# OBJCUT: EFFICIENT SEGMENTATION USING TOP-DOWN AND BOTTOM-UP CUES

M. Pawan Kumar, Philip H.S. Torr, Andrew Zisserman

▶ **To cite this version:**

M. Pawan Kumar, Philip H.S. Torr, Andrew Zisserman. OBJCUT: EFFICIENT SEGMENTATION USING TOP-DOWN AND BOTTOM-UP CUES. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010. hal-00773609

**HAL Id: hal-00773609**

**https://inria.hal.science/hal-00773609**

Submitted on 14 Jan 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# OBJCUT: Efficient Segmentation using Top-Down and Bottom-Up Cues

M. Pawan Kumar
Dept. of Eng. Science
University of Oxford
pawan@robots.ox.ac.uk

P.H.S. Torr
Dept. of Computing
Oxford Brookes University
philiptorr@brookes.ac.uk

A. Zisserman
Dept. of Eng. Science
University of Oxford
az@robots.ox.ac.uk

**Abstract**

We present a probabilistic method for segmenting instances of a particular object category within an image. Our approach overcomes the deficiencies of previous segmentation techniques based on traditional grid conditional random fields (CRF), namely that (i) they require the user to provide seed pixels for the foreground and the background; and (ii) they provide a poor prior for specific shapes due to the small neighborhood size of grid CRF. Specifically, we automatically obtain the pose of the object in a given image instead of relying on manual interaction. Furthermore, we employ a probabilistic model which includes shape potentials for the object to incorporate top-down information that is global across the image, in addition to the grid clique potentials which provide the bottom-up information used in previous approaches. The shape potentials are provided by the pose of the object obtained using an object category model. We represent articulated object categories using a novel layered pictorial structures model. Non-articulated object categories are modelled using a set of exemplars. These object category models have the advantage that they can handle large intra-class shape, appearance and spatial variation. We develop an efficient method, OBJCUT, to obtain segmentations using our probabilistic framework. Novel aspects of this method include: (i) efficient algorithms for sampling the object category models of our choice; and (ii) the observation that a sampling-based approximation of the expected log likelihood of the model can be increased by a single graph cut. Results are presented on several articulated (e.g. animals) and non-articulated (e.g. fruits) object categories. We provide a favorable comparison of our method with the state of the art in object category specific image segmentation, specifically the methods of Leibe & Schiele and Schoenemann & Cremers.

**Index Terms**

Object Category Specific Segmentation, Conditional Random Fields, Generalized EM, Graph Cuts

## I. INTRODUCTION

Image segmentation has seen renewed interest in the field of Computer Vision, in part due to the arrival of new efficient algorithms to perform the segmentation [5], and in part due to the resurgence of interest in object category detection [11], [26]. Segmentation fell from favor due to an excess of papers attempting to solve ill posed problems with no means of judging the result. In contrast, interleaved object detection and segmentation [4], [26], [29], [37], [45] is both well posed and of practical use. Well posed in that the result of the segmentation can be quantitatively judged, e.g. how many pixels have been correctly and incorrectly assigned to the object. Of practical use because image editing tools can be designed that provide a *power assist* to cut out applications like 'Magic Wand', e.g. "I know this is a horse, please segment it for me, without the pain of having to manually delineate the boundary."

The conditional random field (CRF) framework [25] provides a useful model of images for

segmentation and their prominence has been increased by the availability of efficient publically available code for their solution. The approach of Boykov and Jolly [5], and more recently its application in a number of systems including *GrabCut* by Rother *et al.* [34], strikingly demonstrates that with a minimum of user assistance objects can be rapidly segmented (e.g. by employing user-specified foreground and background seed pixels). However samples from the distribution defined by the commonly used grid CRFs (e.g. with 4 or 8-neighborhood) very rarely give rise to realistic shapes and on their own are ill suited to segmenting objects. For example, Fig. 1(c) shows the result of segmenting an image containing a cow using the method described in [5]. Note that the segmentation does not look like the object despite using a large number of seed pixels (see Fig. 1(b)) due to the small neighborhood size of the grid CRF, which cannot provide global top-down information.

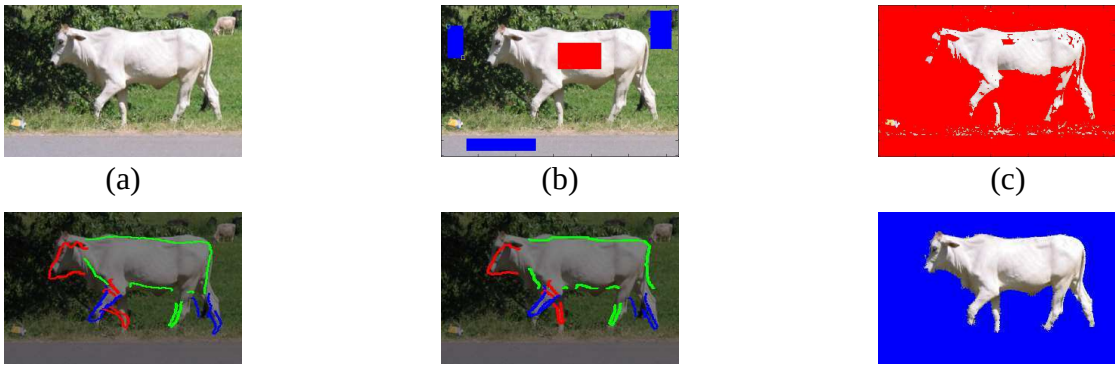



Fig. 1. *Segmentation obtained using the* CRF *formulation.* **(a)** *Original image containing a cow.* **(b)** *The red and blue rectangles indicate the object and background seed pixels respectively which are provided by the user.* **(c)** *Segmentation obtained using the method described in [5]. Note that the segmentation is not object-like due to the poor prior provided by the grid* CRF. **(d)**,**(e)** *The cow is roughly localized using the pictorial structures model [9], [12]. The parts detected are shown overlaid on the image. Note that the position and shape of the parts differs between the two detections (e.g. the head and the torso).* **(f)** *The segmentation obtained using our method. Unlike previous methods [2], [5], [34], the segmentation is object-like.*

In contrast, models used for object detection utilize the global information of the object to localize it in the image. Examples of such models include deformable templates [6] and triangulated polygons [8]. In this work we employ a set of shape and texture exemplars, similar to [14], [40], [42]. Given an image, the object is detected by matching the exemplars to the image. Such a model is particularly suitable for non-articulated object categories where a sufficiently large set of exemplars can be used to handle intra-class shape and appearance variation.

For articulated objects, in addition to shape and appearance, there might also be a considerable

spatial variation (e.g. see Fig. 18). In order to manage this variability there is a broad agreement that articulated object categories should be represented by a collection of spatially related parts each with its own shape and appearance. This sort of approach dates back to the pictorial structures model introduced by Fischler and Elschlager three decades ago [12]. Recently, pictorial structures [9] and other related models [11], [29], [37] have been shown to be very successful for the task of object recognition. Furthermore, pictorial structures have been highly effective in detecting fronto-parallel views of objects [9], [32]. Here (and throughout the rest of the paper), by detection we mean obtaining a rough localization of the object given that the image contains an instance of the object category of interest. However, these models alone are not suitable for obtaining a pixel-wise segmentation of the image. For example, Fig. 1(d) and (e) show two samples of the distribution obtained by matching the pictorial structures model of a cow to an image (as described in section V).

In this work, we combine the models used for object detection with the grid CRF framework used for segmentation. The coarse localization of the object obtained by matching a model to an image provides us rough regions where the foreground (i.e. the object) and background are present. These regions are used to obtain the object and background seed pixels. The seed pixels could then be directly employed to obtain the segmentation using CRF based methods. The result would be an automated Boykov-Jolly style segmentation algorithm [5]. However, such an approach would still suffer from the problem that the distribution of the grid CRF would not provide a good estimate for the shape of the object. In order to address this deficiency, we go beyond the probabilistic models of previous approaches. Specifically, we introduce a new framework that combines the grid CRF (which provides bottom-up information) with an object category model (which provides global top-down information across the image plane).

Using the above framework, pixels of an image can be labelled as belonging to the foreground or the background by jointly inferring the MAP estimate of the object detection and segmentation. However, it would be undesirable to depend only on the MAP detection since it may not localize some portion of the object well. We overcome this problem by marginalizing over various detections obtained for a given image. Fig. 1(d) and (e) show two such detections found using the pictorial structures model of a cow (see section V). Fig. 1(f) shows the segmentation obtained using our approach. Unlike previous methods, the segmentation is object-like.

In summary, we cast the problem of object category specific segmentation as that of estimating

a probabilistic model which consists of an object category model in addition to the grid CRF. Put another way, the central idea of the paper is to incorporate a 'shape prior' (either non-articulated or articulated) to the problem of object segmentation. We develop an efficient method, OBJCUT, to obtain segmentations using this framework. The basis of our method are two new theoretical/algorithmic contributions: (i) we provide efficient algorithms for marginalizing or optimizing the object category models of our choice; and (ii) we make the observation that a sampling-based approximation of the expectation of the log likelihood of a CRF under the distribution of some latent variables can be efficiently optimized by a single st-MINCUT.

**Related Work:** Many different approaches for segmentation using both top-down and bottom-up information have been reported in the literature. We start by describing the methods which require a limited amount of manual interaction. Huang *et al.* [19] describe an iterative algorithm which alternates between fitting an active contour to an image and segmenting it on the basis of the shape of the active contour. Cremers *et al.* [7] extend this by using multiple competing shape priors and identifying the image regions where shape priors can be enforced. However, the use of level sets in these methods makes them computationally inefficient. Freedman *et al.* [13] describe a more efficient algorithm based on st-MINCUT which uses a shape prior for segmentation. However, note that in all the above methods the user provides the initial shape of the segmentation. The quality of the solution heavily depends on a good initialization. Furthermore, these methods are not suited for parts based models and cannot handle articulation.

There are a few automatic methods for combining top-down and bottom-up information. For example, Borenstein and Ullman [4] describe an algorithm for segmenting instances of a particular object category from images using a patch-based model learnt from training images. Leibe and Schiele [26] provide a probabilistic formulation for this while incorporating spatial information of the relative locations of the patches. Winn and Jojic [44] describe a generative model which provides the segmentation by applying a smooth deformation field on a class specific mask. Shotton *et al.* [38] propose a novel texton-based feature which captures long range interactions to provide pixel-wise segmentation. However, all the above methods use a weak model for the shape of the object which does not provide realistic segmentations.

Winn and Shotton [45] present a segmentation technique using a parts-based model which incorporates spatial information between neighboring parts. Their method allows for arbitrary scaling but it is not clear whether their model is applicable to articulated object categories. Levin

and Weiss [27] describe an algorithm that learns a set of fragments for a particular object category which assist the segmentation. The learnt fragments provide only local cues for segmentation as opposed to the global information used in our work. The segmentation also relies on the maximum likelihood estimate of the position of these fragments on a given test image (found using normalized cross-correlation). This has two disadvantages:

- The spatial relationship between the fragments is not considered while matching them to an image (e.g. it is possible that the fragment corresponding to the legs of a horse is located above the fragment corresponding to the head). Thus the segmentation obtained would not be object-like. In contrast, we marginalize over the object category model while taking into account the spatial relationships between the parts of the model.
- The algorithm becomes susceptible to error in the presence of background clutter. Indeed, the segmentations provided by [27] assume that a rough localization of the object is known *a priori* in the image. It is not clear whether normalized cross-correlation would provide good estimates of the fragment positions in the absence of such knowledge.

More recently, Schoenemann and Cremers [35] proposed an approach to obtain the globally optimal segmentation for a given shape prior. Although they extended their framework to handle large deformations [36], it still cannot handle articulated object categories such as humans and quadrupeds. Moreover, it is not clear whether their approach can be extend to parts based models which are known to provide an elegant representation of several object categories.

**Outline:** The paper is organized as follows. In section II the probabilistic model for object category specific segmentation is described. Section III gives an overview of an efficient method for solving this model for foreground-background labeling. We provide the details of our choice of representations for articulated and non-articulated objects in section IV. The important issue of automatically obtaining the samples of the object from a given image is addressed in section V. Results for several articulated and non-articulated object categories and a comparison with other methods is given in section VI.

A preliminary version of this paper has appeared as [23]. Since its publication, similar techniques have been successfully applied for accurate object detection in [31], [33].

