LOCUS: Learning Object Classes with Unsupervised Segmentation

J. Winn
Microsoft Research, Cambridge, UK

N. Jojic
Microsoft Research, Redmond, USA

Abstract

We address the problem of learning object class models and object segmentations from unannotated images. We introduce LOCUS (Learning Object Classes with Unsupervised Segmentation) which uses a generative probabilistic model to combine bottom-up cues of color and edge with top-down cues of shape and pose. A key aspect of this model is that the object appearance is allowed to vary from image to image, allowing for significant within-class variation. By iteratively updating the belief in the object's position, size, segmentation and pose, LOCUS avoids making hard decisions about any of these quantities and so allows for each to be refined at any stage. We show that LOCUS successfully learns an object class model from unlabeled images, whilst also giving segmentation accuracies that rival existing supervised methods. Finally, we demonstrate simultaneous recognition and segmentation in novel images using the learned models for a number of object classes, as well as unsupervised object discovery and tracking in video.

1. Introduction

Object recognition is one of the fundamental problems in computer vision. A practical object recognition system needs to be able to recognize tens of thousands of classes of objects and learn about new object classes from a small number of examples. When dealing with this number of classes, it is essential that learning does not require any human involvement in annotating training images. This paper addresses the problem of learning a model of an object class from a small number of example images which have not been segmented or marked up in any way. Instead, the object segmentation is inferred during the learning process.

LOCUS uses a generative model to combine bottom-up cues of color, texture and edge with top-down cues of object shape and pose. The generative model provides a framework for performing localisation, segmentation and pose estimation simultaneously. Rather than making any hard decisions, an iterative procedure allows successive refinement of each object's segmentation, its position and its pose.

Our key assumption is that whilst there can be significant changes in color/texture from one object instance to another, the object shape is consistent, and the variability of color/texture within a single instance of an object is limited. Hence, LOCUS aims to learn a representative object shape which (after undergoing a deformation, shift and scale) defines object interior and exterior regions of low color/texture variability within a single image, but typically immense variability across all the training images.

2. Related work

Previous approaches to learning object models can be categorised by the degree of human intervention required. The most labour-intensive approaches involve models whose structure is hand-crafted for a particular object class [1] or that require object-specific landmarks to be annotated on all training images [2]. More recent approaches use models applicable to a range of objects [3] or object classes [4, 5] but still require hand-segmented training data and so would not scale to large numbers of classes. A less labour-intensive alternative in [6] uses motion in video to learn an object model, which is then applied to still images. However, this method only learns the variability in appearance of a single object (e.g. due to pose and illumination); it does not capture variation between different objects of the same class.

Now let us turn to unsupervised approaches which do not require hand-segmented training data. Constellation models [7] can be learned from cluttered, unsegmented images. However, due to computational restrictions, such models learn a sparse set of parts that do not cover the entire object and so do not allow for object segmentation. Alternatively, Borenstein and Ullman [8] use the overlap between automatically extracted object fragments to determine the foreground/background segmentation. However, as no global shape model is used, there is no guarantee that the resulting cover corresponds to an actual object and is not just caused by background clutter resembling random object parts.

Our approach learns from unsegmented, cluttered images using a generative model incorporating both a global shape model and bottom-up edge and color cues. The part of our model used to infer the segmentation is similar to the stand-alone segmentation tool GrabCut [9]. However, in our case the segmentation is guided by the object class model rather than by a human user. The palette-invariance in our model builds on probabilistic index map (PIM) models [10], but LOCUS uses a more expressive color distribution in the entries of the palette, and introduces a deformation model for the index maps.

3. The LOCUS Generative Model

Figure 1 shows the hierarchical generative model of images used in LOCUS. Each class is represented by describing the process of forming a set of images containing an object of that class. The aim is to infer the details of this process given an image set and so determine the location, segmentation and pose of each contained object, whilst learning a common class model for all the objects.

The class-specific information is captured in the class mask model defining the broad global shape, the edge model defining the typical edge locations and (optionally) in a mild prior on the appearance features (color or texture). In addition, each image instance has hidden variables for its own object position, size, deformation, mask and appearance distribution or palette (either over color or texture). By allowing different object instances to have dramatically different appearance distributions, we make our model palette-invariant, similar to [10]. Even though the palette is treated as a hidden variable special to a particular image, the large number of pixels and low variability of color or texture within an object make the palette inference possible. Therefore, the object is defined primarily by its shape, edge map, and in terms of appearance, by the self-similarity of its colors or texture within a single image, instead of a strong global color or texture model. This palette-invariance allows us to cope with the dramatic appearance differences among different objects of the same class. For example, different horses can be brown, black, yellow, red, white or spotted but their shape remains consistent. This flexibility is illustrated in Fig. 2, where we show that the learned class mask model is typically quite strong, while the appearance of different instances of the same object varies dramatically.

It is important to note, however, that while our model performs well on segmentation tasks even when full palette invariance is assumed, we get an additional boost in recognition performance from capturing a (typically weak) class-dependent prior distribution over the palettes.

