Unsupervised Segmentation of Objects using Efficient Learning

{Himanshu Arora, Nicolas Loeff}[*] David A. Forsyth, Narendra Ahuja
University of Illinois at Urbana-Champaign, Urbana, IL, 61801
{harora1,loeff,daf,n-ahuja}@uiuc.edu

Abstract

We describe an unsupervised method to segment objects detected in images using a novel variant of an interest point template, which is very efficient to train and evaluate. Once an object has been detected, our method segments an image using a Conditional Random Field (CRF) model. This model integrates image gradients, the location and scale of the object, the presence of object parts, and the tendency of these parts to have characteristic patterns of edges nearby. We enhance our method using multiple unsegmented images of objects to learn the parameters of the CRF, in an iterative conditional maximization framework. We show quantitative results on images of real scenes that demonstrate the accuracy of segmentation.

1. Introduction

Keypoint based object templates are efficient and accurate [6, 7, 14, 12], and can be used to localize objects. This paper presents a novel model that integrates a keypoint based template into a random field to achieve unsupervised object segmentation. We show that the information generated while learning the template is rich enough so that segmentation can be learned without supervision. In particular, coherence of edges within the frame of reference of a part in the template and across the training examples is a strong cue for segmentation.

Towards this, we extend the template based object model introduced in [14] to estimate scale. We introduce a new unsupervised edge model that aids in the segmentation of the object. This model encodes object boundaries in the local coordinate system of the parts in the template.

We integrate the template and image gradient information into a Conditional Random Field model. To the best of our knowledge, it is the first attempt to unite keypoint-based object models with random field segmentation, in an unsupervised setting. We learn the CRF using an iterative mechanism, alternating between estimating segmentation and parameters, imposing coherence among the unsegmented training images. The final detection-segmentation system is fast, and can tolerate large intra class variations. Both quantitative and qualitative results on real images demonstrate that this iterative procedure produces good segmentations.

Object dependent segmentation. There has been extensive work recently on model driven object segmentation in images. Kumar *et al.* [11] use a Bayesian approach to integrate a layer based object model with Markov Random Fields. While this method produces good segmentations, it is very demanding of training data, requiring a video for each category, and also cannot tolerate large intraclass variations. Winn *et al.* [21] use a Conditional Random Field based model to segment and handle multiple objects with occlusion. Training the model requires already segmented images. Leibe *et al.* [12] use a supervised part based model to resolve conflicting segmentation of objects in images. Their model uses only keypoint features and not image texture information. Levin and Weiss [13] present a supervised framework for learning bottom-up and top-down cues simultaneously. Borenstein *et al.* [2, 1] present a formulation to integrate top-down and bottom-up cues for unsupervised segmentation, using image fragments. Winn *et al.* [20] present a pixel based deformable object model that uses color cues for segmentation. The model in [16] presents an algorithm to segment and detect people, using a superpixel representation. Carbonetto *et al.* [4] integrate a bag of keypoints model with a CRF to output a per superpixel confidence map of object localization, but no segmentation of the object. Russell [17] *et al.* learn object classes as a topic model with superpixels as features. He *et al.* use a mixture of CRFs [9] to obtain a per superpixel labeling of object classes. Many of these works pose the final segmentation estimation problem as a graph cut energy minimization [3].

2. Object Model

We extend the generative template model of [14] to scale invariance and introduce new learning algorithms. An image is represented as a set of F features $\{f_j\}_{j=1}^F$, described by their location in the image (f_j^l) and appearance (f_j^a). The model generates the image by first choosing an object con-

[*] Both are first authors

1-4244-1180-7/07/$25.00 ©2007 IEEE

figuration $o_c = (o_l, o_s)$, where o_l is the object location, and o_s is the object scale. Conditioned on the object configuration, the model produces features independently. For each feature the model chooses a part $i \in \{1, \ldots, P\}$ with probability π_i. This conditional independence assumption makes inference very efficient in comparison to a fully connected constellation model [6]. The part i generates the feature appearance f_j^a from a multinomial distribution $p_i^A(f_j^a)$ with parameter $\overline{h}_i$ on a dictionary of patches, and location f_j^l from a Gaussian $p_i^L(f_j^l|o_c) \sim \mathcal{N}(\mu_i \cdot (o_s) - o_l, \Sigma_i \cdot (o_s)^2)$. The mean and variance of this distribution are corrected for the scale of the object. The variance term also compensates for the fact that smaller objects tend to produce lower number of feature detections. In addition, there is a background part with uniform distribution in location.

