ClassCut for Unsupervised Class Segmentation

Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari

Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland
{bogdan,deselaers,ferrari}@vision.ee.ethz.ch

Abstract. We propose a novel method for unsupervised class segmentation on a set of images. It alternates between segmenting object instances and learning a class model. The method is based on a segmentation energy defined over all images at the same time, which can be optimized efficiently by techniques used before in interactive segmentation. Over iterations, our method progressively learns a class model by integrating observations over all images. In addition to appearance, this model captures the location and shape of the class with respect to an automatically determined coordinate frame common across images. This frame allows us to build stronger shape and location models, similar to those used in object class detection. Our method is inspired by interactive segmentation methods [1], but it is fully automatic and learns models characteristic for the object class rather than specific to one particular object/image. We experimentally demonstrate on the Caltech4, Caltech101, and Weizmann horses datasets that our method (a) transfers class knowledge across images and this improves results compared to segmenting every image independently; (b) outperforms Grabcut [1] for the task of unsupervised segmentation; (c) offers competitive performance compared to the state-of-the-art in unsupervised segmentation and in particular it outperforms the topic model [2].

1 Introduction

Image segmentation is a fundamental problem in computer vision. Over the past years methods that use graph-cut to minimize binary pairwise energy functions have become the de-facto standard for segmenting specific objects in individual images [1, 3, 4]. These methods employ appearance models for the foreground and background which are estimated through user interactions [1, 3, 4].

On the one hand, analog approaches have been presented for object class segmentation where the appearance models are learned from a set of training images with ground-truth segmentations [5–7]. However, obtaining ground-truth segmentations is cumbersome and error-prone.

On the other hand, approaches to unsupervised class segmentation have also been proposed [2, 8–10, 12, 13]. In unsupervised segmentation a set of images depicting different instances of an object class is given, but without information about the appearance and shape of the objects to be segmented. The aim of an algorithm is to automatically segment the object instance in each image.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part V, LNCS 6315, pp. 380–393, 2010.

Interestingly, most previous approaches to unsupervised segmentation do not use energy functions similar to those in interactive and supervised segmentation, but instead use topic models [2] or other specialized generative models [10, 12] to find recurring patterns in the images.

We propose ClassCut, a novel method for unsupervised segmentation based on a binary pairwise energy function similar to those used in interactive/supervised segmentation. As opposed to those, our energy function is defined over a set of images rather than on one image [1, 3–5]. Inspired by GrabCut [1], where the two stages of learning the foreground/background appearance models and segmenting the image are alternated, our method alternates between learning a *class model* and segmenting the objects in all images jointly. The class model is learned from all images at the same time, so as to capture knowledge about the class rather than specific to one image [1]. Therefore, it helps the next segmentation iteration, as it transfers between images knowledge about the appearance and shape of the class. Thanks to the nature of our energy function, we can segment all images jointly using existing efficient algorithms used in interactive segmentation approaches [1, 3, 14, 15].

Inspired by representations successfully used in supervised object class detection [16, 17], our approach anchors the object class in a reference coordinate frame common across images. This enables modeling the spatial structure and shape of the class, as well as designing novel priors tailored to the unsupervised segmentation task. We determine this reference frame automatically in every image with a procedure based on a salient object detector [18].

At each iteration ClassCut updates the class model, which captures the appearance, shape, and location distribution of the class within the reference frame. The final output of the method are images with segmented object instances as well as the class model.

In the experiments, we demonstrate that our method (a) transfers knowledge between images and this improves the performance over segmenting each image independently; (b) outperforms the original GrabCut [1], which is the main inspiration behind it and turns out to be a very competitive baseline for unsupervised segmentation; (c) offers competitive performance compared to the state-of-the-art in unsupervised segmentation; (d) learns meaningful, intuitive class models. Source code for ClassCut is available at http://www.vision.ee.ethz.ch/~calvin.

Related Work. We discussed in the introduction that our method employs energy minimization techniques used in interactive segmentation [1, 3, 4, 14, 15], and how it is related to supervised [5, 7, 19] as well as to unsupervised [2, 10–12] class segmentation methods.