We augment each image $\mathbf{I}$ by creating a corresponding edge image $\mathbf{e}$ using a standard Canny edge detector. Our generative model is of both the normal image and the edge image, which are treated as separate observations. The statistical process of generating these two images is illustrated in Figure 1. First a (smooth) deformation field as well as an object position and size are sampled. Next the class mask probability image and edge mean and variance images are sampled from their respective prior distributions. These mask/edge images are then deformed by the deformation field, scaled and translated according to the previously generated position and size variables. The mask defining the foreground-background segmentation of the image is sampled from the transformed mask probability image. A foreground edge image is sampled using the transformed edge

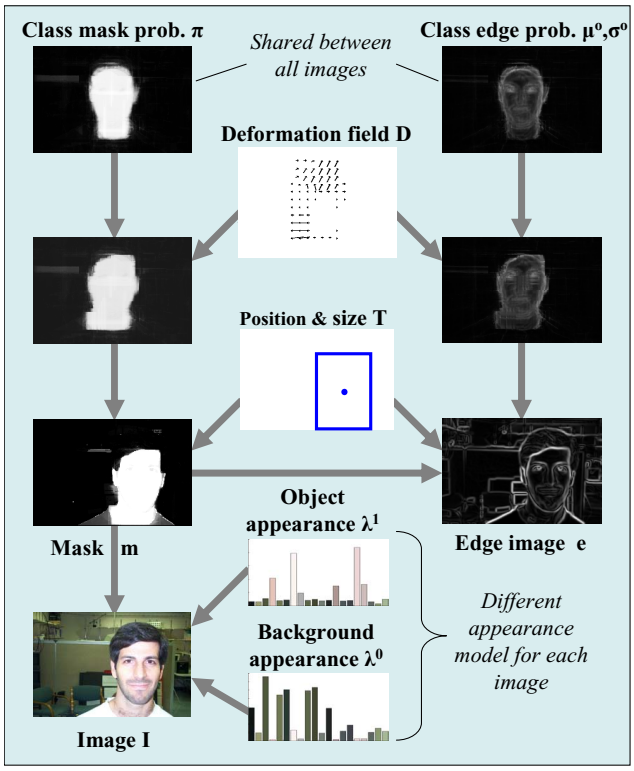

Figure 1: Illustration of the LOCUS generative model for images of an object class (in this case, faces). Class-specific information is contained in the class mask and edge probability models, which govern the broad shape and typical edge locations for an object class. The position, size, deformation, mask and appearance of individual object instances are allowed to vary from image to image. This generative process is detailed in in Section 3.

mean and variance images, and composited with a background edge image sampled from a background edge model to create the edge image. Finally, object and background appearance distributions are sampled and used with the mask to form an image.

In the rest of this section we will describe the elements of this model in more detail, and then use it in Section 4 to learn the object class parameters and perform unsupervised image segmentation through probabilistic inference.

The class mask and edge models

The class mask and edge models define distributions over the shape and edge locations of instances of the object class. They are illustrated at the top of Figure 1. We represent the object class by the learned parameters of these two models and in the simplest version of LOCUS they are the only model elements shared between images.

The class mask probability image defines a distribution over the shape of an object of that class in a canonical pose. It consists of a real-valued number $\pi_i \in [0, 1]$ for each pixel, giving the probability that the pixel lies inside the object. The indicator mask variables m_i are used to define

the foreground-background segmentation, so that

$$p(m_i = 1) = \pi_i. \quad (1)$$

These probabilities can be arranged into an image $\boldsymbol{\pi} = \{\pi_1, \pi_2, \ldots\}$ as illustrated in Figure 1. We define a prior distribution over $\boldsymbol{\pi}$ to be a product of beta distributions $P(\boldsymbol{\pi}) = \prod_i \mathrm{Beta}(\pi_i \,|\, u_0, u_1)$, where we set both the pseudo-counts u_0 and u_1 to be unity.

The class edge model defines a distribution over the presence or absence of edges for an instance of the class in a canonical pose. It consists of a Gaussian distribution over the Canny edge strength e_i for each pixel. The edge model is defined by mean image $\boldsymbol{\mu} = \{\mu_i\}_{i=1}^I$ and a variance image $\boldsymbol{\sigma} = \{\sigma_i\}_{i=1}^I$, which define the Gaussian models on edge observations. We have two such models, one for the background and another for the object, denoted by $\boldsymbol{\mu}^b, \boldsymbol{\sigma}^b$ and $\boldsymbol{\mu}^o, \boldsymbol{\sigma}^o$, respectively. Therefore,

$$\begin{aligned} p(e_i|m_i = 1) &= \mathcal{N}(e_i|\mu_i^o, \sigma_i^o) \\ p(e_i|m_i = 0) &= \mathcal{N}(e_i|\mu_i^b, \sigma_i^b). \end{aligned}$$

For both the background and the foreground, we define a prior distribution over $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ which is a product of Normal-Gamma distributions $P(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \prod_i \mathrm{NormalGamma}(\mu_i, \sigma_i \,|\, m^e, \lambda^e, a^e, b^e)$, where the hyper-parameters m^e, λ^e, a^e, b^e are chosen to give a broad prior distribution. The priors on the class shape and edge models define how much within-class variation we expect a priori, and help regularize learning. We use the same hyperparameters for all classes.

