 / 89.3	73.5 / 89.6
Face easy	91.8 / 65.4	76.6 / 87.9	76.2 / 91.8	77.5 / 92.3
Grand Piano	73.1 / 54.5	86.2 / 61.5	88.0 / 76.8	87.8 / 81.3
Hawksbill	54.3 / 57.4	66.1 / 71.8	69.8 / 64.5	70.5 / 64.3
Helicopter	44.5 / 62.7	51.7 / 57.0	57.1 / 56.4	58.0 / 54.5
Ibis	38.8 / 63.5	60.3 / 68.7	60.9 / 66.6	61.2 / 66.7
Kangaroo	53.7 / 53.3	69.3 / 60.9	65.1 / 58.7	65.6 / 58.6
Ketch	63.0 / 63.9	67.9 / 69.7	67.1 / 72.5	69.8 / 71.0
Laptop	78.8 / 33.2	89.5 / 54.2	91.1 / 48.3	90.1 / 47.8
Leopards	37.0 / 71.7	55.9 / 56.2	55.9 / 56.2	55.9 / 56.2
Menorah	62.6 / 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
Revolver	49.5 / 58.1	75.3 / 72.6	82.8 / 63.9	82.7 / 62.0
Scorpion	47.7 / 48.8	71.0 / 63.8	69.1 / 54.7	68.7 / 54.3
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
Trilobite	66.8 / 50.7	67.5 / 68.3	71.3 / 74.8	71.3 / 74.9
Watch	82.2 / 64.4	94.0 / 63.4	94.9 / 63.9	95.4 / 69.2
Average	**67.9 / 58.4**	**73.5 / 66.5**	**76.6 / 65.8**	**76.9 / 67.4**

(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.

We show some failure modes in Fig. 10. These objects —Leopard and Chandelier—are not best suited for the approach in this paper for the following reasons: 1) Rigid mask (or masks) is (are) not the best way to model the spatial variability of deformable objects like leopards, 2) the texture of leopards and background are often fairly similar, which makes POM-mask not very effective (without using more advanced texture cues), and 3) the shapes of Chandeliers are not well modeled by a fixed mask and it has few reliable regional cues.

TABLE 4
Segmentation Comparison with Cao and Feifei [15]

	POM	Cao and Feifei[15]
Faces easy	86.0%	78.0%
Leopards	71.0%	57.0%
Motorbikes	79.0%	77.0%
Bonsai	76.3%	69.0%
Brain	82.1%	71.0%
Butterfly	85.5%	64.0%
Ewer	79.8%	68.0%
Grand Piano	84.8%	78.0%
Kangaroo	79.1%	63.0%
Laptop	71.0%	63.0%
Starfish	85.9%	69.0%
Sunflower	86.2%	86.0%
Watch	75.5%	60.0%

The measure of segmentation accuracy in pixels is used.

7.5 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 learned the full POM for a data set of faces, where we scaled, translated, and rotated the objects, see Fig. 11. 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 the scale and rotation. The results, see Tables 5 and 6, show only slight degradation in performance for the tasks.

7.6 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 one). The full POM will be able to successfully learn a hybrid

Fig. 9. The rows show the 14 objects that we used. The seven columns are labeled left to right as follows: (a) original image, (b) the bounding box specified by POM-IP, (c) the GraphCut segmentation with the features estimating using the bounding box, (d) the probability object-mask with the edgelets (green means features within the object, red means on the boundary), (e) the thresholded probability mask, (f) the new segmentation using the probability object-mask (i.e., POM-IP + POM-mask), and (g) the parsed result.

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 Fig. 12. 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 Tables 7 and 8. We see that the performance degrades very little, despite the fact that we are giving the system even less supervision. The confusion matrix between faces, motobikes, and airplanes is shown in Table 9. Our result is slightly worse than the one in [14].