Let us denote by ω_{ij} the event that the feature f_j was generated by part i. This is a binary random variable. By the above discussion, we have $\sum_i \omega_{ij} = 1, \forall j$. We thus have $p(f_j|\omega_{ij} = 1, o_c) = p_i^L(f_j^l|o_c) p_i^A(f_j^a)$.

The features $\{f_j\}_1^F$ in an image make up the set v of observed variables. The set of parameters is $\Theta = (\{\mu_i, \Sigma_i\}_{i=1}^P, \{\overline{h}_i, \pi_i\}_{i=1}^{P+1})$. The object configuration, o_c, together with part-feature associations, ω_{ij}'s, are the set h of hidden random variables. Thus, the complete data likelihood is

$$P_\Theta^{obj}(h, v) = \prod_{j,i} \left\{ p_L^i(f_j^l|o_c) p_A^i(f_j^a) \pi_i \right\}^{[\omega_{ij}=1]} P(o_c) \quad (1)$$

where $[expr]$ is 1 if $expr$ is true, and $P(o_c) = P(o_l)P(o_s)$. We assume that o_l and o_s are *a priori* uniformly distributed across their respective domains.

Learning: In appendix A we introduce and analyze two variational approximations to learn this object model in an efficient way. One (H-EM) is a hybrid which uses Expectation Maximization to estimate the location distribution and point estimates to determine the scale of the object. The other (H-V) is a hybrid which uses a mean-field approximation for location instead.

Template Evaluation: To estimate the configuration of the object, we need to maximize the posterior likelihood $p(o_l, o_s|\{f_j\}, \Theta)$ with respect to (o_l, o_s).

3. Object Segmentation

The model described in the previous section can detect and estimate the location and scale of the object, as well as the assignment between features in the image and parts of the object. Given a set of unsegmented images, this information can help rectify them both globally (i.e. location and scale of object) and locally with respect to each part of the model. In this way, by matching several unsegmented images we can compensate for the lack of supervision; we can learn boundaries of object and segment (i.e. label regions as belonging to the object, or background). We approach the problem in an iterative fashion, first segmenting the objects based on the current parameter estimates, and then updating the parameters to match the segmentation in the whole stack of images.

Motivated by [16], we use a superpixel representation obtained by oversegmenting the image into $R = 50$ regions using the normalized cuts algorithm [18]. The superpixel representation not only is computationally advantageous due to small number of nodes, but also is perceptually relevant. In addition, computing features over superpixels allows for capturing more global phenomena when compared to interaction between only adjacent pixels. The main drawback of superpixels is that their boundaries may not align with the object boundaries.

For the object segmentation task, i.e. labeling superpixels in the image as belonging to object or background, we incorporate the following information:

1. Object location and scale estimates provided by the object model (M) to localize the object in the image,
2. Shape and extent of the object for which we introduce a) *Model Edge* response (m_E), which captures the locations of edges with respect to a local reference frame given by the model, and b) *Object/Part Overlap* (m_O), which captures the extent of overlap of each part in the model with the inside of the object. We explain these below in Sec. 3.1 and 3.2.
3. The location of image edges (I_E) with respect to the object edges estimated by the model.

To integrate all this information, we use a random field model. With each superpixel v in the image I, we associate a binary label l_v which takes a value of 1 if v lies inside the object and 0 if it is in the background. The random field model is then expressed for the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of all superpixels in the image and $\mathcal{E}$ is the set of all pairs of neighboring superpixels, as follows:

$$\log(P(\{l_v\}|I)) + \log Z(\lambda_M, \lambda_E, I_E) = \sum_{v \in \mathcal{V}} \underbrace{\phi(l_v|m_E, m_O)}_{\text{Model Unary}} + \sum_{\{v,u\} \in \mathcal{E}} \left(\lambda_M \underbrace{\psi_M(l_v, l_u|m_E)}_{\text{Model Edge Interaction}} + \lambda_E \underbrace{\psi_E(l_v, l_u|I_E)}_{\text{Image Edge Interaction}} \right). \quad (2)$$

where, $\phi(l_v|m_E, m_O)$ is a local estimate of the probability that v is in the object. Superpixels that belong to the object tend to include features assigned to the object m_O as well as model inferred edges m_E. The interaction terms tend to separate superpixels when both model inferred edges ψ_M and image edges ψ_E align. Z is the normalization term.

3.1. Model edge m_E

The template model in Sec. 2 only accounts for the local appearance of the object, and not its boundaries.