The preceding set of equations define a complete generative model of an edge map, with mask variables segmenting the object treated as hidden variables. However, in what follows we enrich the mask and edge model by transforming the prior distributions *before* generating mask and edge responses.

The deformation field

The deformation field allows for the shape of each object instance to vary from the canonical shape, due to within-class variability, changes in object pose or small changes in viewpoint. Our model has some similarities with that of Kannan et al [11] except that the deformation field is constrained to be smooth and overlapping patches are not used. We divide the image into discrete blocks (of 5×5 pixels) and associate a deformation vector $\mathbf{d}_i$ with each block, indexed by i. The deformation vectors are constrained to belong to a fixed set of discrete shifts, where a maximum shift size is imposed to keep this set relatively small (typically a few hundred shifts). This restriction is appropriate because large translations can be captured by the separate global transformation described below. The set of deformation vectors for all blocks $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2 \ldots\}$ defines the deformation field for an image.

The prior over the deformation field $\mathbf{D}$ is a Markov Random Field (MRF) which penalises the squared difference between neighbouring vectors,

$$P(\mathbf{D}) = \frac{1}{Z} \exp \sum_{(i,j)\in\bar{E}} -\alpha|\mathbf{d}_i - \mathbf{d}_j|^2 \quad (2)$$

where the distance $|\mathbf{d}_i - \mathbf{d}_j|^2$ is the Euclidean distance between the two vectors and $\bar{E}$ is the set of all edges in a 4-neighbour connected grid. The constant α controls how smooth the deformation can be (i.e. how much within-class deformation is allowed) and is currently set by hand to a fixed value.

The deformation field is applied to deform the class model latent images $\boldsymbol{\pi}, \boldsymbol{\mu}^o$ and $\boldsymbol{\sigma}^o$. For example, the deformation $\mathbf{D}$ can be applied to the mask probability image $\boldsymbol{\pi}$ to create a deformed image $\widetilde{\boldsymbol{\pi}}$ using

$$\widetilde{\boldsymbol{\pi}}(\mathbf{x}) = \boldsymbol{\pi}(\mathbf{x} - \mathbf{d}_i) \quad (3)$$

where $\mathbf{x}$ is a point in the ith block of $\widetilde{\boldsymbol{\pi}}$. Hence, it is possible for a pixel in $\boldsymbol{\pi}$ to be duplicated when the deformation is applied, allowing for local changes in scale.

Position and size transformation

Whilst the deformation model allows for local changes in pose and shape, the overall position and size of the object is handled by a separate global transformation $\mathbf{T}$ defined as a scaling S followed by a translation $\mathbf{t}$. Following [12], we discretise the space of transformations and restrict it to all whole-pixel translations at a small fixed range of scales. The ratio between successive scales was set at 1.2 as only a moderately coarse discretisation of the scale range is needed because the deformation field can also model small changes in scale.

The transformation can be applied to the deformed mask probability image $\widetilde{\boldsymbol{\pi}}$ to create a transformed, distorted version $\bar{\boldsymbol{\pi}}$ using

$$\bar{\boldsymbol{\pi}}(\mathbf{x}) = \widetilde{\boldsymbol{\pi}}(\mathbf{x}/S - \mathbf{t}). \quad (4)$$

Although we currently restrict the global transformation to a translation and scale, we could extend our model to allow full affine transforms, as described in [13].

The mask model

As indicated above, for each image, the ith pixel with measurement $\mathbf{z}_i$ (a vector representing color and/or texture) has an associated mask variable m_i which is 1 if that pixel is part of the object and 0 if it is part of the background. This choice of a binary mask (rather than a real valued mask) leads to more efficient and robust inference [14]. The dependence of the mask on the class mask probability image $\boldsymbol{\pi}$ is now given by

$$P_{\mathrm{class}}(\mathbf{m} \,|\, \boldsymbol{\pi}) = \prod_i \bar{\pi}_i^{m_i}(1 - \bar{\pi}_i)^{1-m_i}$$

where $\bar{\pi}_i$ is the ith element of a deformed, transformed version of $\boldsymbol{\pi}$.

We would also like to specify a prior which favours short segmentation boundaries that are aligned with intensity gradients in the image, as used in GrabCut[9]. This can be achieved by using an MRF prior of the form

$$P_{\text{mrf}}(\mathbf{m} \,|\, \gamma, \beta) = \frac{1}{Z} \exp \sum_{(i,j) \in \bar{E}} -\delta(m_i \neq m_j) \gamma e^{-\beta ||\mathbf{z}_i - \mathbf{z}_j||^2}$$

where $\bar{E}$ denotes the edges in an 8-neighbour connected grid (i.e. connected horizontally, vertically and diagonally). When the constant $\beta = 0$, this is simply an Ising prior, encouraging short boundaries. We set $\beta > 0$ so that this constraint is relaxed in regions of high contrast, favouring segmentations which align with contrast edges in the image. The setting of the constants γ and β is discussed in [9].

We select between these two distributions using a latent switch variable s so that our mask likelihood is

$$P(\mathbf{m} \,|\, \boldsymbol{\pi}, \gamma, \beta) = P_{\text{class}}(\mathbf{m} \,|\, \boldsymbol{\pi})^s P_{\text{mrf}}(\mathbf{m} \,|\, \gamma, \beta)^{1-s}. \quad (5)$$

During inference, uncertainty in the state of s means that the effect of both the class and MRF distributions is combined.