## II. Object Category Specific Segmentation Model

In this section we describe the model that forms the basis of our work. There are three issues to be addressed in this section: (i) how to make the segmentation conform to foreground and

background appearance models; (ii) how to encourage the segmentation to follow edges within the image; and (iii) how to encourage the outline of the segmentation to resemble the shape of the object. We begin by briefly describing the model used in previous works [2], [5], [34].

**Contrast Dependent Random Field:** We denote the segmentation of an image by a labeling function $f(\cdot)$ such that

$$f(a) = \begin{cases} 0 & \text{if} \quad a \text{ is a foreground pixel,} \\ 1 & \text{if} \quad a \text{ is a background pixel.} \end{cases} \tag{1}$$

Given image $\mathbf{D}$, previous work on segmentation relies on a conditional random field (CRF) [25] or equivalent contrast dependent random field (CDRF) [24] framework which models the conditional distribution $\Pr(f|\mathbf{D}, \boldsymbol{\theta})$. Here $\boldsymbol{\theta}$ denotes the parameters of the CDRF. By assuming the Markovian property on the above distribution and using the Hammersley-Clifford theorem, $\Pr(f|\mathbf{D}, \boldsymbol{\theta})$ can be written as

$$\Pr(f|\mathbf{D}, \boldsymbol{\theta}) = \frac{1}{Z_1(\boldsymbol{\theta})} \exp(-Q_1(f; \mathbf{D}, \boldsymbol{\theta})), \tag{2}$$

where $Z_1(\boldsymbol{\theta})$ is the partition function and the energy $Q_1(f; \mathbf{D}, \boldsymbol{\theta})$ has the form

$$Q_1(f; \mathbf{D}, \boldsymbol{\theta}) = \sum_{v_a \in \mathbf{v}} \theta^1_{a;f(a)} + \sum_{(a,b) \in \mathcal{E}} \theta^p_{ab;f(a)f(b)} + \theta^c_{ab;f(a)f(b)}. \tag{3}$$

The terms $\theta^1_{a;f(a)}$, $\theta^p_{ab;f(a)f(b)}$ and $\theta^c_{ab;f(a)f(b)}$ are called the unary, prior and contrast potentials respectively. As in previous work [5], we define these terms as follows.

*Unary Potential:* The unary potential $\theta^1_{a;f(a)}$ is the emission model for a pixel and is given by

$$\theta^1_{a;f(a)} = \begin{cases} -\log(\Pr(\mathbf{D}_a|\mathcal{H}_{obj})) & \text{if} \quad f(a) = 0 \\ -\log(\Pr(\mathbf{D}_a|\mathcal{H}_{bkg})) & \text{if} \quad f(a) = 1, \end{cases} \tag{4}$$

where $\mathcal{H}_{obj}$ and $\mathcal{H}_{bkg}$ are the appearance models for foreground and background respectively. For this work, $\mathcal{H}_{obj}$ and $\mathcal{H}_{bkg}$ are modelled as RGB distributions. The term $\mathbf{D}_a$ denotes the RGB values of the pixel $a$.

Note that both the data independent prior term $\theta^p_{ab;f(a)f(b)}$ and the data dependent contrast term $\theta^c_{ab;f(a)f(b)}$ are pairwise potentials (i.e. they are functions of two neighboring pixels). Below, we provide the exact form of these terms separately while noting that their effect should be understood together.

*Prior Term:* Let $f(a)$ and $f(b)$ be the labels for neighboring variables $v_a$ and $v_b$ respectively. Then the corresponding prior term is given by

$$\theta^p_{ab;f(a)f(b)} = \begin{cases} \kappa_1 & \text{if} \quad f(a) = f(b), \\ \kappa_2 & \text{if} \quad f(a) \neq f(b), \end{cases} \tag{5}$$

*Contrast Term:* The form of the data dependent contrast term $\theta^c_{ab;f(a)f(b)}$ is inspired by previous work in segmentation [5]. For two neighboring pixels $a$ and $b$, it is given by

$$\theta^c_{ab;f(a)f(b)} = \begin{cases} 0 & \text{if} \quad f(a) = f(b), \\ -\gamma(a,b) & \text{if} \quad f(a) \neq f(b). \end{cases} \tag{6}$$

The term $\gamma(a,b)$ is defined such that it reduces the cost within the Ising model prior for $f(a) \neq f(b)$ in proportion to the difference in intensities of pixels $a$ and $b$, i.e.

$$\gamma(a,b) = \lambda \left(1 - \exp\left(\frac{-\Delta^2(a,b)}{2\sigma^2}\right) \frac{1}{\text{dist}(a,b)}\right), \tag{7}$$

where $\Delta(a,b)$ measures the difference in the RGB values of pixels $a$ and $b$, i.e. $\mathbf{D}_a$ and $\mathbf{D}_b$, and dist$(a,b)$ is the Euclidean distance between $a$ and $b$ [5].

In this work, we use the following weight values: $\kappa_1 = 1$, $\kappa_2 = 2.2$, $\lambda = 1$ and $\sigma = 5$. As shown in [24], these weight value are suitable for encouraging contiguous segments whose boundaries lie on image edges. Empirically, these weights were found to provide good results for a large variety of images.

Reducing the cost of the Ising model prior term in this manner makes the pairwise terms, i.e. $\theta^p_{ab;f(a)f(b)} + \theta^c_{ab;f(a)f(b)}$, discontinuity preserving [3], [16]. Note that the contrast term $\theta^c_{ab;f(a)f(b)}$ cannot be included in the prior (since the prior term is not data dependent). Rather it leads to a pairwise linkage between neighboring random variables and pixels as shown in the graphical model given in Fig. 2.

**Object Category Specific** CDRF**:** We introduce an object category model, parameterized by $\mathbf{\Omega}$, to the grid CDRF framework which will favor segmentations of a specific shape as shown in the graphical model depicted in Fig. 2. We refer to this extension of the grid CDRF model as the *Object Category Specific* CDRF. Note that $\mathbf{\Omega}$ is connected to the hidden variables corresponding to the labeling $f$. This gives rise to an additional term in the energy function of the Object Category Specific CDRF which depends on $\mathbf{\Omega}$ and $f$. Following previous work on object category specific image segmentation [7], [19], we define the energy function as

$$Q_2(f, \mathbf{\Omega}; \mathbf{D}, \boldsymbol{\theta}) = \sum_{v_a \in \mathbf{v}} \left(\theta^A_{a;f(a)} + \theta^S_{a;f(a)}\right) + \sum_{(a,b)\in\mathcal{E}} \left(\theta^p_{ab;f(a)f(b)} + \theta^c_{ab;f(a)f(b)}\right), \tag{8}$$

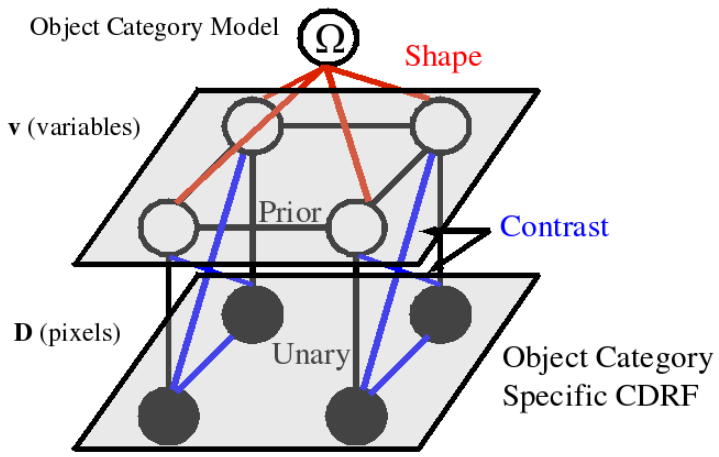



Fig. 2. *Graphical model representation of the Object Category Specific* CDRF. *The random variables* **v** *are shown as unfilled circles, while the observed data* **D** *is shown using filled circles. The connections induced by the contrast term are shown as the blue edges below the random variables. Note that some of these connections (e.g. connecting the random variables on the left with pixels on the right) are not shown for the sake of clarity of the image. The random variables* **v** *lie in a plane. Together with the pixels shown below this plane, these form the* CDRF *used for segmentation. In addition to these terms, the Object Category Specific* CDRF *makes use of an object category model* $\mathbf{\Omega}$ *(shown lying above the plane). The model* $\mathbf{\Omega}$ *guides the segmentation towards a realistic shape closely resembling the object of interest.*

with posterior

$$\Pr(f, \mathbf{\Omega}|\mathbf{D}, \boldsymbol{\theta}) = \frac{1}{Z_2(\boldsymbol{\theta})} \exp(-Q_2(f, \mathbf{\Omega}; \mathbf{D}, \boldsymbol{\theta})). \tag{9}$$

Here $\boldsymbol{\theta}$ is the parameter of the Object Category Specific CDRF and $Z_2(\boldsymbol{\theta})$ is the partition function. The prior term $\theta^p_{ab;f(a)f(b)}$ and contrast term $\theta^c_{ab;f(a)f(b)}$ are as defined above. The potentials $\theta^A_{a;f(a)}$ and $\theta^S_{a;f(a)}$ are described below.

*Appearance Potential:* The appearance potential $\theta^A_{a;f(a)}$ is the same as the unary potential of the CDRF i.e.

$$\theta^A_{a;f(a)} = \begin{cases} -\log(\Pr(\mathbf{D}_a|\mathcal{H}_{obj})) & \text{if} \quad f(a) = 0 \\ -\log(\Pr(\mathbf{D}_a|\mathcal{H}_{bkg})) & \text{if} \quad f(a) = 1, \end{cases} \tag{10}$$

*Shape Potential:* We call the term $\theta^S_{a;f(a)}$ as the shape potential since it influences the shape of the segmentation to resemble the object. The shape potential $\theta^S_{a;f(a)}$ is chosen such that, given $\mathbf{\Omega}$ (i.e. one possible localization of the object), the random variables corresponding to pixels that fall near to a detected object would be more likely to have foreground label (i.e. $l_0$) than random variables corresponding to pixels lying far from the object. It has the form:

$$\theta^S_{a;f(a)} = -\log \Pr(f(a)|\mathbf{\Omega}). \tag{11}$$

Following [7], [19], we choose to define $\Pr(f(a)|\mathbf{\Omega})$ as

$$\Pr(f(a)=0|\mathbf{\Omega}) \propto \frac{1}{1+\exp(\mu * \text{dist}(a,\mathbf{\Omega}))}, \;\; \Pr(f(a)=1|\mathbf{\Omega}) = 1 - \Pr(f(a)=0|\mathbf{\Omega}), \quad (12)$$

where $\text{dist}(a,\mathbf{\Omega})$ is the spatial distance of a pixel $a$ from the outline of the object defined by $\mathbf{\Omega}$ (being negative if inside the shape). The weight $\mu$ determines how much the pixels outside the shape are penalized compared to the pixels inside the shape.

Hence, the model $\mathbf{\Omega}$ contributes the unary term $\theta^S_{a;f(a)}$ for each pixel $a$ in the image for a labeling $f$ (see Fig. 2). Alternatively, $\mathbf{\Omega}$ can also be associated with the CDRF using pairwise terms as described in [13]. However, by reparameterizing the CDRF [18], both formulations can be shown to be equivalent. We prefer the use of unary terms since they do not effect the submodularity of the energy. Hence, it can easily be shown that the energy function $Q_2(f,\mathbf{\Omega};\mathbf{D},\boldsymbol{\theta})$ can be minimized via st-MINCUT [21]. Fig. 3 shows the advantage of introducing an object category model in the CDRF.