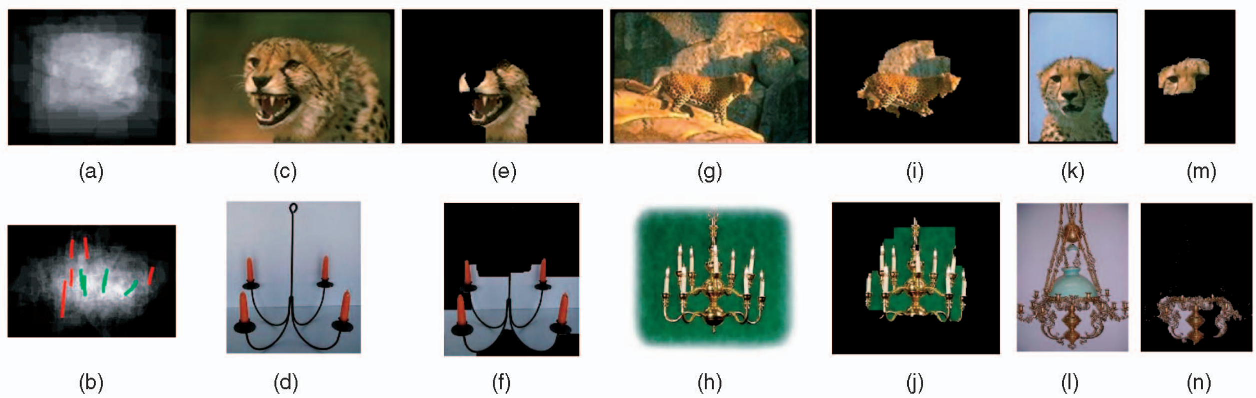

Fig. 10. Failure modes. Panel (a) the leopard mask. Panels (c), (g), (k): Input images of leopards. Panels (e), (i), (m): The segmentations output by POMs are of poor quality—parts of the leopard are missed in (e) and (m) and the segmentation includes a large background region in (g). We note that segmentation is particularly difficult for leopards because their texture is similar to the background in many images. Panel (b): The chandelier mask. Panels (d), (h), (l): Example images of chandeliers. Panels (f), (j), (n): The segmentations output by POMs. Chandeliers are not well suited to our approach because they are thin and sparse, so the regional cues, used in the POM-mask, are not very effective (geometric cues might be better).

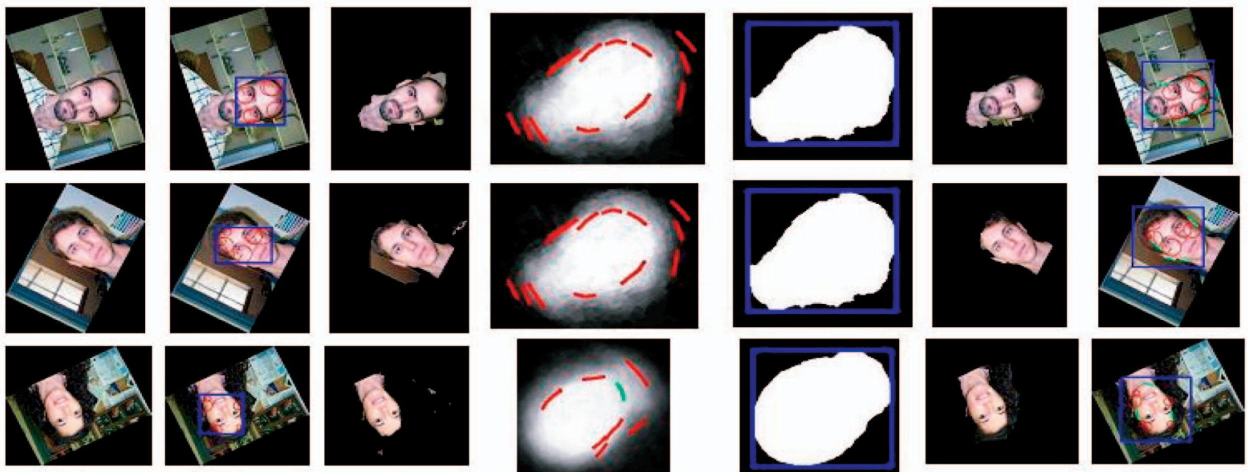

Fig. 11. The full POM can be learned even when the training images are randomly translated, scaled, and rotated.