The edge model

In the same way that they affect the mask prior, the deformation field $\mathbf{D}$ and the rigid transformation $\mathbf{T}$ are applied to the foreground edge prior $\boldsymbol{\mu}^o, \boldsymbol{\sigma}^o$ to yield $\bar{\boldsymbol{\mu}}^o, \bar{\boldsymbol{\sigma}}^o$,

$$\begin{aligned} p(e_i | m_i = 1) &= \mathcal{N}(e_i | \bar{\mu}_i^o, \bar{\sigma}_i^o) \\ p(e_i | m_i = 0) &= \mathcal{N}(e_i | \mu_i^b, \sigma_i^b). \end{aligned} \quad (6)$$

In our experiments, the background model is not transformed, given the expected diversity in it, but if the objects in the background are to be fully explained away, then a mixture of LOCUS foreground models should be applied to each layer in the scene.

The appearance model

Each image has as a hidden variable its own palette, which either models its color or its texture (we try both alternatives). The appearance model consists of a full covariance Gaussian mixture model in either RGB color space or a 'texture space' consisting of the responses of 17 texture filters. For efficiency, the mixture components are shared between the foreground and background models and only the mixing proportions differ, as suggested in [15]. This sharing allows for the components to be learned once (at initialization) and from then on only the proportions of pixels in each component need to be updated for each layer, with the proportions defining a histogram over clusters. Hence, we achieve significantly greater efficiency than updating a separate mixture model for each layer. The mixture model parameters are $\boldsymbol{\theta} = \{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^0, \eta, \Sigma\}$, where $\boldsymbol{\lambda}^0$ and $\boldsymbol{\lambda}^1$ are the mixture coefficients (histograms) for the background and object respectively. We define a weak Dirichlet prior for each $\boldsymbol{\lambda}$.

The distribution over an RGB pixel or vector of filter responses $\mathbf{z}_i$ depends on the corresponding mask pixel m_i and the mixture model parameters $\boldsymbol{\theta}$,

$$P(\mathbf{z}_i \,|\, m_i, \boldsymbol{\theta}) = \sum_{k=1}^{K} \lambda_k^{m_i} \mathcal{N}(\mathbf{z}_i \,|\, \eta_k, \Sigma_k). \quad (7)$$

The use of a shared K-component model provides additional flexibility as it allows the use of fewer components for a simple background than for a complex object, or vice versa.

4. Inference and Learning

The aim of the inference algorithm is to learn a distribution over the latent variables in our model given the observed variables, the set of images $\mathcal{I} = \{\mathbf{I}^1, \ldots, \mathbf{I}^N\}$. Each image is defined by its measurement map $\{\mathbf{z}_i\}$ and edge map e_i. Exact inference is intractable and so we resort to using a structured variational approximation [16].

The approximate posterior over the variables corresponding to each image takes the factorised form

$$P(\mathbf{D}, \mathbf{T}, \mathbf{m}, \boldsymbol{\lambda}, s \,|\, \mathcal{I}) \approx Q(\mathbf{D}) Q(\mathbf{T}) Q(\lambda) Q(s) \prod_i Q(m_i)$$

and the posterior over the class and background variables is

$$P(\boldsymbol{\pi}, \boldsymbol{\mu}^o, \boldsymbol{\sigma}^o, \boldsymbol{\mu}^b, \boldsymbol{\sigma}^b \,|\, \mathcal{I}) \approx \prod_i Q(\pi_i) Q(\mu_i, \sigma_i) Q(\mu_i^b, \sigma_i^b)$$

where the product is over pixels in each case. We give each Q factor the same functional form as the corresponding factor in P, so that $Q(\pi_i)$ is a Beta distribution, $Q(\mu_i, \sigma_i)$ is a Normal-Gamma distribution, and so on. The only exception to this is that we select $Q(\mathbf{D})$ to be a delta function $\delta(\mathbf{D} = \mathbf{D}^\star)$ for reasons of efficiency.

Let the set of all latent variables be $\mathbf{X}$. Variational inference involves maximising the quantity

$$\mathcal{L}(Q) = \langle \log P(\mathbf{X}, \mathcal{I}) \rangle_Q + \mathbb{H}(Q) \quad (8)$$

where $\langle . \rangle_Q$ is an expectation under the distribution Q and $\mathbb{H}(Q)$ is the entropy of Q. We can maximise $\mathcal{L}(Q)$ with respect to any factor $Q(X_i), X_i \in \mathbf{X}$ by applying

$$\log Q(X_i) = \langle \log P(\mathbf{X}, \mathcal{I}) \rangle_{Q(X \setminus X_i)} + \text{const.} \quad (9)$$

where the notation $\langle . \rangle_{Q(X \setminus X_i)}$ means expectation under the distribution consisting of the product of all factors of Q except $Q(X_i)$. Approximate inference proceeds by repeatedly iterating though all latent variables in the order $\{\boldsymbol{\lambda}, \mathbf{m}, \mathbf{T}, \mathbf{D}, \boldsymbol{\pi}, \boldsymbol{\mu}^o, \boldsymbol{\sigma}^o, \boldsymbol{\mu}^b, \boldsymbol{\sigma}^b\}$ and optimising each in turn. Rather than expand (9) for each factor separately and implement each update by hand, we use Variational Message Passing (VMP) [17] to apply this variational inference procedure automatically.