A different task is object discovery, which aims at finding multiple object classes from a mixed set of unlabeled images [11, 29]. In our work instead, all images contain instances of one class.

The two closest work to ours are [8, 9], which have a procedure iterating between updating a model and segmenting the images. In [8] the model is given a set of class and non-class images and then it iteratively improves the foreground/background labeling of image fragments based on their class likelihoods.

Their method learns local segmentations masks for image fragments, while our method learns a more complete class model, including appearance, shape and location in a global reference frame.

Arora et al. [9] learn a template consistent over all images using variational inference. Their template model is very different from our class model, and closer to a constellation model [20]. Moreover, their method optimizes the segmentation of the images individually rather than jointly.

Finally, our approach is also related to co-segmentation [21] where the goal is to segment a specific object from two images at the same time. Here we try to go a step further and co-segment a set of images showing different object instances of an unknown class.

2 Overview of Our Method

The goal is to jointly segment objects of an unknown class from a set of images. Analog to the scheme of GrabCut [1], ClassCut alternates two stages: (1) learning/updating a class model given the current segmentations (sec. 4); (2) jointly segmenting the objects in all images given the current class model (sec. 3). It converges when the segmentation is unchanged in two consecutive iterations.

Our segmentation model for stage (2) is a binary pairwise energy function, which can be optimized efficiently by techniques used in interactive segmentation [1, 3, 22], but jointly over all images rather than on a single image [1] (sec. 3).

In stage (1), learning the class model over all images at once enables capturing knowledge characteristic for the *class* rather than specific to a particular *image* [1]. As the class model is used in the next segmentation iteration it transfers knowledge across images, typically from easier images to more difficult ones, aiding their segmentation. For example, the model might learn in the first iteration that airplanes are typically grayish and the background is often blue (fig. 1). In the next iteration, this will help in images where the airplane is difficult to segment (e.g. because of low contrast).

The class model we propose (sec. 3.2) consists of several components modeling different class characteristics: appearance, location, and shape. In addition to a color component also used in GrabCut [1], the appearance model includes a bag-of-words [23] of SURF descriptors [24], which is well suited for modeling class appearance. Moreover, we model the location (sec. 3.2) and shape (sec. 3.2) of the object class w.r.t. a reference coordinate frame common across images (sec. 5). Overall, our model focuses on knowledge at the class level rather than at the level of one object as in the works it is inspired from [1, 4].

In addition to the class model, the segmentation energy include priors tailored for segmenting classes (sec. 3.1). The priors are defined on superpixels [25], which act as grouping units for homogeneous areas. Superpixels bring two advantages: (i) they provide additional structure, i.e. the set of possible segmentations is reduced to those aligning well with image boundaries; (ii) they reduce the computational complexity of segmentation. We formulate four class segmentation priors over superpixels and multiple images (sec. 3.1).

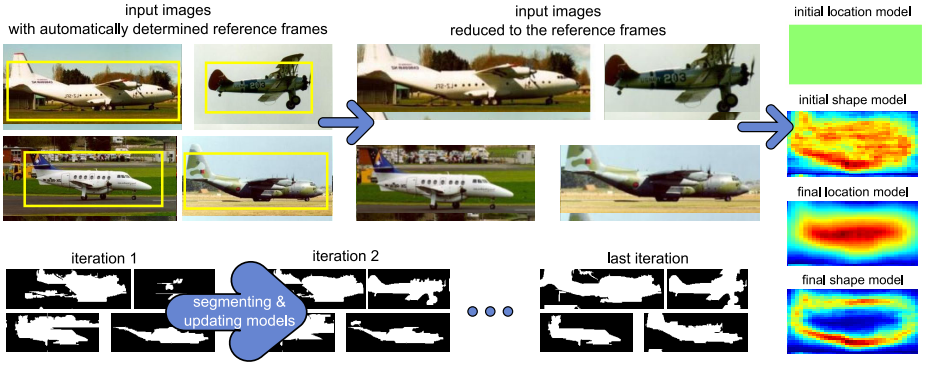

Fig. 1. Overview of our method. The top row shows the input images, the automatically determine reference frames and the initial location and shape models. The bottom row shows how the segmentations evolve over the iterations as well as the final location and shape models.