7.7 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. Fig. 13 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 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 Fig. 14.

8 Discussion

This paper is part of a research program, where the goal is to learn object models capable of performing all object-related visual tasks. In this paper, we built on previous work [1], [2], which used weak supervision to learn a PGMM that used interest point features and performed classification. Our extension is based on combining elementary POMs, which

TABLE 5
Classification Results with Variable Scale and Orientation

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

TABLE 6
Comparisons of Segmentation by Different POMs When Scale and Orientation Are Variable

Dataset	PGMM	POM-IP	POM-IP+Mask	full POM
Faces	86 / 64	72 / 87	72 / 89	73 / 89
Scaled	83 / 63	71 / 90	76 / 87	76 / 89
Rotated	80 / 61	62 / 90	70 / 88	70 / 90
Sca.+Rot.	81 / 57	63 / 84	68 / 85	68 / 87

The precision and recall measure is reported.

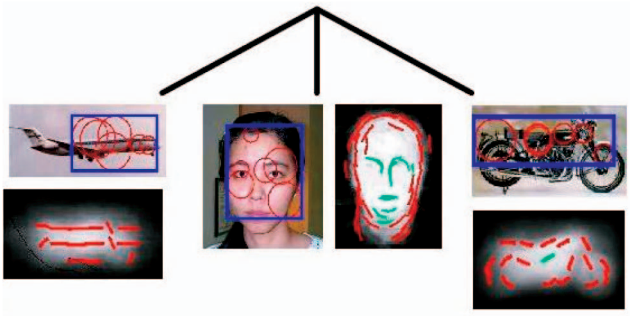

Fig. 12. Hybrid model. The training images consist of faces, motorbikes, and airplanes, but we do not know which type of object is in the image.

TABLE 7
The Classification Results for Hybrid Models

Dataset	full POM	PGMM[1]
Hybrid	87.8	84.6

TABLE 8
The Segmentation Results for Hybrid Models Using Different POMs

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

The precision and recall measure is reported.

TABLE 9
The Confusion Matrix for the Hybrid Model

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

The mean of the diagonal is 89.8 percent (i.e., classification accuracy), which is comparable with the 92.9 percent reported in [14].

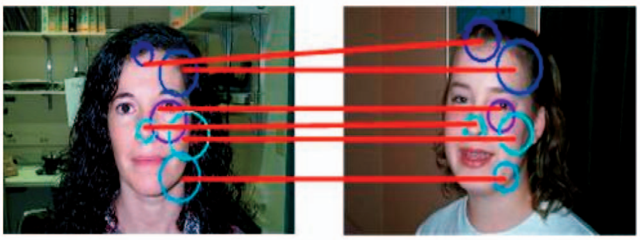

Fig. 13. An example of correspondence obtained by POM.

use different visual cues and can combine to perform a variety of visual tasks. The POMs 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 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 the objects are 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.

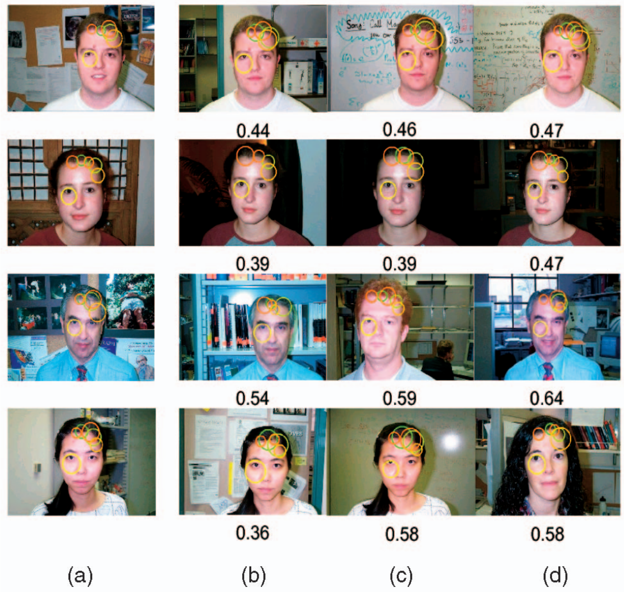

Fig. 14. Recognition examples. The first column is the prototype. The next three columns show the top three rankings. A distance to the prototype is shown under each image.