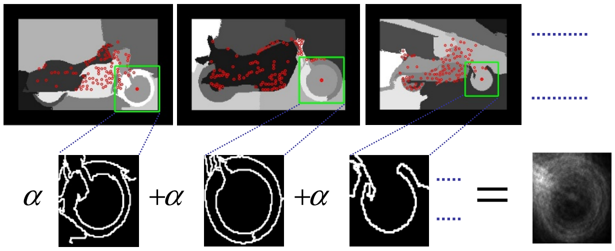

Figure 1. **Computation of per part mean edge response**. We obtain the dominant edges as the boundaries of normalized cut algorithm. **Top row** shows the detected feature locations overlaid on the normalized cut segmentation for the training images. **Bottom row** shows edge response computed **centered in the feature frame of reference**. Weighted average of these give the per part mean edge response as shown by the final figure. This procedure is described in section 3.1

Here we propose a novel mechanism to enhance the model for encoding object boundaries. Given the set of training images, we can infer the configuration of objects in them, and the assignments between features in the image and parts in the model. We expect that in the **local** coordinate frame of each model part, object edges will be consistent, while background edges will not. This suggests that a model for edge generation can be added to each part.

In addition to the local appearance of a feature, part i now also generates a local edge response, as a Gaussian with mean edge response $\mu e_i^{\boldsymbol{p}}$, for a point $\boldsymbol{p}$ centered at the feature location f_j^l. Assume that the observed dominant edge response for the n^{th} image is given by $e^n(\boldsymbol{p})$. Also assume that the parameters θ of the model have been learned, and object configuration $(\hat{o}_l, \hat{o}_s)$ has been inferred. Then, μe_i^p can be estimated as

$$\mu e_i^{\boldsymbol{p}} = \sum_{n,j} \alpha_{ij}^n e^n((\boldsymbol{p} - f_j^l)/\hat{o}_s), \;\; \alpha_{ij}^n \triangleq P(w_{ij}^n = 1|\hat{o}_l, \hat{o}_s, f_j^n) \quad (3)$$

i.e., an average of the feature centered edge responses scaled by the estimated scale $\hat{o}_s$, weighted by the posterior of part i generating that feature α_{ij}. The entire process is depicted in Fig. 1. We use normalized cuts segmentation with $R = 10$ regions as the image edge response during learning.

Given a new image, the (local) per feature inferred edges are aggregated over the entire image to produce a map of model edges on the image m_E. This edge generation process is illustrated in Fig. 2 . Note that this edge response is *independent* of the actual edges in the image. We use m_E to define both the unary and interaction terms in our model (Eq. (2)).

3.2. Overlap between features and object m_O

Each part in the object model has a position and extent in location described by its mean location and variance. Due to this extent, a part can generate features that lie completely inside the object, or on its periphery. For instance, consider an airplane model: window features usually lie completely inside the airplane, while wing, tail and specially landing gear features tend to overlap to a lesser extent.

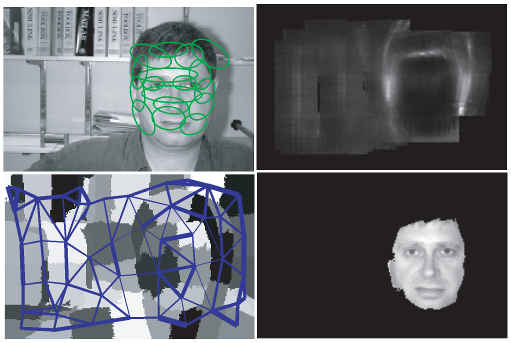

Figure 2. **Template based segmentation**. **Top left** image shows an original face image with the estimated configuration of the object using the keypoint model. **Top right** image shows the model based edge response computed using locally rectified template edges weighted by votes from image keypoints. Note that this **does not depend** on the **edges of the image** itself, which are encoded by the superpixels. **Bottom left** image shows the superpixel graph. The width of edges here represents the total interaction terms $(\lambda_M \psi_M(l_u, l_v|m_E) + \lambda_E \psi_E(l_u, l_v|I_E))$. The final segmentation is shown in **bottom right** image.

Thus for each part i, we introduce an object-part overlap parameter $q_i \in [0, 1]$, which denotes the fraction of extent of a feature lying inside the object, given that the feature is generated by part i. Note that this information is vital for using a keypoint template based model for segmentation, since the object model does not explicitly model each pixel's probability of lying inside the object. This is another novel contribution of this paper.