If a common reference frame on the objects is available, our method exploits it to anchor the location and shape models to it and to improve the effectiveness of some of the priors. We apply a salient object detector [18] to determine this reference frame automatically (sec. 5). In sec. 6 we show how this detector improves segmentation results compared to using the whole image as a reference frame. Fig. 1 shows an overview of the entire method.

3 Segmentation

In the set of images $\mathcal{I} = \{I_1, \ldots, I_N\}$ each image I_n (given either as a full image or as automatically determined reference frame) consists of superpixels $\{S_n^1, \ldots, S_n^{K_n}\}$. We search for the labeling $L^* = \Big((l_1^1, \ldots, l_1^{K_1}), \ldots, (l_n^1, \ldots, l_n^{K_n}), \ldots, (l_N^1, \ldots, l_N^{K_N})\Big)$ that sets $l_n^k = 1$ for all superpixels S_n^k on the foreground and $l_n^j = 0$ for all superpixels S_n^j on the background.

To determine L^*, we minimize

$$L^* = \arg\min_L \{E_\Theta(L, \mathcal{I})\} \quad \text{with} \quad E_\Theta(L, \mathcal{I}) = \Phi_\Theta(L, \mathcal{I}) + \Psi_\Theta(L, \mathcal{I}) \tag{1}$$

where Φ is the segmentation prior (sec. 3.1) and Ψ is the class model (sec. 3.2). In sec. 3.3 we describe how to minimize eq. (1). Θ are the parameters of the model.

3.1 Prior $\Phi_\Theta(L, \mathcal{I})$

The prior Φ consists of four terms

$$\Phi_\Theta(L, \mathcal{I}) = w_\Lambda \Lambda(L, \mathcal{I}) + w_\chi \chi(L, \mathcal{I}) + w_\Gamma \Gamma(L, \mathcal{I}) + w_\Delta \Delta(L, \mathcal{I}) \tag{2}$$

The scalars w are part of the model parameters Θ and weight the terms. Below we describe the terms in detail.

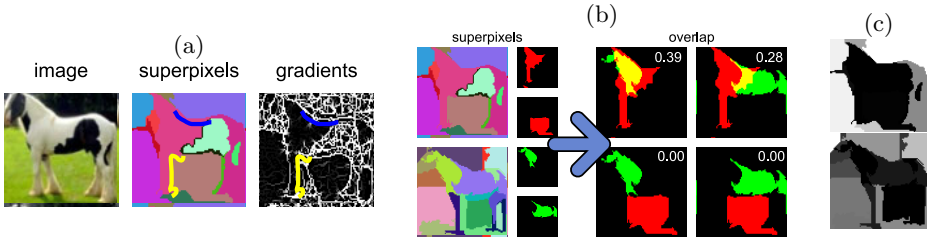

Fig. 2. Priors. (a) The smoothness prior between two superpixels is weighted inversely to the sum over the gradients along their boundary (shown in yellow and blue for two pairs of superpixels). (b) The between image smoothness prior is weighted by the overlap (yellow) of superpixels (shown for two pairs of superpixels (red/green) in two images). (c) The border penalty assigns high values to superpixels touching the reference frame boundary (dark=low values, bright=high values).