Acknowledgments

Long (Leo) Zhu and Alan Yuille were supported by the National Science Foundation (NSF) grants 0413214, 0736015, 0613563, and the W.M. Keck Foundation in performing this research. The authors thank Microsoft Research Asia for providing the internship to Yuanhao Chen to perform the research, and Iasonas Kokkinos, Zhuowen Tu, and Ying-Nian Wu for helpful feedback. Three anonymous reviewers gave detailed comments, which greatly improved the clarity of the paper.

References

[1] L. Zhu, Y. Chen, and A.L. Yuille, "Unsupervised Learning of a Probabilistic Grammar for Object Detection and Parsing," *Proc. Conf. Neural Information Processing Systems,* pp. 1617-1624, 2006.

[2] L. Zhu, Y. Chen, and A.L. Yuille, "Unsupervised Learning of Probabilistic Grammar-Markov Models for Object Categories," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 1, pp. 114-128, Jan. 2009.

[3] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 264-271, 2003.

[4] B. Leibe, A. Leonardis, and B. Schiele, "Combined Object Categorization and Segmentation with an Implicit Shape Model," *Proc. European Conf. Computer Vision Workshop Statistical Learning in Computer Vision,* pp. 17-32, May 2004.

[5] R. Fergus, P. Perona, and A. Zisserman, "A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 380-387, 2005.

[6] D.J. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial Priors for Part-Based Recognition Using Statistical Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 10-17, 2005.

[7] D.J. Crandall and D.P. Huttenlocher, "Weakly Supervised Learning of Part-Based Spatial Models for Visual Object Recognition," *Proc. European Conf. Computer Vision,* vol. 1, pp. 16-29, 2006.

[8] A. Kushal, C. Schmid, and J. Ponce, "Flexible Object Models for Category-Level 3d Object Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* 2007.

[9] G. Bouchard and B. Triggs, "Hierarchical Part-Based Visual Object Categorization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 710-715, 2005.

[10] E. Borenstein and S. Ullman, "Learning to Segment," *Proc. European Conf. Computer Vision,* vol. 3, pp. 315-328, 2004.

[11] A. Levin and Y. Weiss, "Learning to Combine Bottom-Up and Top-Down Segmentation," *Proc. European Conf. Computer Vision,* vol. 4, pp. 581-594, 2006.

[12] X. Ren, C. Fowlkes, and J. Malik, "Cue Integration for Figure/Ground Labeling," *Proc. Conf. Neural Information Processing Systems,* 2005.

[13] J.M. Winn and N. Jojic, "Locus: Learning Object Classes with Unsupervised Segmentation," *Proc. Int'l Conf. Computer Vision,* pp. 756-763, 2005.

[14] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman, "Discovering Objects and Their Localization in Images," *Proc. Int'l Conf. Computer Vision,* pp. 370-377, 2005.

[15] L. Cao and L. Fei-Fei, "Spatially Coherent Latent Topic Model for Concurrent Object Segmentation and Classification," *Proc. Int'l Conf. Computer Vision,* 2007.

[16] U. Grenander, *Pattern Synthesis: Lectures in Pattern Theory,* vol. 1. Springer, 1976.

[17] U. Grenander, *Pattern Analysis: Lectures in Pattern Theory,* vol. 2. Springer, 1978.

[18] Y. Chen, L. Zhu, A.L. Yuille, and H. Zhang, "Unsupervised Learning of Probabilistic Object Models (POMs) for Object Classification, Segmentation and Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[19] N. Friedman and D. Koller, "Being Bayesian about Bayesian Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks," *Machine Learning,* vol. 50, nos. 1/2, pp. 95-125, 2003.