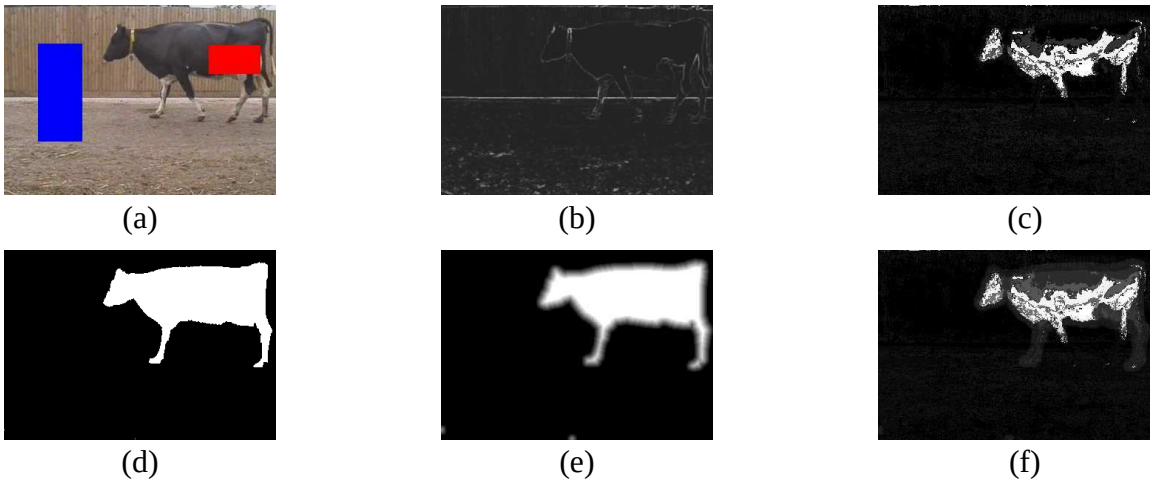



Fig. 3. **(a)** *An example cow image. The red and blue rectangles show the seed pixels which are used to learn the* RGB *distribution of foreground ($\mathcal{H}_{obj}$) and background ($\mathcal{H}_{bkg}$) respectively.* **(b)** *The pairwise terms (prior+contrast) for a pixel summed over its entire neighborhood. Pixels marked bright white indicate the presence of an image edge and are hence, more likely to define the segmentation boundary.* **(c)** *The unary potential ratio $\theta^1_{a;0}/\theta^1_{a;1}$ of a pixel computed using $\mathcal{H}_{obj}$ and $\mathcal{H}_{bkg}$. Pixels marked white are more likely to belong to foreground than the pixels marked black. Clearly, the likelihoods obtained using only* RGB *values are not sufficient to obtain a good segmentation.* **(d)** *The object category model $\mathbf{\Omega}$. White pixels are the points that lie inside the object while black pixels lie outside it.* **(e)** *The ratio $\theta^S_{a;0}/\theta^S_{a;1}$ corresponding to the model $\mathbf{\Omega}$. Again, pixels marked white are more likely to belong to foreground than background.* **(f)** *The ratio of the unary terms, i.e. $(\theta^A_{a;0}+\theta^S_{a;0})/(\theta^A_{a;1}+\theta^S_{a;1})$. Compared to (c), the unary terms in (f) provide more information about which pixels belong to the foreground and the background. Together with the pairwise terms shown in (b), this allows us to obtain a good segmentation of the object shown in (a).*

In this work we use two types of object category models: (i) For non-articulated objects, $\mathbf{\Omega}$

is represented by a set of exemplars; (ii) For articulated objects, $\mathbf{\Omega}$ is specified by our extension of the pictorial structures model [9], [12]. However, we note here that our methodology is completely general and could be combined with any sort of object category model.

The Object Category Specific CDRF framework defined above provides the probability of the labeling $f$ and the object category model $\mathbf{\Omega}$ as defined in equation (9). This is similar to the model used by Huang *et al.* [19] and Cremers *et al.* [7]. However, our approach differs from these works in the following respects: (i) in contrast to the level-sets based methods employed in [7], [19], we develop an efficient algorithm based on st-MINCUT; (ii) we do not require an accurate manual initialization to obtain the segmentation; and (iii) unlike [7], [19], we obtain the foreground-background labeling by maximizing the posterior probability $\Pr(f|\mathbf{D})$, instead of the joint probability $\Pr(f, \mathbf{\Omega}|\mathbf{D})$. In order to achieve this, we must integrate out $\mathbf{\Omega}$ i.e.

$$\Pr(f|\mathbf{D}) = \int \Pr(f, \mathbf{\Omega}|\mathbf{D}) d\mathbf{\Omega}. \tag{13}$$

The surprising result of this work is that this intractable looking integral can in fact be optimized by a simple and computationally efficient set of operations, as described in the next section.

## III. Roadmap of the Solution

We now provide a high-level overview of our approach. Given an image $\mathbf{D}$, the problem of segmentation requires us to obtain a labeling $f^*$ which maximizes the posterior probability $\Pr(f|\mathbf{D})$, i.e.

$$f^* = \arg\max \Pr(f|\mathbf{D}) = \arg\max \log \Pr(f|\mathbf{D}). \tag{14}$$

We have dropped the term $\boldsymbol{\theta}$ from the above notation to make the text less cluttered. We note however that there is no ambiguity about $\boldsymbol{\theta}$ for the work described in this paper (i.e. it always stands for the parameter of the Object Category Specific CDRF). In order to obtain realistic shapes, we would also like to influence the segmentation using an object category model $\mathbf{\Omega}$ (as described in the previous section). Given an Object Category Specific CDRF specified by one instance of $\mathbf{\Omega}$, the required posterior probability $\Pr(f|\mathbf{D})$ can be computed as

$$\Pr(f|\mathbf{D}) = \tfrac{\Pr(f, \mathbf{\Omega}|\mathbf{D})}{\Pr(\mathbf{\Omega}|f, \mathbf{D})}, \tag{15}$$

$$\Rightarrow \quad \log \Pr(f|\mathbf{D}) = \log \Pr(f, \mathbf{\Omega}|\mathbf{D}) - \log \Pr(\mathbf{\Omega}|f, \mathbf{D}), \tag{16}$$

where $\Pr(f, \mathbf{\Omega}|\mathbf{D})$ is given by equation (9) and $\Pr(\mathbf{\Omega}|f, \mathbf{D})$ is the conditional probability of $\mathbf{\Omega}$ given the image and its labeling. Note that we consider the log of the posterior probability

$\Pr(f|\mathbf{D})$. As will be seen, this allows us to marginalize the object category model $\mathbf{\Omega}$ using the generalized Expectation Maximization (generalized EM) [15] framework in order to obtaining the desired labeling $f^*$. By marginalizing $\mathbf{\Omega}$ we would ensure that the segmentation is not influenced by only one instance of the object category model (which may not localize the entire object correctly, leading to undesirable effects such as inaccurate segmentation).

We now describe the generalized EM framework which provides a natural way to deal with $\mathbf{\Omega}$ by treating it as missing (latent) data. The generalized EM algorithm starts with an initial estimate $f^0$ of the labeling and iteratively refines it by marginalizing over $\mathbf{\Omega}$. It has the desirable property that during each iteration the posterior probability $\Pr(f|\mathbf{D})$ does not decrease (i.e. the algorithm is guaranteed to converge to a local maximum). Given the current guess of the labeling $f'$, the generalized EM algorithm consists of two steps: (i) E-step: where the probability distribution $\Pr(\mathbf{\Omega}|f', \mathbf{D})$ is obtained; and (ii) M-step: where a new labeling $\hat{f}$ is computed such that $\Pr(\hat{f}|\mathbf{D}) \geq \Pr(f'|\mathbf{D})$. We briefly describe how the two steps of the generalized EM algorithm can be computed efficiently in order to obtain the segmentation. We subsequently provide the details for both the steps.

**Efficiently Computing the E-step:** Given the estimate of the labeling $f'$, we approximate the desired distribution $\Pr(\mathbf{\Omega}|f', \mathbf{D})$ by sampling efficiently for $\mathbf{\Omega}$. For non-articulated objects, this involves computing similarity measures at each location in the image. In § V-A, we show how this can be done efficiently. For the case of articulated objects, we develop an efficient sum-product belief propagation algorithm in § V-B, which efficiently computes the marginals for a non regular Potts model (i.e. when the labels are not specified by an underlying grid of parameters, complementing the result of Felzenszwalb and Huttenlocher [10]). As will be seen, these marginals allow us to efficiently sample the object category model using a method similar to [9].

**Efficiently Computing the M-step:** Once the samples of $\mathbf{\Omega}$ have been obtained in the E-step, we need to compute a new labeling $\hat{f}$ such that $\Pr(\hat{f}|\mathbf{D}) \geq \Pr(f'|\mathbf{D})$. We show that such a labeling $\hat{f}$ can be computed by minimizing a weighted sum of energy functions of the form given in equation (8), where the weighted sum over samples approximates the marginalization of $\mathbf{\Omega}$ (see below for details). The weights are given by the probability of the samples. This allows the labeling $\hat{f}$ to be obtained efficiently using a single st-MINCUT operation [21].

**Details:** We concentrate on the M-step first. We will later show how the E-step can be

approximately computed using image features. Given the distribution $\Pr(\mathbf{\Omega}|f', \mathbf{D})$, we average equation (16) over $\mathbf{\Omega}$ to obtain

$$\log \Pr(f|\mathbf{D}) = E(\log \Pr(f, \mathbf{\Omega}|\mathbf{D})) - E(\log \Pr(\mathbf{\Omega}|f, \mathbf{D})), \tag{17}$$

where $E(\cdot)$ indicates the expectation under $\Pr(\mathbf{\Omega}|f', \mathbf{D})$. The key observation of the generalized EM algorithm is that the second term on the right side of equation (17), i.e.

$$E(\log \Pr(\mathbf{\Omega}|f, \mathbf{D})) = \int (\log \Pr(\mathbf{\Omega}|f, \mathbf{D})) \Pr(\mathbf{\Omega}|f', \mathbf{D}) d\mathbf{\Omega} \tag{18}$$

is maximized when $f = f'$ [15]. We obtain a labeling $\hat{f}$ such that it maximizes the first term on the right side of equation (17), i.e.

$$\hat{f} = \arg\max E(\log \Pr(f, \mathbf{\Omega}|\mathbf{D})) = \arg\max \int (\log \Pr(f, \mathbf{\Omega}|\mathbf{D})) \Pr(\mathbf{\Omega}|f', \mathbf{D}) d\mathbf{\Omega}. \tag{19}$$

then, if $\hat{f}$ is different from $f'$, it is guaranteed to increase the posterior probability $p(f|\mathbf{D})$. This is due to the following two reasons: (i) $\hat{f}$ increases the first term of equation (17), i.e. $E(\log \Pr(\mathbf{\Omega}, f|\mathbf{D}))$, as it is obtained by maximizing this term; and (ii) $\hat{f}$ decreases the second term of equation (17), i.e. $E(\log \Pr(\mathbf{\Omega}|f, \mathbf{D}))$, which is maximized when $f = f'$. If $\hat{f}$ is the same as $f'$ then the algorithm is said to have converged to a local maximum of the distribution $\Pr(f|\mathbf{D})$. The expression in equation (19) is called the expected complete-data log-likelihood in the generalized EM literature.

In section V, it will be shown that we can efficiently sample from the object category model $\mathbf{\Omega}$ of our choice. This suggests a sampling based solution to maximizing equation (19). Let the set of $s$ samples be $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_s$, with weights $w_i = \Pr(\mathbf{\Omega}_i|f', \mathbf{D})$. Using these samples equation (19) can be approximated as

$$\hat{f} = \arg\max_f \sum_{i=1}^{i=s} w_i \log \Pr(f, \mathbf{\Omega}_i|\mathbf{D},$$

$$= \arg\max_f \textstyle\sum_{i=1}^{i=s} w_i(-Q_3(f, \mathbf{\Omega}_i; \mathbf{D})) - C, \tag{20}$$

$$\Rightarrow \quad \hat{f} = \arg\min_f \textstyle\sum_{i=1}^{i=s} w_i Q_3(f, \mathbf{\Omega}_i; \mathbf{D}) + C. \tag{21}$$

The form of the energy $Q_3(f, \mathbf{\Omega}_i; \mathbf{D})$ is given in equation (8). The term $C = \sum_i w_i \log Z_3(\boldsymbol{\theta})$ is a constant which does not depend on $f$ or $\mathbf{\Omega}$ and can be therefore be ignored during the minimization. *This is the key equation of our approach.* We observe that this energy function is a weighted linear sum of the energies $Q_3(f, \mathbf{\Omega}; \mathbf{D})$ which, being a linear combination with

positive weights $w_i$, can also be optimized using a single st-MINCUT operation [21] (see Fig. 4). This demonstrates the interesting result that for CDRF (and MRF) with latent variables, it is computationally feasible to optimize the complete-data log-likelihood.