The Within Image Smoothness Λ is a smoothness prior for superpixels which generalizes the pixel-based smoothness priors typically used in interactive segmentation [1]. It penalizes neighboring superpixels having different labels.

$$\Lambda(L,\mathcal{I}) = \sum_n \sum_{j,k} \delta(l_n^j \neq l_n^k) \exp(-\text{grad}(S_n^j, S_n^k)) \tag{3}$$

where j,k are the indices of neighboring superpixels S_n^j, S_n^k within image I_n. $\delta(l_n^j \neq l_n^k) = 1$ if the labels l_n^j, l_n^k are different and 0 otherwise. The gradient $\text{grad}(S_n^j, S_n^k)$ between S_n^j and S_n^k is computed by summing the gradient magnitudes [26] along the boundary between S_n^j, S_n^k (fig. 2a) normalized w.r.t. the length of the boundary. Thus, the penalty is smaller if the two superpixels are separated by high gradients. This term encourages segmentations aligned with the image gradients.

The Between Image Smoothness χ operates on superpixels across images. It encourages superpixels in different images but with similar location w.r.t. the reference frame to have the same label:

$$\chi(L,\mathcal{I}) = \sum_{n,m} \sum_{j,k} \delta(l_n^j \neq l_m^k) \frac{|S_n^j \cap S_m^k|}{|S_n^j \cup S_m^k|} \tag{4}$$

where n,m are two images and j,k superpixels, one in I_n, the other in I_m. This penalty grows with the overlap of the superpixels (measured as area of intersection over area of union). Therefore only overlapping superpixels interact (fig. 2b). This term encourages similar segmentations across all images (w.r.t. the reference frame).

The Border Penalty Γ prefers superpixels at the image boundary to be labeled background. Objects rarely touch the boundary of the reference frame. Notice how the object would touch even a tight bounding-box around itself only in a few points (e.g. fig. 2a). The border penalty

$$\Gamma(L,\mathcal{I}) = \sum_n \sum_k l_n^k \frac{\text{border}(S_n^k)}{\text{perimeter}(S_n^k)} \tag{5}$$

assigns a penalty proportional to the number of pixels touching the reference frame (border(S_n^k)) to each superpixel S_n^k normalized by its perimeter (perimeter(S_n^k)). This term penalizes superpixels touching the border of the reference frame to be labeled foreground (fig. 2).

Γ is only meaningful on superpixels. If the segmentation is performed at the pixel-level, the border penalty can be compared to a low prior on the boundary pixels which may be propagated toward the image center using the smoothness prior. This shows how superpixels introduce additional structure into the model.

The Area Reward Δ encourages a large foreground region in order to find the entire recurring object and not just a small recurring object part. The term

$$\Delta(L,\mathcal{I}) = \sum_n \sum_m -l_n^m \frac{|S_n^m|}{|I_n|} \tag{6}$$

assigns to each superpixel a reward proportional to its area (normalized w.r.t. the area of the reference frame).

The combined effects of Γ and Δ are similar to the (more complex) bounding-box prior [4]: the foreground region should be as large as possible while not crossing the boundary of the reference frame (here, touching it).

3.2 Class Model $\Psi_\Theta(L,\mathcal{I})$

The class model $\Psi_\Theta(L,\mathcal{I})$ accounts for the appearance, shape, and location of the objects:

$$\Psi_\Theta(L,\mathcal{I}) = w_\Omega \Omega_\Theta(L,\mathcal{I}) + w_\Pi \Pi_\Theta(L,\mathcal{I}) + \sum_f w_{\Upsilon^f} \Upsilon_\Theta^f(L,\mathcal{I}) \tag{7}$$

The scalars w are part of the model parameters Θ and weight the terms. Below we describe these models in detail. In sec. 4 we explain how they are initialized and updated over the iterations.

The Location Model Ω accounts for the locations of objects w.r.t. the reference frames. We model the probability for a pixel s at its position to be foreground $p^\Omega(l|s)$ as the empirical probability in the reference frame. p^Ω is quantized to 32×32 locations within the reference frame.

To compute the energy contribution for a superpixel S_n^k labeled foreground, we average over all positions in S_n^k and incorporate this into eq. (7) as

$$\Omega_\Theta(L,\mathcal{I}) = \sum_n \sum_k \frac{1}{|S_n^k|} \sum_{s \in S_n^k} -\log p^\Omega(l_n^k|s) \tag{8}$$

Fig. 3a shows a final location model obtained after convergence. The location model encourages similar segmentations w.r.t. the reference frame in all images.