3.3. Unary potential ϕ

Using these parameters, we can compute a unary per pixel classifier by averaging over the observed features as

$$P(l_v|m_O) = \frac{1}{A_v}\sum_{p\in v}\frac{1}{N_p}\sum_{i,j}(q_i)^{l_v}(1-q_i)^{(1-l_v)}\alpha_{ij}\delta_{p\in f_j} \quad (4)$$

where, N_p is the total number of features which include pixel p (i.e. $\sum_j \delta_{p\in f_j}$), A_v is the number of pixels in superpixel v and α_{ij} is the posterior of part i given feature f_j and inferred location and scale of the object, as defined in Eq. (3). Pixels where no features exist are given equal probabilities of belonging to object and background.

Since consistent model edges should only lie on the boundary or inside that object, a model edge response in a superpixel should indicate the presence of an object. Considering this, we choose to estimate $P(l_v = 1|m_E)$ as the mean of the normalized object model edge response inside superpixel v.

We define the unary term ϕ in the random field (Eq. (2)) as a weighted average of these estimates as follows:

$$\phi(l_v|m_E, m_O) = \log(\gamma P(l_v|m_E) + (1-\gamma)P(l_v|m_O))$$

3.4. Interaction Terms ψ

The presence of a strong inferred model edge at the boundary between neighboring superpixels u and v indicates that they might have different labels. We compute the interaction potential between superpixels ψ_M in Eq. (2) as the inverse of the integral of model edge response across the boundary between u and v for $l_v \neq l_u$ and zero otherwise.

The image edge dependent term ψ_E is motivated by the fact that image edges should align with object boundaries. We thus want to penalize two neighboring superpixels having different labels if there is no image edge between them. We compute ψ_E as the normalized-cut cost (sum of intra region affinity as defined in [18] for both superpixels over inter region affinity between superpixels) for $l_v \neq l_u$ and zero otherwise.

3.5. Inference and parameter estimation

Since we do not use supervised training data with ground truth segmentations for training, $Q = \{q_i\}_1^P$ is not known a priori and needs to be estimated. We learn these parameters in an iterative fashion. We first segment all training images, labeling every superpixel as object or background. We then fix the labels and reestimate the parameters, pooling information from all training images simultaneously. The procedure stops when no segmentation of training images changes for one iteration.

Inference: Given the parameters, the globally optimal labeling that maximizes the log likelihood in Eq. (2) can be posed as a min-cut problem on the graph G [3]. The solution to this problem is very fast.

Parameter Estimation: Given the segmentation, we adopt a piecewise approximation [19] to the normalization term Z, which is a bound that decouples the unary and interaction potentials. Consider a set of N training images I^n with object model inferred m_E, m_O, and assumed labels $\{l_v^n\}_{v=1}^R$, where $n = 1 \ldots N$. Then the piecewise approximation is

$$\log Z \leq \log\left(\textstyle\sum_{\{l_v\}} \exp \sum_v \phi(l_v|m_E, m_O)\right) + \tag{5}$$
$$\log \textstyle\sum_{\{l_v\}} \exp \left\{ \sum_{\{u,v\}\in E} (\lambda_M \psi_M(l_u, l_v|m_E) + \lambda_E \psi_E(l_u, l_v|I_E)) \right\}$$

Since $\phi(\cdot)$ is a log probability, the first term in above expression is zero, and thus this upper bound is independent of Q. Using eq. 2 and 3.3, the parameter estimate becomes

$$Q \leftarrow \operatorname*{argmax}_{q_i} \sum_{n,v} \log \left(\gamma P(l_v^n|m_E) + (1-\gamma) \sum_i q_i \beta_{vi}^n \right) \tag{6}$$

where $\beta_{vi} = \frac{1}{A_v} \sum_{p\in v} \frac{1}{N_p} \sum_j \alpha_{ij} \delta_{p\in f_j}$.

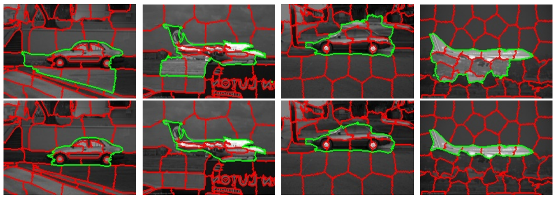

Figure 3. **Effect of overlap between parts and object**. (**Top** images): Before learning the overlap between parts and object (Q), object area is overestimated. This is because features on the boundary of the object result in regions close to the objects being assigned to it during segmentation. (**Bottom** images): After reestimation, the problem is diminished. All images are taken from **test** sets. The superpixel representation is shown in **red** and the final segmentation is shown in **green**. The brightness of the estimated background superpixels is diminished to aid visibility. This figure is most easily interpreted when viewed in color.