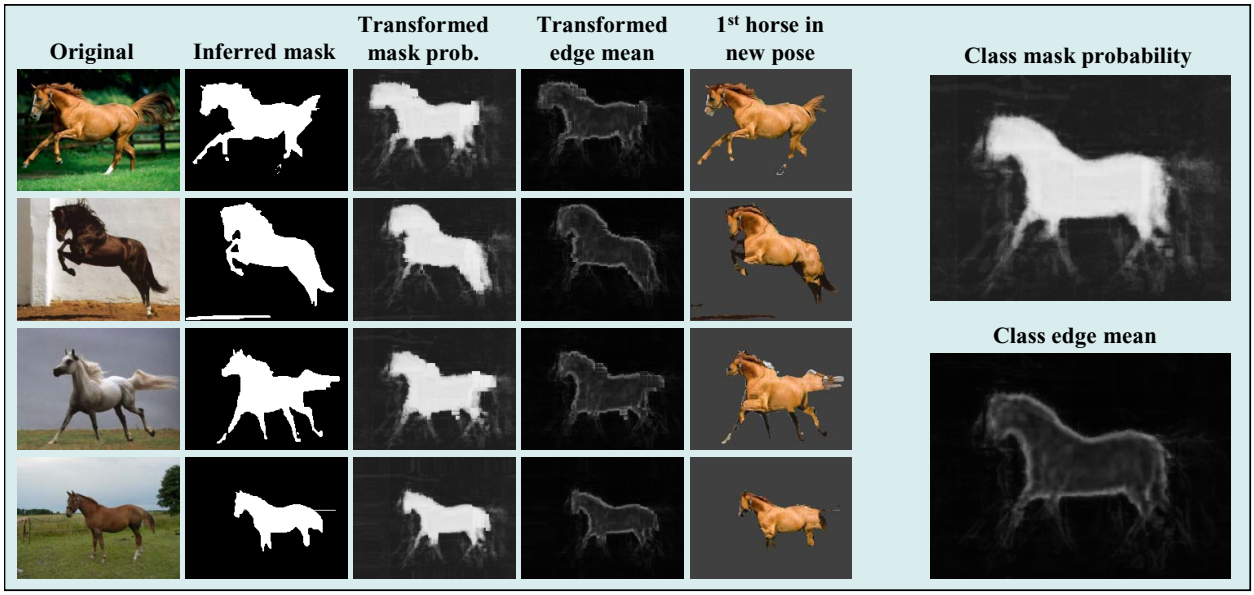

Figure 2: Unsupervised learning of a horse model. The left column shows four of the 20 images used to learn a class model consisting of a mask model and an edge model. The learned mask model is illustrated at the top right by the expected value of $\boldsymbol{\pi}$ i.e. the probability of the mask being on at each pixel. Beneath it, the class edge mean shows average edge strength for each pixel (the variance in strength is also learned but is not shown). These two images demonstrate that the model has captured the shape and outline of a horse in a neutral position, despite the dramatic variation in the color and pose of the horses in the example images. Column 2 gives the most probable segmentation of each horse under the variational distribution. Columns 3 and 4 demonstrates that the pose of each horse has been captured by showing the mask probability and edge mean images transformed into the inferred pose of each horse. Column 5 shows the first horse transformed into the position, size and pose of each of the other three, showing that a dense registration between horses has been obtained.

To achieve the optimisation of the factor $Q(\mathbf{D})$ corresponding to the deformation field $\mathbf{D}$, we need to maximise terms of the form

$$\sum_i \psi(\mathbf{d}_i) - \sum_{(i,j)\in\bar{E}} \alpha|\mathbf{d}_i - \mathbf{d}_j|^2 \qquad (10)$$

where the sums are over blocks and edges between blocks. We can find the deformation field $\mathbf{D}^\star$ which maximises (10) by using graph cuts in an α-expansion algorithm, as described in [18, 19].

The terms in $\mathcal{L}(Q)$ involving the mask $\mathbf{m}$ take the form

$$\sum_i \langle\phi(m_i)\rangle_{Q(\mathbf{m})} - \sum_{(i,j)\in E} \beta' \langle\delta(m_i \neq m_j)\rangle_{Q(\mathbf{m})} \qquad (11)$$

where the sums are over pixels and edges between pixels. Notice that the expression now includes expectations under Q because we wish to model the uncertainty in our mask, and so we cannot use graph cuts to optimise (11) directly. If we assume a factorised posterior distribution, we can apply (9) to optimise each pixel in turn (holding its neighbours fixed). However, this assumption is flawed as the posterior over mask pixels contains strong correlations due to the strong pairwise interaction terms and so the optimisation can get caught in a local minima. To avoid this, we first use graph cuts to find the configuration $\mathbf{m}^\star$ which would maximise (11) if the expectations were removed. Then in the variational framework we use a Q distribution of the form $Q(\mathbf{m}) = f[\mathbf{m} = \mathbf{m}^\star] + (1-f)q(\mathbf{m})$, where [] is the indicator function defining the deterministic distribution over $\mathbf{m}$ with all mass at $\mathbf{m}^\star$, and $q(\mathbf{m})$ is a normalized distribution over $\mathbf{m}$ thus ensuring that Q is also normalized. In our experiments, using $f \approx 0.3$ provided a bias towards the most probable mask (with at least 30% of mass on $\mathbf{m}^\star$), while still allowing for uncertainty.