The Shape Model Π accounts for the global shape of the objects within the reference frames. We model the global shape of the objects as the probability

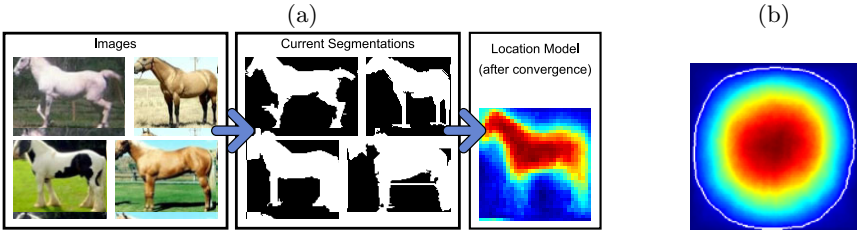

Fig. 3. (a) **Training the location model Ω.** In each iteration, we segment all images and reestimate a location model specific to the current class using the current segmentations. (b) **Generic object location prior.** The initial segmentation used to initialize appearance models is drawn in white.

$p^{\Pi}(\text{boundary}|s,\beta)$ that an object boundary with orientation β is at position s. This is modeled as the empirical probability of oriented object boundaries quantized into 5 orientations and 32×32 spatial bins.

For a pair of neighboring superpixels S_n^j, S_n^k in image I_n this probability is accumulated along their boundary $S_n^j \wr S_n^k$ to obtain the probability that one of them is foreground and the other background as:

$$p^{\Pi}(l_n^j \neq l_n^k | S_n^j, S_n^k) = \frac{1}{|S_n^j \wr S_n^k|} \sum_{s \in S_n^j \wr S_n^k} p^{\Pi}(\text{boundary}|s,\beta_s) \tag{9}$$

where β_s is orientation of pixel s. This model is then incorporated in eq. (7) as:

$$\Pi_{\Theta}(L,\mathcal{I}) = \sum_n \sum_{j,k} \delta(l_n^j \neq l_n^k) \left(\mu - p^{\Pi}(l_n^j \neq l_n^k | S_n^j, S_n^k)\right) \tag{10}$$

where $\mu = \frac{1}{5 \cdot 32 \cdot 32} \sum_{s,\beta} p^{\Pi}(\text{boundary}|s,\beta)$ is the mean probability of a boundary over all locations and orientations.

Fig. 4 shows an initial shape model and a shape model after convergence. The shape model encourages segmentations with similar shapes w.r.t. the reference frame in all images.

The Appearance Models Υ^f capture the visual appearance of the foreground and background regions according to different visual descriptors f. As visual descriptors f we use color distributions (COL) and bag-of-words [23] of SURF descriptors [24] (BOW).

For a pixel s, the probability to be foreground (or background) $p^f(l|s)$ is modelled using Gaussian mixtures for $p^{\text{COL}}(l|s)$, closely following [1], and using empirical probabilities for $p^{\text{BOW}}(l|s)$. It is incorporated into eq. (11) by averaging over all pixels within a superpixel.

Note that our appearance model extends the model of GrabCut [1] by the bag of SURF descriptor which is known to perform well for object classes.

$$\Upsilon_{\Theta}^f(L,\mathcal{I}) = \sum_n \sum_k -\frac{1}{|S_n^k|} \sum_{s \in S_n^k} \log p^f(l_n^k|s) \tag{11}$$

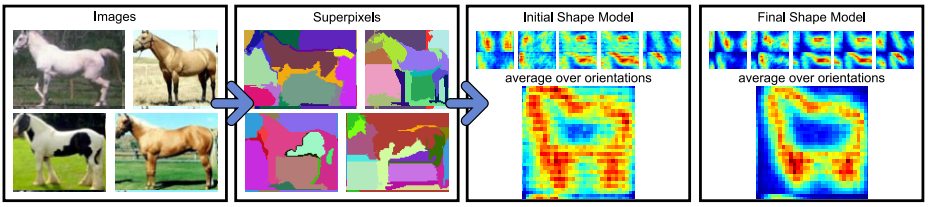

Fig. 4. The shape model. We initialize our shape model Π using only boundaries between superpixels. The shape model after convergence is shown on the right.