[20] A. Blake, C. Rother, M. Brown, P. Pérez, and P.H.S. Torr, "Interactive Image Segmentation Using an Adaptive gmmrf Model," *Proc. European Conf. Computer Vision,* vol. 1, pp. 428-441, 2004.

[21] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," *Proc. Int'l Conf. Computer Vision,* pp. 105-112, 2001.

[22] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *Proc. Int'l Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition,* pp. 359-374, 2001.

[23] C. Rother, V. Kolmogorov, and A. Blake, "'Grabcut': Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 309-314, 2004.

[24] M.P. Kumar, P.H.S. Torr, and A. Zisserman, "Obj Cut," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 18-25, 2005.

[25] N. Jojic, J.M. Winn, and L. Zitnick, "Escaping Local Minima through Hierarchical Model Selection: Automatic Object Discovery, Segmentation, and Tracking in Video," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 117-124, 2006.

[26] B. Frey and N. Jojic, "Transformation-Invariant Clustering Using the em Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 1, pp. 1-17, Jan. 2003.

[27] T. Kadir and M. Brady, "Saliency, Scale and Image Description," *Int'l J. Computer Vision,* vol. 45, no. 2, pp. 83-105, 2001.

[28] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[29] Y. Amit and D. Geman, "A Computational Model for Visual Selection," *Neural Computation,* vol. 11, no. 7, pp. 1691-1715, 1999.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "A Sparse Texture Representation Using Local Affine Regions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 8, pp. 1265-1278, Aug. 2005.

[31] Y. Wu, Z. Si, C. Fleming, and S. Zhu, "Deformable Template as Active Basis," *Proc. Int'l Conf. Computer Vision,* 2007.

[32] R.M. Neal and G.E. Hinton, "A View of the em Algorithm that Justifies Incremental, Sparse, and Other Variants," *Learning in Graphical Models,* pp. 355-368, MIT Press, 1999.

[33] 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," *Computer Vision and Image Understanding,* vol. 106, no. 1, pp. 59-70, 2007.

Yuanhao Chen received the BS degree from the Department of Automation, University of Science and Technology of China in 2003. He is currently working toward the PhD degree at the University of Science and Technology of China. He is also a student in the joint PhD program of Microsoft Research Asia and the University of Science and Technology of China. His research interests include computer vision, machine learning, and information retrieval. He is a member of the IEEE.

Long (Leo) Zhu received the BS degree in computer science from Northeastern University, China, in 2001, and the PhD degree in statistics from the University of California, Los Angeles in 2008. He is a postdoctoral associate at the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. His research interests include computer vision and machine learning. He is a member of the IEEE.

Alan Yuille received the BA degree in mathematics from the University of Cambridge in 1976 and the PhD degree on theoretical physics, supervised by Professor S.W. Hawking, in 1981. He was an NATO postdoctoral research fellow studying physics at the University of Texas Austin and the Institute for Theoretical Physics at the University of California Santa Barbara in 1981/1982. He was a research scientist at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology and the Division of Applied Sciences at Harvard University from 1982 to 1988. He served as an assistant and associate professor at Harvard University until 1996. He was a senior research scientist at the Smith-Kettlewell Eye Research Institute from 1996 to 2002. In 2002, he joined the University of California Los Angeles as a full professor with a joint appointment in statistics and psychology. He received a joint appointment in computer science in 2007. His research interests include computational models of vision, mathematical models of cognition, and artificial intelligence and neural networks. He is a fellow of the IEEE.

Hongjiang Zhang received the BS and PhD degrees in electrical engineering from Zhengzhou University, Henan, China, in 1982, and the Technical University of Denmark, Lyngby, in 1991, respectively. In 1999, he joined Microsoft Research, where he is currently the managing director of the Advanced Technology Center in Beijing. From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. From 1995 to 1999, he was a research manager at Hewlett-Packard Labs, Palo Alto, California, where he was responsible for research and development in the areas of multimedia management and intelligent image processing. He has coauthored/coedited four books, more than 350 papers and book chapters, numerous special issues of international journals on image and video processing, content-based media retrieval, and computer vision, as well as more than 60 granted patents. He is a fellow of the IEEE and the ACM.