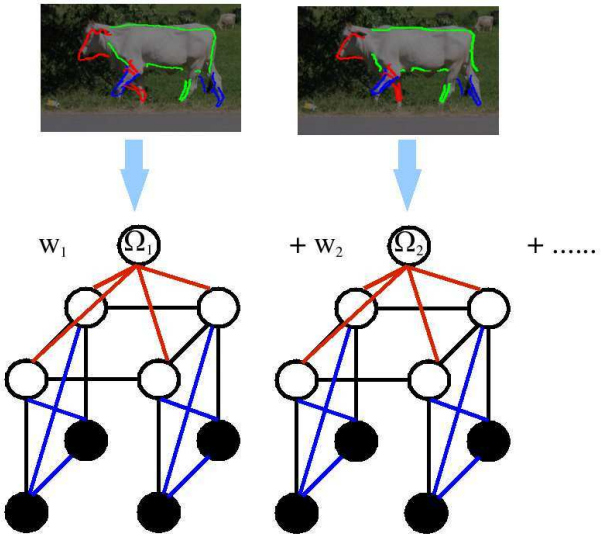



Fig. 4. *The top row shows various samples of a cow model for a given image. Each sample $\mathbf{\Omega}_i$ gives rise to one instance of the Object Category Specific* CDRF *which can be solved to obtain a segmentation using a single st-*MINCUT *operation on a graph, say $\mathcal{G}_i$. The segmentation which increases the expected complete-data log-likelihood is found using a single st-*MINCUT *operation on the weighted average of all graphs $\mathcal{G}_i$ where the weights $w_i$ are defined by the probability $\Pr(\mathbf{\Omega}_i|f', \mathbf{D})$ of the various samples.*

The generalized EM algorithm converges to a local maximum of $\Pr(f|\mathbf{D})$ and its success depends on the initial labeling $f^0$. In the last section a graphical model for pixel by pixel segmentation was set up. However, it would be computationally unsuccessful to use this model straight off. Rather, we adopt an initialization stage in which we get a rough estimate of the posterior probability of $\mathbf{\Omega}$ from a set of image features $\mathbf{Z}$ (defined in § IV-A). Image features (such as textons and edges) can provide high discrimination at low computational cost. We approximate the initial distribution $\Pr(\mathbf{\Omega}|f^0, \mathbf{D})$ as $g(\mathbf{\Omega}|\mathbf{Z})$, where $\mathbf{Z}$ are some image features chosen to localize the object in a computationally efficient manner. The weights $w_i$ required to evaluate equation (21) on the first EM iteration are obtained by sampling from the distribution $g(\mathbf{\Omega}|\mathbf{Z})$ (defined in section IV).

One might argue that if the MAP estimate of the object has a very high posterior probability compared to other poses, then equation (21) can be approximated using only the MAP estimate $\mathbf{\Omega}^*$ instead of the samples $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_s$. However, we found that this is not the case especially

when the RGB distribution of the background is similar to that of the object. For example, Fig. 15 shows various samples obtained by matching the model for a cow to two images. Note that different samples localize different parts of the object correctly and have similar posterior probabilities. Thus it is necessary to use multiple samples of the object category model.

The roadmap described above results in the OBJCUT algorithm, which obtains object category specific segmentation. Algorithms 1 and 2 summarize the main steps of OBJCUT for non-articulated and articulated object categories respectively. Note that we can keep on iterating over the E and the M steps. However, we observed that the samples (and the segmentation) obtained from one iteration to the next do not differ substantially (e.g. see Fig. 16). Hence, we run each step once for computational efficiency, as described in Algorithms 1 and 2. As will be seen in the experiments, we obtain accurate results for different object categories using only one iteration of the generalized EM algorithm.

In the remainder of the chapter, we provide details of the object category model $\mathbf{\Omega}$ of our choice. We propose efficient methods to obtain the samples from the posterior probability distribution of $\mathbf{\Omega}$ required for the marginalization in equation (21). We demonstrate the results on a number of articulated and non-articulated object categories.

## IV. OBJECT CATEGORY MODELS

When choosing the model $\mathbf{\Omega}$ for the Object Category Specific CDRF, two issues need to be considered: (i) whether the model can handle intra-class shape and appearance variation (and, in the case of articulated objects, spatial variation); and (ii) whether samples from the distribution $g(\mathbf{\Omega}|\mathbf{Z})$ (which are required for segmentation) can be obtained efficiently.

We represent the *shape* of an object (or a part, in the case of articulated objects) using multiple exemplars of the boundary. This allows us to handle the intra-class shape variation. The *appearance* of an object (or part) is represented using multiple texture exemplars. Again, this handles the intra-class appearance variation. Note that the exemplars model the shape and appearance of an object category. These should not be confused with the shape and appearance potentials of the Object Category Specific CDRF used to obtain the segmentation.

Once an initial estimate of the object is obtained, its appearance is known. Thus the localization of the object can be refined using a better appearance model (i.e. one which is specific to that instance of the object category). For this purpose, we use histograms which define the distribution of the RGB values of the foreground and the background.

- Input: An image **D** and a non-articulated object category model.
- Initial estimate of pose using edges and texture (§ V-A.1):
  1) A set of candidate poses $t_o = (x_o, y_o, \theta_o, \sigma_o)$ for the object is identified using a tree cascade of classifiers which computes a set of image features **Z**.
  2) The maximum likelihood estimate is chosen as initial estimate of the pose.
- Improved estimation of pose taking into account color (§ V-A.2):
  1) The appearance model of both foreground and background is updated.
  2) A new set of candidate poses is generated for the object by densely sampling pose space around the estimate found in the above step (again, using a tree cascade of classifiers for computing **Z**).
- The samples $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_s$ are obtained from the posterior $g(\mathbf{\Omega}|\mathbf{Z})$ of the object category model as described in § V-A.3.
- OBJCUT
  1) The weights $w_i = g(\mathbf{\Omega}_i|\mathbf{Z})$ are computed.
  2) The energy in equation (21) is minimized using a single st-MINCUT operation to obtain the segmentation $f$.

Algorithm 1: *The* OBJCUT *algorithm for non-articulated object categories.*

- Input: An image **D** and an articulated object category model.
- Initial estimate of pose using edges and texture (§ V-B.1):
  1) A set of candidate poses $t_i = (x_i, y_i, \theta_i, \sigma_i)$ for each part is identified using a tree cascade of classifiers which computes a set of image features **Z**.
  2) An initial estimate of the poses of the parts is found without considering the layering of parts using an efficient sum-product BP algorithm (described in the Appendix).
- Improved estimation of pose taking into account color and occlusion (§ V-B.2):
  1) The appearance model of both foreground and background is updated.
  2) A new set of candidate poses is generated for each part by densely sampling pose space around the estimate found in the above step (again, using a tree cascade of classifiers for computing **Z**).
  3) The pose of the object is estimated using efficient sum-product BP and the layering of the parts.
- The samples $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_s$ are obtained from the posterior $g(\mathbf{\Omega}|\mathbf{Z})$ of the object category model as described in § V-B.3.
- OBJCUT
  1) The weights $w_i = g(\mathbf{\Omega}_i|\mathbf{Z})$ are computed.
  2) The energy in equation (21) is minimized using a single st-MINCUT operation to obtain the segmentation $f$.

Algorithm 2: *The* OBJCUT *algorithm for articulated object categories.*

We define the model $\mathbf{\Omega}$ for non-articulated objects as a set of shape and texture exemplars (see Fig. 5). In the case of articulated objects, one must also allow for considerable spatial variation. For this purpose, we use the pictorial structures (PS) model. However, the PS models used in previous work [9], [12] assume non-overlapping parts connected in a tree structure. We extend the PS by incorporating the layering information of parts and connecting them in a complete

graph structure. We call the resulting representation the *layered pictorial structures* (LPS) model. Below, we describe the object category models in detail.

### A. Set of Exemplars Model

We represent non-articulated object categories as 2D patterns with a probabilistic model for their shape and appearance. The shape of the object category is represented using a set of shape exemplars $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_e\}$. For this work, each shape exemplar $\mathbf{S}_i$ is given by a set of points $\{s_{i;1}, s_{i;2}, \cdots, s_{i;m}\}$ describing the outline of the object. Similarly, the appearance is represented using a set of texture exemplars $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \cdots, \mathbf{T}_e\}$, where each exemplar is an image patch (i.e. a set of intensity values). Note that we use multiple exemplars (i.e. $e > 1$) to handle the shape and appearance variations which are common in non-articulated object categories. We call this the set of exemplars (SOE) model. Note that similar models were used for object detection in [14], [40], [42].

*1) Feature likelihood for object:* Given the putative pose of an object, i.e. $\mathbf{t}_o = \{x_o, y_o, \phi_o, \rho_o\}$ (where $\{x_o, y_o\}$ is the location, $\phi_o$ is the rotation and $\rho_o$ is the scale), we computed two features $\mathbf{Z} = \{z_1, z_2\}$ for the shape and appearance of the object respectively. Let $\mathbf{D}_o \subseteq \mathbf{D}$ be the set of pixels corresponding to the object at pose $\mathbf{t}_o$. The features $z_1$ and $z_2$ are computed using $\mathbf{D}_o$. Assuming independence of the two features, the likelihood based on the whole data is approximated as

$$\Pr(\mathbf{Z}|\mathbf{\Omega}) = \Pr(z_1)\Pr(z_2) \tag{22}$$

where $\Pr(z_1) \propto \exp(-z_1)$ and $\Pr(z_2) \propto \exp(-z_2)$. We also assume the prior $\Pr(\mathbf{\Omega})$ to be uniform. This provides us with the distribution $g(\mathbf{\Omega}|\mathbf{Z})$ as

$$g(\mathbf{\Omega}|\mathbf{Z}) \propto \Pr(\mathbf{Z}|\mathbf{\Omega})\Pr(\mathbf{\Omega}) \propto \Pr(\mathbf{Z}|\mathbf{\Omega}). \tag{23}$$

We now describe the features $z_1$ and $z_2$ in detail.

**Outline ($z_1$):** The likelihood of the object shape should be robust to outliers resulting from cluttered backgrounds. To this end, we define $z_1$ as the minimum of the truncated chamfer distances over all the exemplars of the object at pose $\mathbf{t}_o$. Let $\mathbf{U} = \{u_1, u_2, \cdots, u_m\}$ represent the edges of the image at $\mathbf{t}_o$. Then $z_1$ is computed as $z_1 = \min_{\mathbf{S}_i \in \mathcal{S}} d_{cham}(\mathbf{S}_i, \mathbf{U})$. The truncated chamfer distance $d_{cham}(\cdot,\cdot)$ is given by $d_{cham}(\mathbf{S}_i, \mathbf{U}) = \frac{1}{m}\sum_j \min\{\min_k ||u_k - s_{i;j}||, \tau_1\}$, where $\tau_1$ is a threshold for truncation which reduces the effect of outliers and missing edges. Orientation

information is included by computing $\min_k ||u_k - s_{i;j}||$ only over those edge points $u_k$ which have a similar orientation to $s_{i;j}$. This makes the chamfer distance more robust [14]. We use 8 orientation groups for the outline points.

**Texture ($z_2$):** We use the VZ classifier [43] which provides a histogram representation $\mathcal{H}_i$ for each exemplar $\mathbf{T}_i$[1]. It also provides a histogram $\mathcal{H}_o$ for the image patch $\mathbf{D}_o$. The feature $z_2$ is computed as $z_2 = \min_{\mathbf{T}_i \in \mathcal{T}} d_{chi}(\mathcal{H}_i, \mathcal{H}_o)$, where $d_{chi}(\cdot,\cdot)$ is the $\chi^2$ distance[2].

*2) Learning the exemplars:* In order to learn the exemplars, we use manually segmented images. The outline of each segmented image provides us with an exemplars $\mathbf{S}_i \in \mathcal{S}$. The texture exemplars $\mathbf{T}_i$ are given by the subimage marked as foreground. We use 20 segmented images each for the 'banana' and the 'orange' categories. A subset of the shape exemplars $\mathcal{S}$ of these two categories is shown in Fig. 5.

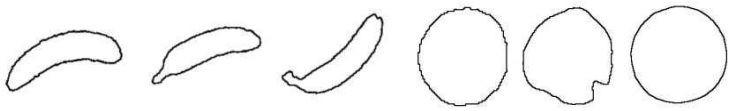


Fig. 5. *A selection of the multiple exemplars used to represent the model for bananas and oranges. Multiple shape exemplars are required to handle intra-class shape variability.*

We now describe an extension to the PS which is used as the model $\mathbf{\Omega}$ for articulated objects.