5. Experimental Results

We applied LOCUS to several sets of images, each containing an object of the same class in a variety of positions, sizes and poses, against a range of backgrounds. Our algorithm segments the images automatically using *all images together* to find a consistent segmentation based on there being the same object in each.

The first image set contained 20 images from the Weizmann horse database [8], a selection of which are shown in the left hand column of Figure 2. The horses in these images have differing poses and scales but are all facing to the left. The images were scaled so that the longest side of the image was 200 pixels, so that object scale was learned relative to image size and independent of image resolution. On these re-scaled images, a Matlab implementation of our algorithm converged in a few minutes on a 3GHz PC.

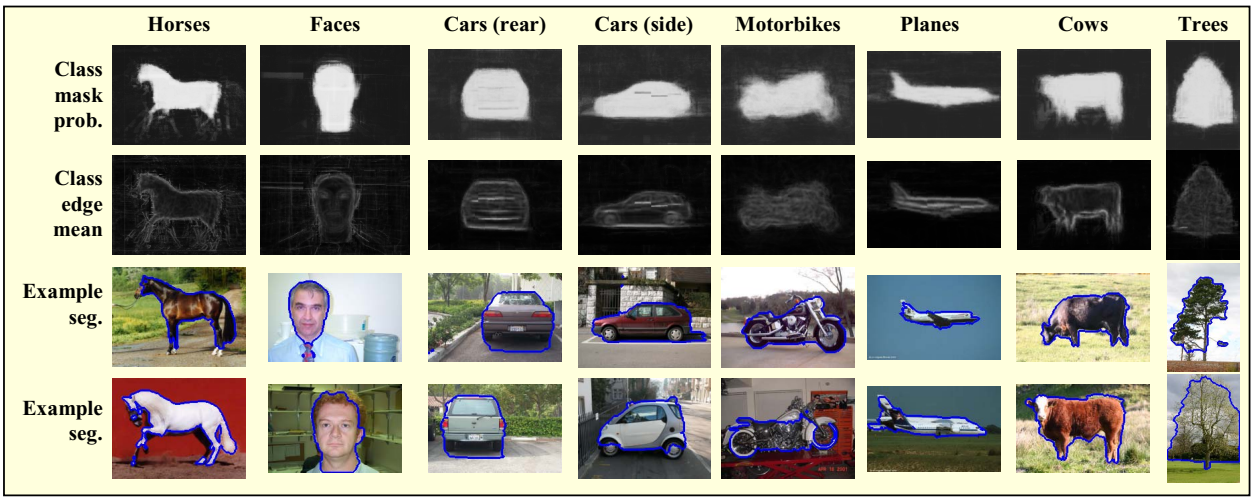

Figure 3: Results of unsupervised learning of eight object classes. The top two images of each column summarise the class mask and edge models learned from 20 images containing an object of that class. Beneath these are the inferred segmentations of two of images, indicated by a blue outline. For each class, LOCUS has extracted the shape of the object in a neutral pose. For the face and car classes some edge structure has been captured which corresponds to features internal to the object (e.g. eyes/wheels). Thus, the model for these classes represents more than just the outline/shape of the object.

The results of the learning process are summarised in Figure 2. The two images at the right of the figure summarise the learned class model by indicating the mask probability and average edge strength for each pixel. Each resembles a horse in a neutral position, showing that the global shape and outline of the horse has been inferred correctly. Some parts of the horse (e.g. the legs and tail) appear more faintly, either because they are not present in all images or because the position of the part is too far from the neutral position for the deformation field to cope with.

The second column of Figure 2 shows the inferred mask (i.e. segmentation) for each of the images in the first column. Subjectively, these appear very accurate for many of the images (a quantitive assessment is below). Typical segmentation mistakes are the inclusion of small areas of background close to the horse silhouette and the omission of all or part of the head, leg or tail, especially when they are in an unusual position.

The third and fourth columns show the mask probability and edge mean images transformed into the learned position and pose of the object for each training image. Notice how the deformation field is able to capture significant changes in object pose and also some changes in viewpoint. Because the deformation field provides a mapping from each object instance to the model, it has effectively provided a dense registration between corresponding parts of the different object instances. This allows us to transform one horse into the position and pose of another by first transforming it into the neutral position and then applying the deformation field learned from another image, as illustrated in the fifth column where the horse in the first image has been transformed into the learned poses of the other three.

Multiple object classes

Because our learning process does not require any labelling of images, it can be readily applied to many sets of images, so as to learn a range of object classes. To demonstrate the flexibility of the LOCUS model, we applied it to images of eight classes: horses, cows, cars (side and rear views), motorbikes, aeroplanes, faces and trees. Images were used from the Caltech [7] and TU Darmstadt [5] databases. For each class, the model was trained on 20 images containing objects of that class. Other than re-scaling each image, the only preprocessing performed was to flip asymmetric objects (e.g. cows) to face a consistent direction and to remove images where the object is very small (less than 20% of the width/height of the largest instance). The objects were not centred in each image, but they were moderately large – typically occupying around $15 - 30\%$ of the image pixels.