Lower bounded using Jensen's inequality; maximizing this gives $q_k \leftarrow \sum_{n,i} l_v^n \beta_{ik}^n / \sum_{n,i} \beta_{ik}^n$.

This iteration between the estimation of labels l_v^n (segmenting the images) and parameters Q can be seen as akin to learning with partially labeled data. The term $P(l_v|m_E)$, which does not depend on Q, biases the search toward meaningful local maxima (figure 3).

4. Experiments and Results

We test our algorithm on standard datasets [6]. Each category has an object image dataset and a background dataset. The scale-saliency keypoint detector [10] is used to extract F features from each image in the dataset. The SIFT [15] representations of features in the object images are clustered into representative vectors. The appearance of each feature is vector quantized using this dictionary. We train the hybrid-EM (H-EM) and hybrid-Variational (H-V) models using $P = 30$ parts and $N_x N_y = 238$. We initialized the appearance and location of the parts using randomly selected features in the training set. The stopping criterion used is the change in the free energy F_e (c.f. appendix A).

Template performance: The median equal error rate detection performance for 10 random test/train splits (50% training/50% testing) of the dataset for each category of both inference algorithms (*Airplanes*: H-EM: 96.6, H-V: 96.5, *Motorbikes*: H-EM: 97.1, H-V: 97.1, *Faces*: H-EM: 98.8, H-V: 98.7) is significantly better than the constellation model [6] and comparable to the best performing discriminative approach [5]. The H-V model has virtually the same performance as the H-EM, but is more than 50 times faster, converging in less than 5 minutes for hundreds of training images.

Qualitative segmentation results are shown in figure 4 on **gray-scale** images for the Caltech faces, airplanes, motorbikes, and Caltech 101 cars **test** sets. Images in these datasets have large intraclass variations in appearance.

	Faces			Airplanes			Cars			Motorbikes		
	Tot	**Obj**	**Bg**	**Tot**	**Obj**	**Bg**	**Tot**	**Obj**	**Bg**	**Tot**	**Obj**	**Bg**
Best	97.2	92.2	98.4	96.2	77.4	98.6	96.9	89.3	98.0	89.6	85.6	92.2
Algorithm	92.4	67.1	99.1	93.1	75.9	95.9	95.1	87.0	96.1	83.1	79.8	88.9

Table 1. **Segmentation performance** for different datasets. The measures used are percentage of total correctly classified pixels (**Tot**), percentage of object pixels classified as object (**Obj**), and percentage of background pixels classified as background (**Bg**). We compare the performance of segmentation achieved by the **CRF** against the **best** possible performance given the superpixels. To compute the best possible segmentation, a superpixel is assigned to the object if more than 50% of its area is inside the object. Even for an **unsupervised** approach, performance is close to the optimal, with the possible exception of the faces dataset. The reason is that due to high variability, hair is not learned to be included inside the object (see figure 4).

There are many images for which it is difficult to segment the gray-scale object without a model. We use $\gamma = 0.5$ and $\lambda_M = \lambda_E = 10/3$ for our experiments.

Quantitative segmentation results: Table 1 shows the performance of the algorithm on the test sets in terms of correctly labeled pixels after the CRF parameter learning. The performance measures are 1) Total performance, the percentage of correctly classified pixels, 2) Object performance, the percentage of object pixels classified as object, and 3) Background performance, the percentage of background pixels classified as background. For instance, for 100% object performance means that the estimated segmentation always encloses the ground truth segmentation, but the segmentation might overestimate the object. Hence this does not mean perfect segmentation. The results are compared with ideal (best) results, which are obtained by labeling all superpixels with $> 50\%$ overlap with the ground truth as object. Even for an unsupervised approach, performance is close to the optimal, with the possible exception of the faces dataset. Hair tends to change from subject to subject, and therefore the algorithm learned a low value for object/part overlap q_i for the hair parts (figure 4). As a result, face segmentations do not include hair for most images (and thus the background performance is better than the 'optimal'). This behavior can be expected in an unsupervised framework.

5. Conclusion

We present a novel framework for integrating keypoint-based template object models with segmentation, in an **unsupervised** fashion. We enhance a generative object model, and introduced very **efficient** learning algorithms. We extend the object model to produce boundary information based on keypoint features independently of the actual edges in a given image. We feed this all this information into a random field model and introduce an iterative parameter reestimation. The result is very **good segmentation** even for images with very low object-background contrast.