The appearance models capture the appearance of foreground and background region. The color model closely resembles those used in interactive segmentation and together with the bag-of-SURF model captures class appearance.

3.3 Energy Minimization

As the energy (eq. (1)) is defined over binary variables and comprises only unary $(\Gamma, \Delta, \Omega, \Upsilon_f)$ and pairwise (χ, Λ, Π) terms, we minimize it using QPBO [22]. Since QPBO labels only those superpixels for which it is guaranteed to have the global optimum, some superpixels might be left unlabeled. To label these superpixels we use TRW-S [15]. TRW-S not only labels them but also computes a lower bound on the energy which may be used to assess how far from the global optimum the solution is.

Note that all pairwise terms except for the shape model are submodular. We observed that on average only about 2% of the pairwise terms in the final model (i.e. incorporating all cues) are non-submodular.

In our experiments, we observed that QPBO labels on average 91% of the superpixels according to the global optimum.

Furthermore, we observed that the minimization problem is hardest in the first few iterations and easier in the later iterations: over the iterations QPBO labels more superpixels and the difference between the lower bound and the actual energy of the solutions is also decreased.

4 Initializing and Updating the Class Model

We describe how to initialize the model and how to update the parameters of the class models at each iteration.

4.1 Location Model

The location model Ω is initialized uniformly. At each iteration, we update the parameters of the location model using the current segmentation of all images of the current class according to the maximum likelihood criterion (fig. 3a): for each cell in the 32×32 grid we reestimate the empirical probability of foreground using the current segmentations.

4.2 Shape Model

The shape model Π is initialized by accumulating the boundaries of all superpixels in the reference frame over all images. As the boundaries of superpixels follow likely object boundaries, they will reoccur consistently along the true object boundaries across multiple images. The initial shape model (fig. 4) already contains a rough outline of the unknown object class.

At each iteration, we update the parameters of the shape model using the current segmentation of all images according to the maximum likelihood criterion: for each of the 5 orientations in the 32×32 grid, we reestimate the empirical probability for a label-change at this position and with this orientation.

While the shape model only knows about the boundaries of an object but not on which side is foreground or background, jointly with the location model (and with the between-image smoothness) it will encourage similar shapes in similar spatial arrangements to be segmented in all the images.

4.3 Appearance Model

The parameters of the appearance models Υ_f are initialized using the color/SURF observations from all images using an initial segmentation. This initial segmentation is obtained from a generic prior of object location trained on an external set of images with objects of other classes and their ground-truth segmentations (fig. 3b). From this object location prior, we select the top 75% pixels as foreground; the remaining 25% as background. We observe that this location prior is essentially a Gaussian in the middle of the reference frame.

In each iteration the Υ_f are updated according to the current segmentations like the location and shape models.

If we are using automatically determined reference frames, the observations for the background are collected from both pixels outside the reference frame and pixels inside the reference frame but labelled as background.

5 Finding the Reference Frame

To find the reference frame, we use the objectness measure of [18] which quantifies how likely it is for an image window to contain an object of *any* class. Objectness is trained to distinguish windows containing an object with a well-defined boundary and center, such as cows and telephones, from amorphous

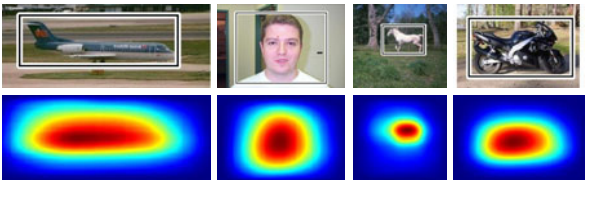

Fig. 5. Finding the reference frame. Images with automatically determined reference frames (top) and the objectness maps (bottom).