Figure 3 shows the learned class models and typical segmentations for each class. In each case, LOCUS has successfully extracted the shape and outline of the object in a neutral pose. In several classes (e.g. faces, cars), some internal edge structure has also been captured corresponding the internal parts such as eyes or wheels. The inferred segmentations are subjectively of good quality for all the classes, with excellent performance on cars, horses, faces. The weakest performance was on motorbikes because differing viewpoints meant that translation/scale was insufficient to align the bikes and learn a consistent outline.

Segmentation accuracy

For the horse and cars (side) classes, ground truth segmentations were available, which allowed us to assess the accuracy of the segmentation quantitively. The most proba-

ble mask $\mathbf{m}^\star$ under the learned posterior was determined for each image (see Section 4) and the percentage of pixels in agreement with the ground truth segmentation was found. The average percentage across all images in each class is shown in Table 1. For comparison, the table also includes results from Borenstein et al [4] whose method used 54 hand-segmented training images. Their quoted results are for 200 test images and so we re-ran LOCUS with 200 horse images to provide a fairer comparison – this is the result which is reported in the table. However, our segmentation accuracy was very similar (within 0.5%) for various subsets of the data, even for small datasets of 20 horse images.

As can be seen, our unsupervised method achieved segmentation accuracies in excess of 93%, which rivals the fully supervised method. Borenstein and Ullman [8] also report results for unsupervised segmentation. They achieve accuracies of 90-92% but on the task of segmenting cropped horses heads – a significantly more constrained situation than segmenting entire horses in arbitrary poses.

The benefit of segmenting all images together is highlighted by the final row of Table 1. This gives the segmentation accuracy if we remove the class model (the shared part of the model) and segment each image separately. Hence, we just learn a mask $\mathbf{m}$ and the color models $\boldsymbol{\lambda}$ for each image so as to minimize the visual entropy within each part of the scene. For both cases, this gives a significant reduction in accuracy. There is a greater reduction for cars than for horses because the backgrounds of the car images are typically more cluttered – if we remove the class model for object shape, the model is free to erroneously label background clutter as foreground.

	Segmentation accuracy	
	Horses	Cars (side)
LOCUS (color)	93.1%	91.4%
LOCUS (texture)	**93.0%**	**94.0%**
unannotated training images		
Borenstein et al	93.6%	-
hand-segmented training images		
LOCUS - no class model	88.6%	82.1%

Table 1: Average segmentation accuracy for the horse and car (side) classes with both color and texture appearance models. The accuracy given is the average percentage of correctly labelled pixels across all the images of each class. For comparison, we include results from Borenstein et al [4] showing that our unsupervised learning achieves similar segmentation accuracies to a method which requires hand segmented training data. The use of a texture model instead of a color model gives a significant improvement in accuracy for the car class but leaves the horse segmentation accuracy effectively unchanged. The final row demonstrates the reduction in accuracy if the class model is removed.

True label	**Inferred label**					
	Cars (rear)	Cars (side)	Cows	Faces	Horses	Planes
Cars (rear)	18				1	
Cars (side)		16				4
Cows			6			
Faces		1		19		
Horses		1	3		12	2
Planes						19

Table 2: Confusion matrix for recognition with LOCUS.

Recognition with segmentation

We applied the LOCUS model to the task of object recognition to see what performance could be achieved using essentially only outline information. For each class, we fixed the variational parameters for $\boldsymbol{\pi}, \boldsymbol{\mu}^o, \boldsymbol{\sigma}^o$ to the values learned previously from 20 training images of that class. We then estimated the variational posterior over the remaining variables for each test image and used $\mathcal{L}(Q)$ for classification. Because we learned a distribution over all hidden variables including the mask, we also learned a segmentation of the object being classified. The average classification accuracy for six object classes was 88% (see Table 2 for a confusion matrix). Notice that the global shape model allows discrimination between horses and cows, which are difficult to distinguish using only local appearance information.

Unsupervised motion segmentation

The LOCUS model can also be applied to the frames of a video sequence, which results in unsupervised segmentation of video into two layers[1]. The palette invariance gives robustness to large changes in illumination, pose and background clutter.

6. Future work and conclusions

We would like to extend LOCUS to model more details of the interior appearance of objects, so as to improve classification performance. However, we would still like the model to be largely invariant to the choice of the palette for an object instance. To achieve this, we use the probabilistic index map (PIM) representation of structure, as proposed in [10]. The PIM model allows us to learn, in an unsupervised fashion, the internal parts within the object, corresponding to, for example, an aeroplane's tail fin, a person's skin or a car's windows (see Figure 4). A PIM can be seen as an extension of a Multiple Cause Vector Quantizer [20] with a more general form of part appearance model.

For the foreground class, instead of a single distribution over local features, we use the PIM model which assigns to every pixel i an index s_i. These indices point to an entries of a smaller palette of appearance models (e.g., in Figure 4, we used six different indices denoted by different colors). As

[1] See http://johnwinn.org/Research/LOCUS.html for video results.