### *B. Layered Pictorial Structures*

In the case of articulated objects, we use the PS model to handle large deformations. PS are compositions of 2D patterns, termed *parts*, under a probabilistic model for their shape, appearance and spatial layout. However, the PS models used previously in [9], [12] are not directly suitable for applications such as efficient segmentation due to the following reasons: (i) they use a weak likelihood model which results in a large number of putative poses for each part; (ii) the parts are connected in a tree structure and hence, provide a weak spatial model; and (iii) they do not explicitly model self-occlusion. Hence, different parts with similar shape and appearance (e.g.

[1]The VZ classifier obtains a texton dictionary by clustering intensity values in an $N \times N$ neighborhood of each pixel in $\mathbf{T}_i$ for all $\mathbf{T}_i \in \mathcal{T}$. The histogram $\mathcal{H}_i$ is given by the frequency of each entry of this texton dictionary in $\mathbf{T}_i$. We use $N = 3$ and 60 clusters in our experiments.

[2]The feature $z_2$ described here handles the intra-class variation in appearance and is used to determine an initial estimate of the pose of the object. This estimate is then refined using a better appearance model (i.e. specific to a particular instance of the object category) as described in § V-A.2.

the legs of cows or horses) are often incorrectly detected at the same pose (i.e. even in cases where they are actually at different poses in the given image).

We overcome the deficiencies of previous PS models by extending them in three ways: (i) similar to SOE, the likelihood of a part includes both its outline and its texture which results in a small number of putative poses for each part in a given image (see § V-B.1); (ii) all parts are connected to each other to form a complete graph instead of a tree structure which provides a better spatial model; and (iii) similar to the model described in [1], each part $p_i$ is assigned an occlusion number $o_i$ which determines its relative depth. The occlusion numbers allow us to explicitly model self-occlusion. Specifically, a part $p_i$ can partially or completely occlude part $p_j$ if and only if $o_i > o_j$. Note that several parts can have the same occlusion number if they are at the same depth. Such parts, which share a common occlusion number, are said to lie in the same layer. We call this model layered pictorial structures (LPS).

*1) Posterior of the* LPS*:* An LPS can also be viewed as an MRF where the random variables of the MRF correspond to the $n_P$ parts. Each random variable takes one of $n_L$ labels which encode the putative poses of the part. Similar to the pose of an object described in § IV-A, the pose of the $i^{th}$ part is defined by a label $\mathbf{t}_i = \{x_i, y_i, \phi_i, \rho_i\}$. For a given pose $\mathbf{t}_i$ and image $\mathbf{D}$, the part $p_i$ corresponds to the subset of the image pixels $\mathbf{D}$ which are used to calculate features $\mathbf{Z}_i$.

The posterior of the LPS is given by

$$g(\mathbf{\Omega}|\mathbf{Z}) = \Pr(\mathbf{Z}|\mathbf{\Omega}) \Pr(\mathbf{\Omega}), \tag{24}$$

where $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_{n_P}\}$ are the image features, $p(\mathbf{Z}|\mathbf{\Omega})$ is the feature likelihood and $p(\mathbf{\Omega})$ is the prior. Like the SOE model, the shape of an LPS is specified by a set of shape exemplars $\mathcal{S}_i$ for each part $p_i$. The appearance of an LPS is modelled using a set of texture exemplars $\mathcal{T}$ for the object category. Note that unlike the shape exemplars, which are specific to a part of an object category, the texture exemplars are specific to the object category of interest. Assuming that the features $\mathbf{Z}_i$ are computed by not including pixels accounted for by parts $p_j$ for which $o_j > o_i$ (i.e. parts which can occlude $p_i$), the feature likelihood is given by $\Pr(\mathbf{Z}|\mathbf{\Omega}) = \prod_{i=1}^{i=n_P} \Pr(\mathbf{Z}_i|\mathbf{\Omega})$. The feature likelihood $\Pr(\mathbf{Z}_i|\mathbf{\Omega})$ for part $p_i$ is computed as described in § IV-A.1. Specifically, the likelihood of the first feature, i.e. $\Pr(z_1)$, is computed using the minimum of the truncated chamfer distance, over the set $\mathcal{S}_i$ for the part, at pose $\mathbf{t}_i$. The texture likelihood, $p(z_2)$, is obtained

from the VZ classifier using the set $\mathcal{T}$ for the object category[3].

LPS, like PS, are characterized by pairwise only dependencies between the random variables. These are modelled as a prior on the relative poses of parts:

$$\Pr(\mathbf{\Omega}) \propto \exp\left(-\sum_{i=1}^{i=n_P}\sum_{j=1,j\neq i}^{j=n_P}\alpha(\mathbf{t}_i,\mathbf{t}_j)\right). \tag{25}$$

Note that we use a completely connected MRF as this was found to provide a better localization of the object than a tree structured MRF [22]. In our approach, the pairwise potentials $\alpha(\mathbf{t}_i,\mathbf{t}_j)$ of putative poses for each pair of parts are given by a non-regular Potts model, i.e.

$$\alpha(\mathbf{t}_i,\mathbf{t}_j) = \begin{cases} d_1 & \text{if valid configuration} \\ d_2 & \text{otherwise,} \end{cases} \tag{26}$$

where $d_1 < d_2$. In other words, all valid configurations are considered equally likely and have a smaller cost. A configuration is considered valid if the difference between the two poses $\mathbf{t}_i$ and $\mathbf{t}_j$ lies in an interval defined by $\mathbf{t}_{ij}^{min} = \{x_{ij}^{min}, y_{ij}^{min}, \theta_{ij}^{min}, \sigma_{ij}^{min}\}$ and $\mathbf{t}_{ij}^{max} = \{x_{ij}^{max}, y_{ij}^{max}, \theta_{ij}^{max}, \sigma_{ij}^{max}\}$, i.e. $\mathbf{t}_{ij}^{min} \leq |\mathbf{t}_i - \mathbf{t}_j| \leq \mathbf{t}_{ij}^{max}$. Note that the above inequalities should be interpreted component-wise (i.e. $x_{ij}^{min} \leq |x_i - x_j| \leq x_{ij}^{max}$ and so on). For each pair of parts $p_i$ and $p_j$ the terms $\mathbf{t}_{ij}^{min}$ and $\mathbf{t}_{ij}^{max}$ are learnt using training video sequences as described in § IV-B.2. Using equation (24), the posterior of the LPS parameters is given by

$$g(\mathbf{\Omega}|\mathbf{Z}) \propto \prod_{i=1}^{i=n_P} \Pr(\mathbf{Z}_i|\mathbf{\Omega}) \exp\left(-\sum_{j\neq i}\alpha(\mathbf{t}_i,\mathbf{t}_j)\right). \tag{27}$$

*2) Learning the* LPS*:* We now describe how we learn the various parameters of the LPS model for cows. To this end, we use 20 cow videos of 45 frames each and learn the shape, appearance and transformations of rigidly moving segments in each frame of the video using the motion segmentation method described in [24]. Correspondence between the segments learnt from two different videos is established using shape context with continuity constraints [40] as shown in Fig. 6. The corresponding segments then define a part of the LPS model. The outline of the segments defines the shape exemplars $\mathcal{S}_i$ (see Fig. 7), while the intensity values of the segmented cows provides the set $\mathcal{T}$. Furthermore, an estimate of $|\mathbf{t}_i - \mathbf{t}_j|$ is also obtained (after rescaling

[3]Again, the feature $z_2$ described here handles the intra-class variation in appearance and is used to determine an initial estimate of the pose of the object. This estimate is then refined using a better appearance model (i.e. specific to a particular instance of the object category) as described in § V-B.2.

the frames of the video such that the width of the cows is $230$ pixels), for each frame and for all pairs of parts $p_i$ and $p_j$. This is used to compute the parameters $\mathbf{t}_{ij}^{min}$ and $\mathbf{t}_{ij}^{max}$ that define valid configurations.

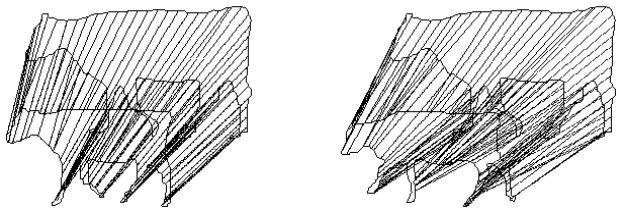


Fig. 6. *Correspondence using shape context matching with continuity constraints. Outlines of two cows which need to be matched are shown. Lines are drawn to indicate corresponding points.*

Fig. 7. *The first row shows a subset of shape exemplars $\mathcal{S}_i$ for the head of a cow (obtained by establishing a correspondence between a set of segmented cow images as shown in Fig. 6). The second row shows shape exemplars of the torso part.*

To obtain the LPS model for horses, we use 20 manually segmented images. The texture exemplars can be obtained using the segmented images. However, since these images do not provide us with any motion information, we cannot use the method in [24] to obtain the shape exemplars of the LPS model. In order to overcome this problem, we establish a point to point correspondence between the outline of a cow from a training video and the outlines of the horses, again using shape context with continuity constraints [40] (see Fig. 8). Using this correspondence and the learnt parts of the cow, the parts of the horse are now easily determined (see Fig. 9). The part correspondence obtained also maps the parameters $\mathbf{t}_{ij}^{min}$ and $\mathbf{t}_{ij}^{max}$ that were learnt for cows to horses. In the next section, we address the important issue of developing efficient algorithms

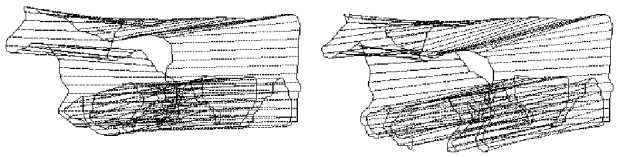


Fig. 8. *Correspondence using shape context matching with continuity constraints. Outlines of a horse and a cow are shown. Lines are drawn to indicate corresponding points.*

for matching the model $\mathbf{\Omega}$ to an image.

Fig. 9. *The first and second row show the multiple exemplars of the head and the torso part respectively. The exemplars are obtained by establishing a correspondence between segmented images of cows and horses as shown in Fig. 8.*

## V. Sampling the Object Category Models

Given an image $\mathbf{D}$, our objective is to match the object category model to it in order to obtain samples from the distribution $g(\mathbf{\Omega}|\mathbf{Z})$. We achieve this in three stages:

- *Initialization*, where we fit the object category model to a given image $\mathbf{D}$ by computing features $z_1$ (i.e. chamfer) and $z_2$ (i.e. texture) using exemplars. This provides us with a rough object pose.
- *Refinement*, where the initial estimate is refined by computing $z_2$ using a better appearance model (i.e. the RGB distribution for the foreground and background learnt using the initial pose together with the shape) instead of the texture feature used during initialization. In the case of articulated objects, the layering information is also used.
- *Sampling*, where samples are obtained from the distribution $g(\mathbf{\Omega}|\mathbf{Z})$.

### A. Sampling the SOE

We now describe the three stages for obtaining samples by matching the SOE model (for a non-articulated object category) to a given image.

*1) Initial estimation of pose:* In order to obtain the initial estimate of the pose of an object, we need to compute the feature likelihood for each pose using all exemplars. This would be computationally expensive due to the large number of possible poses and exemplars. However, most poses have a very low likelihood since they do not cover the pixels containing the object of interest. We require an efficient method which discards such poses quickly. To this end, we use a *tree cascade of classifiers* [39].

We term the rotated and scaled versions of the shape exemplars as *templates*. When matching many similar templates to an image, a significant speed-up is achieved by forming a template hierarchy and using a coarse-to-fine search. The idea is to group similar templates together with an estimate of the variance of the error within the cluster, which is then used to define a matching threshold. For each cluster, a prototype of the cluster is first compared to the image; the

individual templates within the cluster are compared to the image only if the error is below the threshold. This clustering is done at various levels, resulting in a hierarchy, with the templates at the leaf level covering the space of all possible templates (see Fig. 10).