Fig. 6. Results on the Weizmann horses. From left to right: initial shape model, shape model after convergence, location model after convergence, three example images with their segmentations. The ground-truth segmentation is shown in red.

background windows, such as grass and road. Objectness combines several image cues measuring distinctive characteristics of objects, such as appearing different from their surroundings, having a closed boundary, and sometimes being unique within the image.

We sample 1000 windows likely to contain an object from this measure, project the object location prior (sec. 4.3) into these windows and accumulate into an objectness map $\mathcal{M}$ (fig. 5, (bottom)). $\mathcal{M}$ will have peaks on the objects in the image. We apply a fixed threshold to $\mathcal{M}$ and then determine a tight bounding-box around the selected pixels, which we use as the reference frame in our method (fig. 5 (top)).

In the experiments we demonstrate that this method improves the results of unsupervised segmentation compared to using the full images (sec. 6).

6 Experiments

We evaluate the segmentation accuracy of our method as the percentage of pixels classified correctly as either foreground (1) or background (0).

6.1 Datasets

We evaluate our unsupervised segmentation method on three datasets of varying difficulty and compare the results to a single-image GrabCut and to other state-of-the-art methods. In no experiment training images with segmentations of the unknown class are used.

Setting Parameters. The general parameters to be determined for our model are the weights w and the generic object location prior. These are determined on external data, i.e. images showing objects of different classes than the one under consideration for unsupervised segmentation (see below for the exact setups). We find weights w by maximizing segmentation performance on this external data. The weights are optimized using a grid-search on the weight space with the option to switch off individual terms.

Weizmann Horses [8]. We use the experimental setup of [2]: given 327 images with a horse, segment the horse in each image without using any training images with already segmented horses. Note that other approaches using the Weizmann horses typically use ground-truth segmentations in some of the images for training, e.g. [7]. The weights and the generic object location prior for these experiments are determined from the Caltech4 dataset (as discussed above).

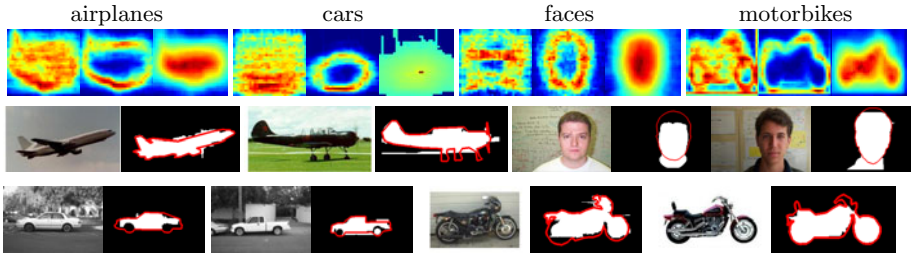

Fig. 7. Results on Caltech4. Top row: the initial shape model as well as the shape model and the location model after convergence. Below: for each class, two examples and their segmentations. The ground-truth segmentation is shown in red.

Caltech4 [27]. We use the experimental setup of [9]: for the classes airplanes, car (sideviews), faces, and motorbikes, we use the test images of [27] and segment the objects using no training data[1]. Weights and generic object location prior are set from the Weizmann Horses dataset.

Caltech101 [28]. We use an experimental setup similar to [2]: for 28 classes, we randomly select 30 images each and determine the segmentations of the objects. Note that [2] additionally uses 30 training images for each class and solves a joint segmentation and classification task (not done here). Weights and generic object location prior are set by leaving-one-out (setting parameters on 27 classes, and testing on the remaining 1; do this 28 times).

Note that most papers on unsupervised segmentation [2, 8–10, 13] use variants of these datasets. However, a few object discovery methods, e.g. [11, 29], evaluate on the more difficult datasets.