Future Work: One of the main drawback of the current model is that the information flows in one way from object model to segmentation. We are working on a new model for multiple object detection/segmentation where the object model proposes several candidate configuration and feeds the information into the random field, which solves **both** the detection and segmentation problems simultaneously.

References

[1] E. Borenstein, E. Sharon, and S. Ullman. Combining Top-Down and Bottom-Up Segmentation. In *Proc. of CVPR.*, pages 46–46, 2004.

[2] E. Borenstein and S. Ullman. Class-Specific, Top-Down Segmentation. In *Proc. of ECCV*, pages 109–124, 2002.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI*, 23(11):1222–1239, 2001.

[4] P. Carbonetto, G. Dorkó, C. Schmid, H. Kück, and N. de Freitas. A semi-supervised learning approach to object recognition with spatial integration of local features and segmentation cues. In *Toward Category-Level Object Recognition*, pages 277–300, 2006.

[5] G. Dorkó and C. Schmid. Object class recognition using discriminative local features. Technical Report RR-5497, INRIA, 2005.

[6] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proc. of CVPR*, 2003.

[7] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. of CVPR*, volume 1, pages 380–387, 2005.

[8] B. Frey and N. Jojic. A Comparison of Algorithms for Inference and Learning in Probabilistic Graphical Models. *IEEE Trans. PAMI*, 27(9):1392–1416, 2005.

[9] X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV (1)*, pages 338–351, 2006.

[10] T. Kadir and M. Brady. Scale saliency : A novel approach to salient feature and scale selection. *International Conference Visual Information Engineering*, pages 25–28, 2003.

[11] M. P. Kumar, P. Torr, and A. Zisserman. Objcut. In *Proc. of CVPR*, pages 18–25, 2005.

[12] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc. of the Workshop on Stat. Learning in Comp. Vision*, Prague, Czech Republic, May 2004.

[13] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV (4)*, pages 581–594, 2006.