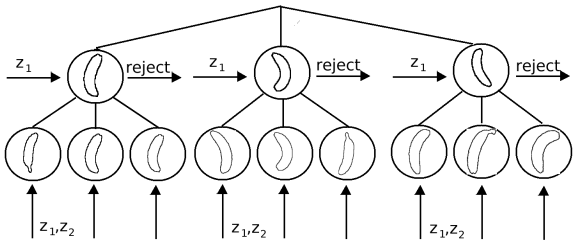
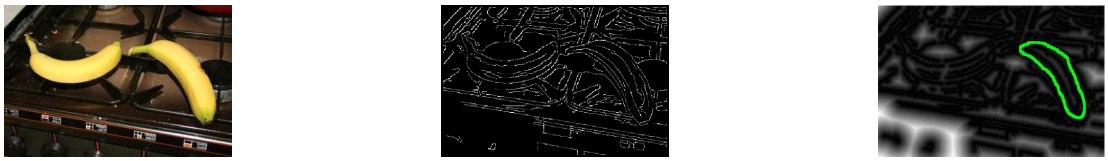



Fig. 10. *The putative poses of the object, e.g. a banana, together with their likelihood are found using a cascade of classifiers.*

In our experiments, we constructed a 3-level tree by clustering the templates using a cost function based on chamfer distance. We use 20 exemplars for each object. The templates are generated by transforming the exemplars using discrete rotations between $-\pi/4$ and $\pi/4$ radians in intervals of $0.1$ radians and scales between $0.7$ and $1.3$ in intervals of $0.1$.

The edge image of **D** is found using edge detection with embedded confidence [28] (a variation on Canny in which a confidence measure is computed from an ideal edge template). The feature $z_1$ (truncated chamfer distance) is computed efficiently by using a distance transform of the edge image that is further filtered as suggested in [30]. This transformation assigns to each pixel in the edge image, the minimum of $\tau_1$ and the distance to its nearest edge pixel. The truncated chamfer distance of an exemplar at an image pose $\mathbf{t}_o = \{x_o, y_o, \phi_o, \rho_o\}$ is calculated efficiently as the mean of the distance transform values at the template point coordinates (using the template defined by rotation $\phi_o$ and scale $\rho_o$ of the exemplar, see Fig. 11).



(a) (b) (c)

Fig. 11. **(a)** *Original image containing bananas in a cluttered scene.* **(b)** *Edgemap of the image.* **(c)** *The distance transform of the edgemap along with an exemplar of banana. Brighter intensity values indicate points which are far away from the edges. Truncated chamfer distance is calculated as the mean of the distance transform values at the exemplar point coordinates.*

The feature $z_2$ (i.e. texture) is computed only at level 3 of the tree cascade by determining the nearest neighbor of the histogram of texton labeling of $\mathbf{D}_o$ among the histograms of texture

exemplars. For this purpose, we use the efficient nearest neighbor method described in [17] (modified for $\chi^2$ distance instead of Euclidean distance).

Associated with each node of the cascade is a threshold used to reject bad poses. The putative poses $\mathbf{t}_o$ of the object are found by rejecting bad poses by traversing through the tree cascade starting from the root node for each pixel $\{x, y\}$ of the image $\mathbf{D}$. The likelihoods $\Pr(\mathbf{Z}|\mathbf{\Omega})$ are computed using equation (22). The initial estimate of the pose is determined by the image location $\{x_o, y_o\}$, template orientation $\phi_o$ and template scale $\rho_o$ which results in the highest likelihood. Fig. 12 (column 1) shows the initial estimate for two banana images. This estimate is refined using a better appearance model as described below.

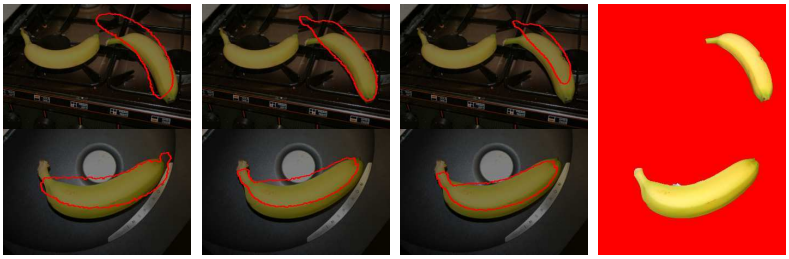


Fig. 12. *The first column shows the initial estimate obtained for the pose of a banana in two images (see § V-A.1). Samples of the model obtained using the* RGB *distribution of foreground and background are shown in the second and third column (see § V-A.3). The detected poses are shown overlaid on the image. The fourth column shows the segmentation obtained.*

*2) Refinement of pose:* Once the initial estimate of the pose of the object is obtained, the object location is used to estimate the RGB distribution of the foreground and background (and texture exemplars are no longer used). These distributions, denoted as $\mathcal{H}_{obj}$ and $\mathcal{H}_{bkg}$ for the foreground and background respectively, are used to define a better appearance feature $z_2$, which is specific to the particular instance of the object category in the image. Specifically,

$$\Pr(z_2) = \prod_{\mathbf{x} \in \mathbf{D}_o} \frac{\Pr(\mathbf{x}|\mathcal{H}_{obj})}{\Pr(\mathbf{x}|\mathcal{H}_{bkg})}. \tag{28}$$

The refined estimate of the putative poses are obtained using the tree cascade of classifiers as described in § V-A.1 by searching around the initial estimate. In our experiments, we consider locations $\{x, y\}$ which are at most at a distance of 15% of the size of the object as given by the initial estimate. When obtaining the refined estimate, all orientations $\phi$ and scales $\rho$ are considered at each location $\{x, y\}$.

*3) Obtaining samples of the* SOE*:* We now obtain samples from the distribution $g(\mathbf{Z}|\mathbf{\Omega})$ for the SOE model. By assuming a uniform prior $\Pr(\mathbf{\Omega})$ for the model parameter $\mathbf{\Omega}$, this distribution

is given by $g(\mathbf{Z}|\mathbf{\Omega}) \propto \Pr(z_1)\Pr(z_2)$. The samples are defined as the best $s$ matches found in § V-A.2 and are obtained by simply sorting over the various matches at all possible locations of the image $\mathbf{D}$. Fig. 12 (second and third column) shows some of the samples obtained using the above method for two banana images.

Next, we describe how to sample the distribution $g(\mathbf{Z}|\mathbf{\Omega})$ for an LPS model in the case of articulated object categories.

### *B. Sampling the* LPS

When matching the LPS model to the image, the number of labels $n_L$ per part has the potential to be very large. Consider the discretization of the putative poses $\mathbf{t} = \{x, y, \phi, \rho\}$ into $360 \times 240$ for $\{x, y\}$ with 15 orientations and 7 scales at each location. This results in 9,072,000 poses which causes some computational difficulty when obtaining the samples of the LPS.

Felzenszwalb and Huttenlocher [9] advocate maintaining all labels and suggest an $O(n_P n_L)$ algorithm for finding the samples of the PS by restricting the form of the prior $\exp(-\alpha(\mathbf{t}_i, \mathbf{t}_j))$ in equation (25). In their work, priors are specified by normal distributions. However, this approach would no longer be computationally feasible as the number of parameters used to represent a pose $\mathbf{t}_i$ increase (e.g. 6 parameters for affine or 8 parameters for projective).

In our approach, we consider the same amount of discretization as in [9] when we are finding candidate poses. However, as noted in § IV-B, using discriminative features for shape and appearance of the object allows us to consider only a small number of putative poses, $n_L$, per part by discarding the poses with low likelihood. We found that using a few hundred poses per part, instead of the millions of poses used in [9], was sufficient. The samples are found by a novel efficient algorithm of complexity $O(n_P n'_L)$ per iteration (where $n'_L \ll n_L^2$) which generalizes the method described in [10] to non-regular Potts model. Our approach is efficient even for affine and projective transformations due to the small number of putative poses $n_L$. We now described the three stages for obtaining samples of the LPS.

*1) Initial estimation of poses:* We find the initial estimate of the poses of the LPS for an image **D** by first obtaining the putative poses for each part (along with the corresponding likelihoods) and then estimating posteriors of the putative poses. Note that we do not use occlusion numbers of the parts during this stage.

The putative poses are found using a tree cascade of classifiers for each part as described in § V-A.1 (see Fig. 13). The first feature $z_1$ is computed using a 3-level tree cascade of classifiers

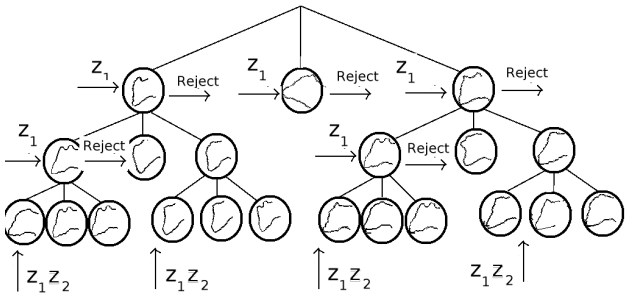



Fig. 13. *The putative poses of a part, e.g. the head, together with their likelihood are found using a cascade of classifiers. Similar to the cascade shown in Fig. 10, a tree structure is used to prune away the bad poses. The texture (i.e. $z_2$) is measured only at the last level of the tree.*

for each part. Similar to the first stage of matching the SOE model, the appearance feature $z_2$ is computed using texture exemplars $\mathcal{T}$ of the object category at the third level of the tree cascade. Note that at this stage the RGB distributions $\mathcal{H}_{obj}$ and $\mathcal{H}_{bkg}$ for the foreground and background are not known. Hence, the feature $z_2$ is computed using only texture exemplars to overcome intra-class variation in appearance.

Next, an initial estimate of the model is obtained by computing the marginals of the putative poses. Note that, unlike the SOE model, LPS provides a prior over the relative poses of the parts which needs to be considered while computing the marginals. The pose of each part in the initial estimate is given by the putative pose which has the highest marginal probability.

We use sum-product belief propagation (sum-product BP) to find the marginal probability of part $p_i$ taking a label $\mathbf{t}_i$. Recall that the time complexity of sum-product BP is $O(n_P n_L^2)$ per iteration which makes it inefficient for large $n_L$. However, we take advantage of the fact that the pairwise potentials of the LPS are given by a non-regular Potts model (as shown in equation (26)). This allows us to reduce the time complexity to $O(n_P n'_L)$ per iteration, where $n'_L \ll n_L^2$, using the efficient sum-product BP algorithm described in the Appendix.

The beliefs for each part $p_i$ and putative pose $\mathbf{t}_i$ computed using sum-product BP (denoted by $b_i(\mathbf{t}_i)$) allow us to determine the MMSE (minimum mean squared error) estimate of the poses of the parts (by choosing the pose with the highest belief). In addition, it also allows us to compute the beliefs for putative poses of every pair of parts, i.e. $b_{ij}(\mathbf{t}_i, \mathbf{t}_j)$, which is later used for sampling (see section V-B.3). Since the parts are connected to form a complete graph, we tend to find valid configurations of the object. Fig. 14 (column 1) shows the initial estimate for two cow images. Note that the occlusion numbers are not used to obtain the initial estimate,

as would be the case when using the PS model instead of the LPS model. Hence the half-limbs (which are similar to each other in shape and appearance) tend to overlap. The initial estimate is refined using the layering as described below.

Fig. 14. *The first column shows the initial estimate obtained for poses of parts of a cow in two images (see § V-B.1). The half-limbs tend to overlap since occlusion numbers are not used. Refined estimates of the poses obtained using the* RGB *distribution of foreground and background together with the* LPS *model are shown in the second column (see § V-B.2). The parts are shown overlaid on the image. The third column shows the segmentation obtained using the* OBJCUT *algorithm.*

*2) Layerwise refinement:* Using the initial estimate of the object obtained above, the RGB distribution of the foreground (i.e. $\mathcal{H}_{obj}$) and the background (i.e. $\mathcal{H}_{bkg}$) is estimated. A better appearance feature $z_2$ (i.e. specific to the particular instance of the object category in the images) is now computed as shown in equation (28). The refined estimate of the poses are obtained by compositing the parts of the LPS in descending order of their occlusion numbers as follows. When considering the layer with occlusion number $o$, putative poses of the parts $p_j$ such that $o_j = o$ are found using the tree cascade of classifiers around the initial estimate of $p_j$. In our experiments, we consider locations $\{x, y\}$ which are at most at a distance of 15% of the size of the part as given by the initial estimate. At each location, all possible orientations $\phi$ and scales $\rho$ are considered. When computing the likelihood of the part at a given pose, pixels which have already been accounted for by a previous layer are not used. Again, the beliefs of each putative pose of every part is computed using efficient sum-product BP. Fig. 14 (column 2) shows the MMSE estimate obtained using layerwise refinement for two cow images. However, for segmentation we are interested in samples of the LPS which are obtained in the third stage.