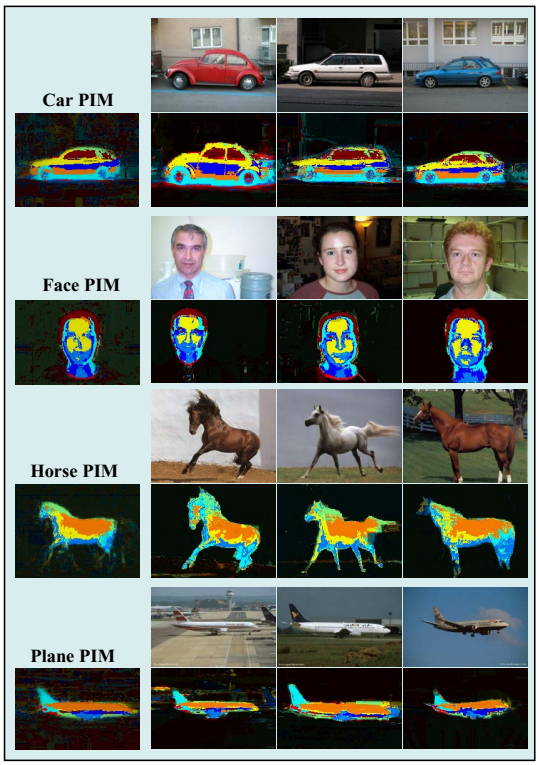

Figure 4: Modes of the deformable probabilistic index maps (dPIMs) and inferred individual index maps. The deformation field allows for local changes in location of particular parts and hence dPIMs are less diffuse and capture more fine-grained internal structure than standard PIMs.

opposed to [10], here all indices share the meta-palette, as did the foreground and background models in the previous sections, and the appearance distributions for each index are defined by the histogram $\lambda_k^{s_i}$

$$P(\mathbf{z}_i \mid s_i, m_i = 1, \boldsymbol{\theta}) = \sum_{k=1}^{K} \lambda_k^{s_i} \mathcal{N}(\mathbf{z}_i \mid \eta_k, \Sigma_k). \qquad (12)$$

The variability in the object structure is then defined by the distribution over indices, $p(\{s_i\}) = \prod_i p(s_i)$.

The mask variable m_i can be seen as a special case of a PIM index, except that in this paper, only the foreground $m_i = 1$ undergoes a deformation. By still using the mask model from the previous section, and requiring $p(s_i)$ to undergo the same deformation as the mask prior, we introduce the *deformable* PIM model illustrated in Figure 4 by the modes of the class priors $p(s_i)$ and the inferred index map s_i for a few images. The addition of the deformable PIM to the model leads to a more precise localization of the object and better registration of object parts. We are in the process of applying this model to improve our segmentation and classification accuracy.

In conclusion, LOCUS achieves simultaneous localisation, pose estimation, segmentation and recognition of objects in still images or video. In addition to PIMs, we are investigating extensions to LOCUS to allow for varying object viewpoints, multiple objects and occlusion.

Acknowledgements

The authors would like to thank Jamie Shotton for suggesting the use of the Weizmann horse database.

References

[1] A. L. Yuille, D. S. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–112, 1992.

[2] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active Shape Models - their training and applications. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[3] S. X. Yu and J. Shi. Object-specific figure-ground segmentation. In *CVPR'03*, 2003.

[4] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *Proceedings IEEE workshop on Perceptual Organization in Computer Vision, CVPR 2004*, 2004.

[5] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop om Statistical Learning in Computer Vision*, May 2004.

[6] M. Pawan Kumar, P. H. S. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *BMVC'04*, 2004.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR'03*, volume II, pages 264–271, 2003.

[8] E. Borenstein and S. Ullman. Learning to segment. In *ECCV 2004*, pages 315–328, 2004.

[9] C. Rother, V. Kolmogorov, and A. Blake. GrabCut -interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (SIGGRAPH)*, August 2004.

[10] N. Jojic and Y. Caspi. Capturing image structure with probabilistic index maps. In *CVPR '04*, July 2004.

[11] A. Kannan, N. Jojic, and B. Frey. Generative model for layers of appearance and deformation. In *AIStats '05*, 2005.

[12] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *CVPR '01*, 2001.

[13] J. Winn and A. Blake. Generative affine localisation and tracking. In *Advances in Neural Information Processing Systems*, volume 17, pages 1505–1512, 2004.

[14] C.K.I. Williams and M. K. Titsias. Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16(5):1039–1062, 2004.

[15] L. Xie and P. Prez. Slightly supervised learning of part-based appearance models. In *Proc. IEEE Workshop on Learning in Comp. Vision and Pattern Recog, LCVPR'04*, June 2004.

[16] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In *Learning in Graphical Models*, pages 105–162. Kluwer, 1998.

[17] J. Winn and C. M. Bishop. Variational Message Passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

[18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *ICCV (1)*, pages 377–384, 1999.

[19] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 26, February 2004.

[20] D. Ross and R. Zemel. Multiple-cause vector quantization. In *Advances in Neural Information Processing Systems*, volume 15, 2002.