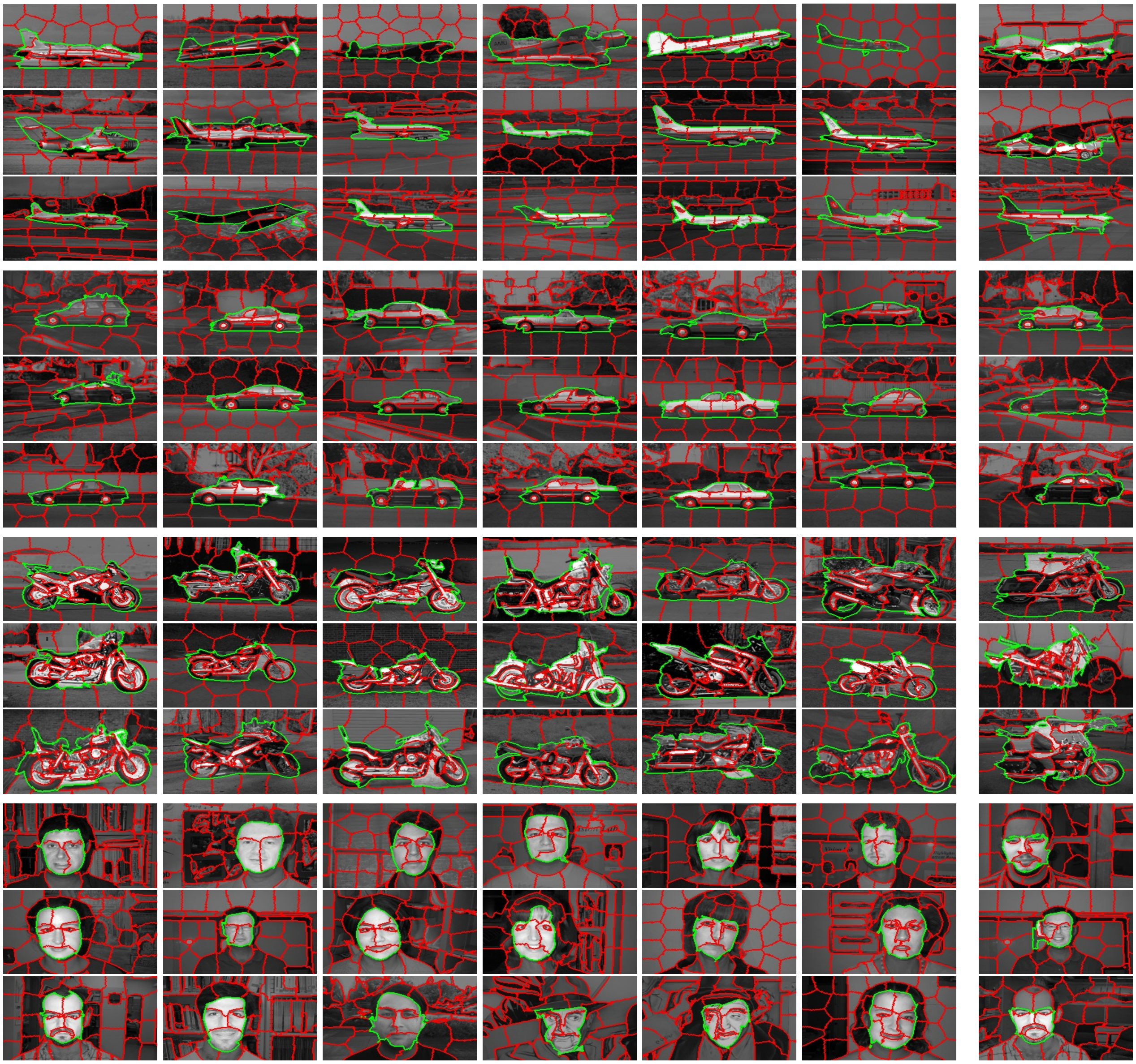

Figure 4. **Segmentation results** of our algorithm on example test images in standard datasets. Gray-scale images are used. The superpixel representation is shown in **red** and the final segmentation is shown in **green**. The brightness of the estimated background superpixels is diminished to aid visibility. We test the algorithm on Caltech faces, airplanes, motorbikes, and Caltech 101 cars. Note that the quality of segmentation is affected due to bad approximation of object boundaries by the superpixel boundaries. For most images the segmentation is quite **accurate**, even in the presence of intra-class variation and poor object-background contrast. Exemplar **failures** are shown in the **last column**. Performance for faces is low in table 1 because the unsupervised algorithm learned to segment faces without hair. This figure is most easily interpreted when viewed in color.

[14] N. Loeff, H. Arora, A. Sorokin, and D. Forsyth. Efficient unsupervised learning for localization and detection in object categories. In *Proc. of NIPS*, 2005.

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] X. Ren, C. Fowlkes, and J. Malik. Cue Integration in Figure/Ground Labeling. In *Proc. of NIPS*, 2005.

[17] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR (2)*, pages 1605–1614, 2006.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, 2000.

[19] C. Sutton and A. McCallum. Piecewise training for undirected models. In *Proc. of UAI*, pages 568–575, 2005.

[20] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *Proc. of ICCV*, pages 756–763, 2005.

[21] J. Winn and J. Shotton. The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects. In *Proc. of CVPR*, pages 37–44, 2006.

A. Learning as Variational Inference

Learning the model from a set of images of an object category involves estimating the parameters Θ of the model. The background distribution parameter $(\overline{h}_{P+1})$ is estimated from the background dataset; this distribution is fixed before estimating Θ. Obtaining the maximum likelihood estimate for Θ involves maximizing the likelihood of $P_{\Theta}(v)$, for which no closed form solution exists. Many algorithms for iterative inference such as EM, ICM, BP can be viewed as instances of variational inference [8]. Variational inference involves minimizing free-energy F_e with respect to an approximation $Q(h)$ to the posterior distribution $P_{\Theta}(h|v)$. Let D_{KL} be the K-L divergence, then F_e is defined as

$$\begin{aligned} F_e(Q,\Theta) &\doteq D_{KL}\{Q(h)||P_{\Theta}(h|v)\} - \log P_{\Theta}(v) \\ &= \int_h Q(h) \log \frac{Q(h)}{P_{\Theta}(h,v)} dh \end{aligned} \qquad (7)$$

EM update: Minimizing F_e without restrictions on $Q(h) = Q(\omega_{ij}|o_l, o_s)Q(o_l)Q(o_s)$ leads to E step of the EM algorithm, with $Q(h) = P_{\Theta}(h|v)$. This can be viewed as taking expectation with respect to the hidden random variables. Thus the complexity of the approach is the dimension space of the hidden variables. Let us assume, for simplicity, that the object center takes values on a $N_x \times N_y$ grid on the image, and scale takes values on a grid of size N_s. Then the complexity of the approach is $O(FPN_xN_yN_s)$. Unlike [6], this is linear in the number of features per image F, and parts in the model P. However, as we include more types of 2D transformations, this number explodes.