*3) Obtaining samples of the* LPS*:* We describe the method for sampling by considering only 2 layers (called layer 1 and layer 2). The extension to an arbitrary number of layers is trivial. The basic idea is to sample the parts in descending order of their occlusion numbers. In our case, this would imply that we sample the parts from layer 2 before we sample the parts from layer 1 (since layer 2 can occlude layer 1). Although this method is not optimal, it produces

useful samples for segmentation in practice. To obtain a sample $\mathbf{\Omega}_i$, parts belonging to layer 2 are considered first. The beliefs of these parts are computed using efficient sum-product BP. The posterior for sample $\mathbf{\Omega}_i$ is approximated as

$$g(\mathbf{\Omega}_i|\mathbf{Z}) = \frac{\prod_{ij} b_{ij}(\mathbf{t}_i, \mathbf{t}_j)}{\prod_i b_i(\mathbf{t}_i)^{q_i - 1}}, \tag{29}$$

where $q_i$ is the number of neighboring parts of $p_i$. Since we use a complete graph, $q_i = n_P - 1$, for all parts $p_i$. Note that the posterior is exact only for a singly connected graph. However, using this approximation sum-product BP has been shown to converge to stationary points of the Bethe free energy [46].

The posterior is then sampled for poses, one part at a time (i.e. Gibbs sampling), such that the pose of the part being sampled forms a valid configuration with the poses of the parts previously sampled. The process is repeated to obtain multiple samples $\mathbf{\Omega}_i$ (which do not include the poses of parts belonging to layer 1). This method of sampling is efficient since often very few pairs of poses form a valid configuration. Further, these pairs are pre-computed during the efficient sum-product BP algorithm as described in the Appendix. The best $n_S$ samples, with the highest belief, are chosen.

To obtain the poses of parts in layer 1 for sample $\mathbf{\Omega}_i$, we fix the poses of parts belonging to layer 2 as given by $\mathbf{\Omega}_i$. We calculate the posterior over the poses of parts in layer 1 using sum-product BP. We sample this posterior for poses of parts such that they form a valid configuration with the poses of the parts in layer 2 and with those in layer 1 that were previously sampled. As in the case of layer 2, multiple samples are obtained and the best $n_S$ samples are chosen. The process is repeated for all samples $\mathbf{\Omega}_i$ for layer 2, resulting in a total of $s = n_S^2$ samples.

However, computing the likelihood of the parts in layer 1 for each $\mathbf{\Omega}$ is expensive as their overlap with parts in layer 2 needs to be considered. We use an approximation by considering only those poses whose overlap with layer 2 is below a threshold $\tau_2$. Fig. 15 shows some of the samples obtained using the above method for the cows in Fig. 14.

Once the samples are obtained, they are used as inputs for the OBJCUT algorithm which provides the segmentation. Note that the segmentation can then be used to obtain more accurate samples (i.e. the generalized EM algorithm can be iterated until convergence). However, we observed that in almost all cases, the samples obtained in the second iteration (and hence, the segmentation) did not differ significantly from the samples in the first iteration (e.g. see Fig. 16).

Fig. 15. *Each row shows three samples obtained by matching the* LPS *model of a cow to an image. Beliefs over the putative poses of parts are calculated using sum-product* BP. *The resulting posterior probability is then sampled to obtain instances of the object (see § V-B.3). Note that different half-limbs are detected correctly in different samples.*

Hence, we ran the generalized EM algorithm for only one iteration for all the images. As shown in the next section, even using a single iteration provides accurate segmentation for a large number of object categories.

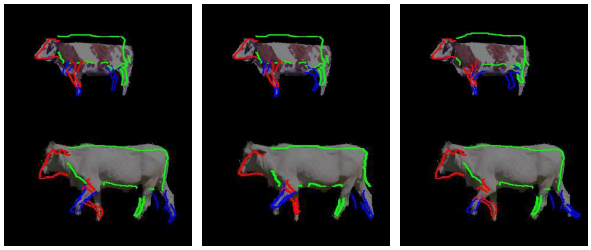


Fig. 16. *Samples found in the second iteration of the generalized* EM *algorithm. The samples were obtained using the method described above, where the features* **Z** *were computed using foreground (i.e. non-black) pixels only. Note that the samples are the same as those shown in Figure 15. This implies that the same segmentation will be obtained in the second iteration. Hence, we stop the generalized* EM *algorithm after one iteration for computational efficiency.*

## VI. RESULTS

We present several results of the OBJCUT algorithm and compare it with a state of the art method and ground truth. In all our experiments, we used the same weight values. As will be seen, OBJCUT provides reliable segmentation by incorporating both: (i) modelled deformations, using a set of exemplars model for non-articulated objects and the LPS model for articulated objects; and (ii) unmodelled deformations, by merging pixels surrounding the detected object into the segmentation via an st-MINCUT operation.

The results for non-articulated objects are shown for two categories: bananas and oranges. Fig. 12 (column 4) shows the results of the OBJCUT algorithm for two banana images. Fig. 17 show the segmentations obtained for images containing oranges. As can be seen, the samples of the SOE model correctly localize the object in the image. The distinctive shape and appearance of the object then allows us to obtain an accurate segmentation using a single st-MINCUT operation. Note that even though it may appear that object categories such as fruits can be easily segmented

using color information alone, we found that the shape potential introduced by the model $\Omega$ significantly improves the segmentation accuracy (see Fig. 17, top row).

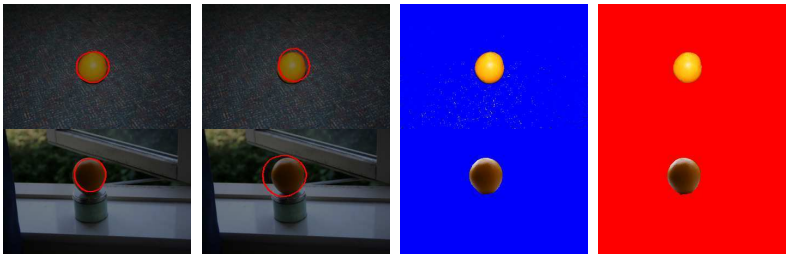


Fig. 17. *Image segmentation results 1. The* SOE *model for oranges was used to obtain segmentations of previously unseen images. The first two images in each column show some of the samples of the* SOE *model. The segmentation obtained without using the shape potential is shown in the third column and is clearly inferior to the results obtained by the* OBJCUT *algorithm (fourth column). In particular, the segmentation of the top image is highly* speckled *since the background also contains orange-colored pixels. In contrast, the* OBJCUT *algorithm segments both the images accurately by also using shape information.*

We also tested the OBJCUT algorithm on two articulated object categories: cows and horses. Fig 14 (column 3) shows the results of our approach for two cow images. Fig. 18 and Fig. 19 show the segmentation of various images of cows and horses respectively.

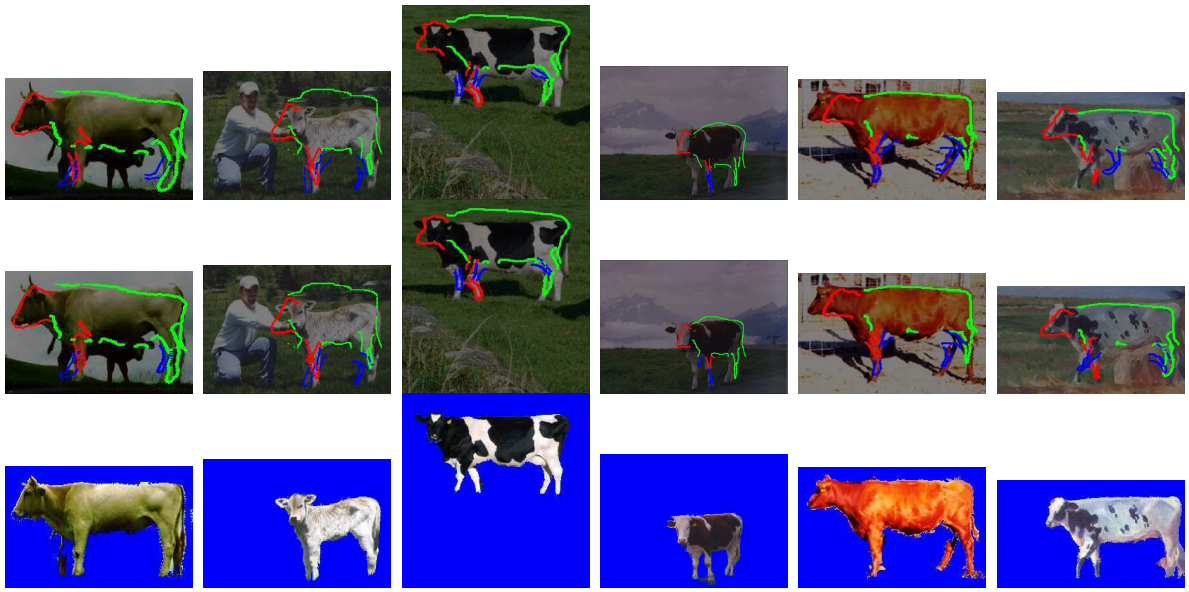


Fig. 18. *Image segmentation results 2. The first two images in each column show some of the samples of the* LPS *model. The segmentation obtained using the Object Category Specific* CDRF *is shown in the last row. Most of the errors were caused by the tail (which was not a part of the* LPS *model) and parts of the background which similar in color to the object.*

The 8 cow images and 5 horse images were manually segmented to obtain ground truth for comparison. For the cow images, out of the 125,362 foreground pixels and 472,670 background pixels present in the ground truth, 120,127 (95.82%) and 466,611 (98.72%) were present in the segmentations obtained. Similarly, for the horse images, out of the 79,860 foreground pixels and

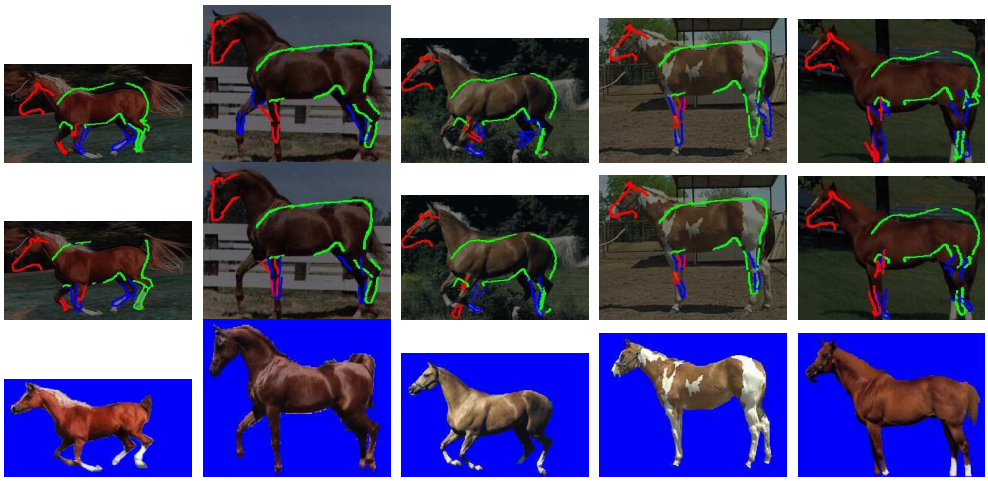


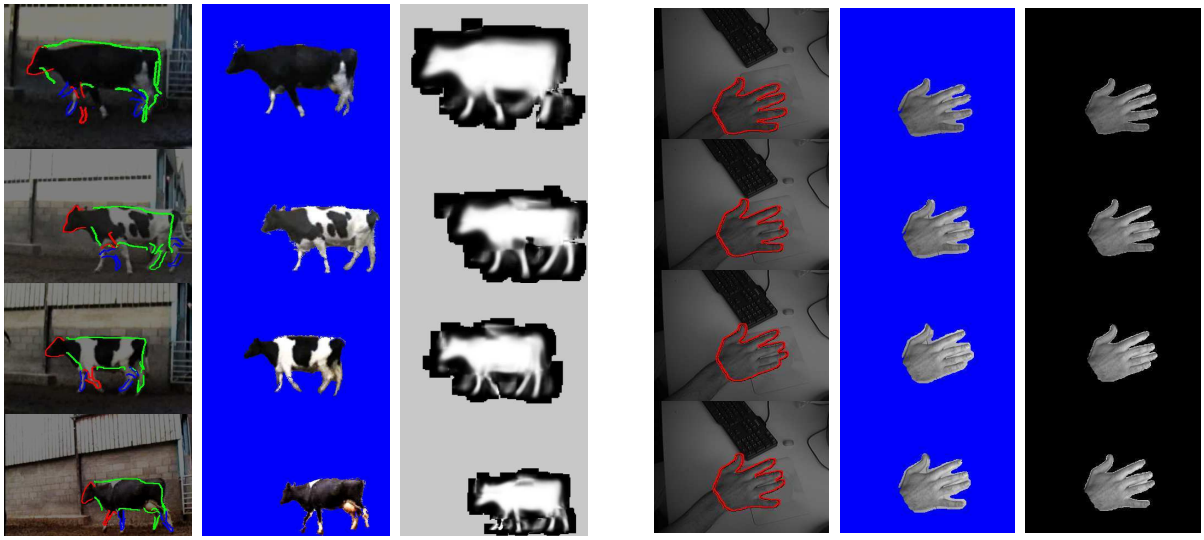
Fig. 19. *Image segmentation results 3. The* LPS *model for the horse learnt using manually segmented images was used to obtain the labeling of previously unseen images. Most of the errors were caused by unmodelled parts i.e. the mane and the tail.*

151,908 background pixels present in the ground truth, 71,397 (89.39%) and 151,185 (99.52%) were obtained in the segmentations computed by our approach. In the case of horses, most errors are due to unmodelled mane and tail parts[4]. Results indicate that, by considering both modelled and unmodelled deformations, excellent segmentations were obtained by OBJCUT.

Figure 20(a) shows a comparison of the segmentation results obtained when using OBJCUT with a state of the art method for object category specific segmentation proposed by Leibe and Schiele [26]. Note that a similar approach was described in [4]. The OBJCUT algorithm provides better segmentations using a significantly smaller number of exemplars. It achieves this by exploiting the ability of st-MINCUT for providing excellent segmentations using a good initialization obtained by the object category model. Figure 20(b) shows a comparison of our results for a non-articulated object (hand) with the global optimum found using the method described in [35]. We use the same shape template for the hand as in [35]. The possible poses of the hand are detected in an image by matching a set of exemplars obtained by applying slight deformation on the shape template. These poses are then provided as input to the OBJCUT algorithm to obtain the pixel-wise segmentation. Note that we are able to accurately detect and segment the object. Furthermore, unlike [35], our method is also applicable to parts based models and can therefore easily handle articulated object categories.

Figure 21 shows the effects of using only the shape potential $\theta^S_{a;f(a)}$ and only the appearance

[4]The images and their ground truth, together with the segmentations obtained by OBJCUT, are available at http://www.robots.ox.ac.uk/~vgg/research/objcut/index.html.



(a) (b)

Fig. 20. *Comparison with other methods. (a) The first image of each row shows a sample obtained by matching the* LPS *model to the image. The second image is the segmentation obtained using the* OBJCUT *algorithm. The third image shows the result obtained using [26] which detects extra half limbs (input and output images were provided by the authors). (b) The first image in each row shows the input and the outline of one of the samples. The second image shows the results of our approach. The third image is the global optimum obtained by the method described in [35] (provided by the authors).*

potential $\theta^A_{a;f(a)}$ by discarding the other completely. Results indicate that good segmentations depend on combining both the potentials, as is the case with the OBJCUT algorithm.

Fig. 21. *Effects of shape and appearance potentials. The first column of each row shows an image containing a cow. The segmentation results obtained by using only the* RGB *histograms for the foreground and the background provided by the* LPS *model are shown in the second column. The results obtained by using only the shape potential provided by the* LPS *model is shown in the third column. The fourth column shows the segmentations we get using the* OBJCUT *algorithm. Results indicate that good segmentation is obtained only when both shape and appearance potentials are used.*

## VII. DISCUSSION

The approach presented in this work overcomes the problems of previous methods. Specifically, it efficiently provides accurate segmentation which resembles the object. The accuracy of the segmentation can be attributed to the novel probabilistic model, i.e. Object Category Specific CDRF. Object Category Specific CDRF combines the grid CDRF models previously used with an

object category model. While the grid CDRF provides bottom-up information, the object category model incorporates top-down information about the shape of the object.

The efficiency of the method is due to two reasons: (i) we showed how the samples of the object category models of our choice can be quickly obtained using a tree cascade of classifiers and efficient sum-product BP. However, we would again like to emphasize that the techniques developed in this paper are general, i.e. they are equally applicable to other object category models such as those described in [26], [29]; and (ii) our observation that, within the generalized EM framework, a sampling-based approximation of the complete data log-likelihood can be optimized using a single st-MINCUT.

However, our method may not scale well when the number of exemplars is huge, e.g. when we want to handle multiple object categories simultaneously. In such cases, the advantage of using feature sharing methods such as [41] within our approach needs to be explored.

Currently, the shape potential provided by the object category model $\mathbf{\Omega}$ is incorporated as a unary term in the Object Category Specific CDRF. An interesting direction for future work would be to use higher order clique potentials provided by $\mathbf{\Omega}$. Some promising work in this area [20] already seems to indicate that vast improvements are possible by using more complex potentials.

## APPENDIX - EFFICIENT BELIEF PROPAGATION

The message that part $p_i$ passes to its neighbor $p_j$ at iteration $t$ is a vector of length equal to the number of discrete part labels $n_L$ of $p_j$ and is given by:

$$m_{ij}^t(\mathbf{t}_j) \leftarrow \sum_{\mathbf{t}_i} p(\mathbf{Z}_i) \exp(-\alpha(\mathbf{t}_i, \mathbf{t}_j)) \prod_{s \neq i, s \neq j} m_{si}^{t-1}(\mathbf{t}_i). \tag{30}$$

The beliefs (posteriors) after $T$ iterations are calculated as:

$$b_i^T(\mathbf{t}_i) = p(\mathbf{Z}_i) \prod_{s \neq i} m_{si}^T(\mathbf{t}_i), \quad b_{ij}^T(\mathbf{t}_i, \mathbf{t}_j) = p(\mathbf{Z}_i) p(\mathbf{Z}_j) \prod_{s \neq i, s \neq j} m_{si}^T(\mathbf{t}_i) m_{sj}^T(\mathbf{t}_j). \tag{31}$$

All messages are initialized to 1. The algorithm is said to have converged when the rate of change of all beliefs falls below a certain threshold. The messages can be computed efficiently as follows. Let $\mathcal{C}_i(\mathbf{t}_j)$ be the set of part labels of $p_i$ which form a valid pairwise configuration with $\mathbf{t}_j$. We define $T(i,j) = \sum_{\mathbf{t}_i} p(\mathbf{Z}_i) \prod_{s \neq i, s \neq j} m_{si}^{t-1}(\mathbf{t}_i)$, which is independent of the part label $\mathbf{t}_j$ of $p_j$ and needs to be calculated only once before $p_i$ passes a message to $p_j$. We also define $S(i, \mathbf{t}_j) = \sum_{\mathbf{t}_i \in \mathcal{C}_i(\mathbf{t}_j)} p(\mathbf{Z}_i) \prod_{s \neq i, s \neq j} m_{si}^{t-1}(\mathbf{t}_i)$, which is computationally inexpensive to calculate since $\mathcal{C}_i(\mathbf{t}_j)$ consists of very few part labels. The message $m_{ij}^t(\mathbf{t}_j)$ is calculated as

$$m_{ij}^t(\mathbf{t}_j) \leftarrow \exp(-d_1) S(i, \mathbf{t}_j) + \exp(-d_2)(T(i,j) - S(i, \mathbf{t}_j)). \tag{32}$$

Clearly, the above message passing equation is equivalent to that shown in equation (30).

*Acknowledgements.* M. Pawan Kumar was funded by the EU CLASS project and by EPSRC grant EP/C006631/1(P). Philip Torr is in receipt of a Royal Society Wolfson Research Merit Award, and would like to acknowledge support from the Royal Society and Wolfson foundation. Andrew Zisserman thanks Microsoft Research and the Royal Academy of Engineering for support.

## REFERENCES

[1] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *ECCV*, pages III:54–65, 2004.

[2] A. Blake, C. Rother, M. Brown, P. Perez, and P.H.S. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, pages I: 428–441, 2004.

[3] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[4] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, page II: 109 ff., 2002.

[5] Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *ICCV*, pages I: 105–112, 2001.

[6] J.M. Coughlan, A.L. Yuille, C. English, and D. Snow. Efficient optimization of a deformable template using dynamic programming. In *CVPR*, pages 747–752, 1998.

[7] D. Cremers, N. Sochen, and Schnoerr C. Mutliphase dynamic labelling for variational recognition-driven image segmentation. *IJCV*, 66:67–81, 2006.

[8] P.F. Felzenszwalb. Representation and detection of deformable shapes. In *CVPR*, pages I: 102–108, 2003.

[9] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, pages II: 66–73, 2000.

[10] P.F. Felzenszwalb and D.P. Huttenlocher. Fast algorithms for large state space HMMs with applications to web usage analysis. In *NIPS*, 2003.

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

[12] M.A. Fischler and R.A. Elschlager. The representation and matching of pictorial structures. *TC*, 22:67–92, January 1973.

[13] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, pages I: 755–762, 2005.

[14] D.M. Gavrilla. Pedestrian detection from a moving vehicle. In *ECCV*, pages II: 37–49, 2000.

[15] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 1995.

[16] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *PAMI*, 6(6):721–741, 1984.

[17] J. Goldstein, J. Platt, and C. Burges. Redundant bit vectors for quickly searching high-dimensional regions. In *Deterministic and Statistical Methods in Machine Learning*, pages 137–158, 2005.

[18] P. Hammer. Some network flow problems solved with pseudo-boolean programming. *Operations Research*, 13:388–399, 1965.

[19] R. Huang, V. Pavlovic, and D.N. Metaxas. A graphical model framework for coupling MRFs and deformable models. In *CVPR*, pages II: 739–746, 2004.

[20] P. Kohli, M. P. Kumar, and P. H. S. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[21] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE PAMI*, 26(2):147–159, 2004.

[22] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *BMVC*, pages II: 789–798, 2004.

[23] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, pages I: 18–25, 2005.

[24] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76(3):301–319, 2008.

[25] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In *ICML*, 2001.

[26] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, pages II: 264–271, 2003.

[27] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, pages IV: 581–594, 2006.

[28] P. Meer and B. Georgescu. Edge detection with embedded confidence. *PAMI*, 23:1351–1365, December 2001.

[29] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR*, pages I: 3–10, 2006.

[30] M. Prasad, A. Zisserman, A. Fitzgibbon, M. P. Kumar and P. H. S. Torr. Learning class-specific edges for object detection and segmentation. In *ICVGIP*, 2006.

[31] D. Ramanan. Using segmentation to verify object hypothesis. In *CVPR*, 2007.

[32] D. Ramanan and D.A. Forsyth. Using temporal coherence to build models of animals. In *ICCV*, pages 338–345, 2003.

[33] J. Rihan, P. Kohli, and P. H. S. Torr. OBJCUT for face detection. In *ICVGIP*, 2006.

[34] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, pages 309–314, 2004.

[35] T. Schoenemann and D. Cremers. Globally optimal image segmentation with an elastic shape prior. In *ICCV*, 2007.

[36] T. Schoenemann and D. Cremers. Globally optimal shape-based tracking in real-time. In *CVPR*, 2008.

[37] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, pages I: 503–510, 2005.

[38] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, pages I: 1–15, 2006.

[39] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Hand pose estimation using heirarchical detection. In *Intl. Workshop on Human-Computer Interaction*, pages 105–116, 2004.

[40] A. Thayananthan, B. Stenger, P.H.S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, pages I: 127–133, 2003.

[41] A. Torralba, K.P. Murphy, and W.T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 29(5):854–869, 2007.

[42] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *ICCV*, pages II: 50–57, 2001.

[43] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *CVPR*, pages II:691–698, 2003.

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

[45] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, pages I: 37–44, 2006.

[46] J. Yedidia, W. Freeman, and Y. Weiss. Bethe free energy, kikuchi approximations, and belief propagation algorithms. Technical Report TR2001-16, MERL, 2001.