Variational (mean-field) update: Another possible approach is to minimize F_e restricting $Q(h) = Q(\omega_i j)Q(o_l)Q(o_s)$. This assumption basically makes the assignments (ω_{ij}) conditionally independent of the object configuration oc given the observations. Following [14], this makes inference much faster, as the complexity is now $O(FP) + O(N_xN_y) + O(N_s)$.

ICM (point estimate) update: Complexity can be reduced even further if we consider oc to be continuous variables and restrict $Q(o_l) = \delta_{o_l^*}$ and $Q(o_s) = \delta_{o_s^*}$. Minimizing F_e now involves finding the most likely oc^{new} as a function of the data, parameters and previous iteration estimate of oc^{old}. This makes inference even faster ($O(FP)$), and we do not constrain the solution to lie on the grid, but as the search space is more limited, the approach is prone to local minima.

Hybrid EM update: In experiments we observed the model is much better behaved in scale (see figure 5), as it has *only* one minimum, than in location, where there are multiple local minima. Thus we consider a hybrid approach with EM updates for $Q(w_{ij}|oc)$ and $Q(o_l)$ and a point estimate update of $Q(o_s)$. The updates of the parameters are:

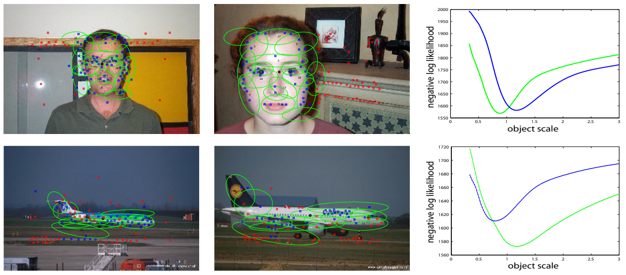

Figure 5. **Model behavior in scale**. The green ellipses represent the distribution in location of object parts of the model superimposed on the image according to the estimated object scale and location (black dot). The dots show the detected feature locations. Blue features are associated by the model to the object, and red features to the background. **Last column** plots negative log likelihood vs. the scale estimate for the images in the other two columns. There is only one local minimum, thus the use of point estimates for scale.

$$\begin{aligned} \mu_i &\leftarrow \frac{\sum_{n,j,o_l} \alpha^n_{ijo_l} o^n_s (f^{nl}_j - o_l)}{\sum_{n,j,o_l} \alpha^n_{ijo_l}} \\ \Sigma_i &\leftarrow \frac{\sum_{n,j,o_l} \alpha^n_{ijo_l} (o^n_s)^2 (f^{nl}_j - o_l)(f^{nl}_j - o_l)^T}{\sum_{n,j,o_l} \alpha^n_{ijo_l}} - \mu_i \mu_i^T \\ \overline{h}_i &\leftarrow \frac{\sum_{n,j,o_l} \alpha^n_{ijo_l} f^a_j}{\sum_{n,j,o_l} \alpha^n_{ijo_l}} \end{aligned}$$

Here, $\alpha^n_{ij,oc} = Q^n(o_l)Q^n(\omega^n_{ij}|o_l, o_s)$, and superscript n represents the attributes for the n-th training image. Maximization with respect to the object scale leads to solving the following quadratic equation

$$a(o^n_s)^2 + b(o^n_s) + c = 0, \quad \text{where} \qquad (8)$$

$$\begin{aligned} a &= -\sum_{i,j,o_l} \alpha^n_{ijo_l} (f^{nl}_j - o_l)^T \Sigma_i (f^{nl}_j - o_l) \\ b &= \sum_{i,j,o_l} \alpha^n_{ijo_l} \mu^T \Sigma_i (f^{nl}_j - o_l) \\ c &= \sum_{i,j,o_l} \alpha^n_{ijo_l} \end{aligned}$$

It can be easily shown that the above quadratic has a unique positive solution, which forms the update of the object scale. The complexity of these updates is $O(FPN_xN_y)$, equal to that of the non-scale invariant version [14].

Hybrid Variational update: We consider variational (mean-field) updates for $Q(w_{ij})$ and $Q(o_l)$ and a point estimate update of $Q(o_s)$. The updates of the parameters vary from the hybrid EM approach only in that $\alpha^n_{ij,oc} = Q^n(o_l)Q^n(\omega^n_{ij})$. The complexity of these updates is $O(FP + N_xN_y)$, equal to that of the non-scale invariant variational approach of [14]. We further refine the resulting model by applying point estimate updates, to remove the conditional independences introduced by the variational updates.