6.2 Baselines and the State of the Art

We compare our method to GrabCut [1]. To initialize GrabCut, we train a foreground color model from the central 25% of the area of the image and a background model from the rest. Using these models, GrabCut is iterated until convergence for each image individually. On Weizmann and Caltech4, we evaluate GrabCut in two different setups: (1) using the full image (Tab. 1, line (c)), (2) using the reference frame found by the method in sec. 5 instead of the full image (Tab. 1, line (d)). On Caltech101, we always use the full image as the objects are rather centered. Notice how the automatic reference frame improves the results of GrabCut from line (c) to (d) and how GrabCut is a strong competitor for previous methods [2, 9] that were designed for unsupervised segmentation.

For the datasets for which results are available, we compare our approach to Spatial Topic Models [2] (Tab. 1, line (a)) and to the approach of Arora et al. [9] (Tab. 1, line (b)).

[1] Ground-truth segmentations of the images for quantitative evaluation are taken from the Caltech101 dataset
`http://www.vision.caltech.edu/Image_Datasets/Caltech101/`

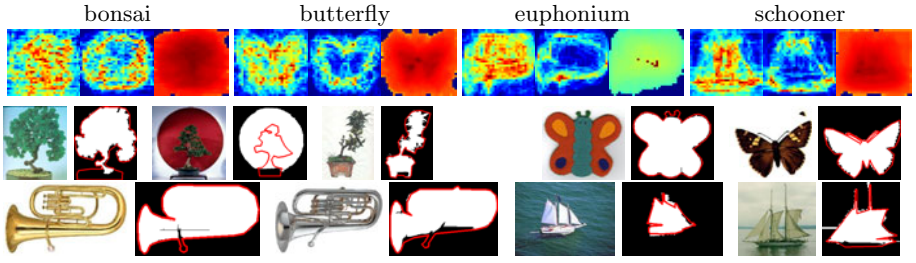

Fig. 8. Results on Caltech101. Top row: the initial shape model as well as the shape model and the location model after convergence for four example classes. Below: for each of these classes, some examples with their segmentation. The ground-truth segmentation is shown in red.

We also report the upper bound on the performance that ClassCut can obtain using superpixels [25] (Tab. 1, line (g)). This upper bound corresponds to labeling each superpixel by the majority ground-truth label of its pixels. As the upper bound is always higher than any method we consider, the superpixels are not a limiting factor for the segmentation accuracy of ClassCut.

6.3 ClassCut

We evaluate the ability of ClassCut to segment objects of an unknown class in a set of images. Qualitatively, the weights determined show that all terms in our model aid the segmentation process, as none was assigned weight 0. Furthermore, the weights are similar across all setups.

Interestingly, on the Weizmann Horses the GrabCut baseline considering only one image at a time (Tab. 1, line (c)) outperforms the (more complex) spatial topic model [2] (line (a)). When GrabCut is applied within the automatically determined reference frames (line (d)), the result is further improved. ClassCut (line (f)) improves the result a little further. Note also, how ClassCut improves its accuracy over iterations (line (e) to (f)), showing that it is properly learning about the class.

On Caltech4, we compare to [9] (line (b)). Again, the GrabCut baseline is improved when using the automatically determined reference frame rather than the entire image (line (c)/line (d)). This holds even for the classes where the automatically determined reference frames contain a considerable amount of background (cars, faces). ClassCut (line (f)) considerably improves over GrabCut (line (d)) for all classes and on average performs about as well as [9] (ClassCut: 90.6 / [9]: 90.9). Again, ClassCut improves over iterations (from (e) to (f)).

As described above, on Caltech101 we use the full images as reference frames. Using ClassCut we obtain a segmentation accuracy of 83.6%, outperforming both GrabCut (line (c)) and the spatial topic model [2] (line (a)).

Additionally, we evaluate our results using the normalized Chamfer distance to assess how well the segmentation masks align with the shape of the objects. The

Table 1. Results are reported as percentage of pixels classified correctly into either foreground or background

	Weizmann	Caltech4				Caltech101
Method	horses	airp.	cars	faces	motorb.	average
(a) Spatial Topic Model [2]	81.8	–	–	–	–	67.0
(b) Arora et al. [9]	–	**93.1**	**95.1**	**92.4**	83.1	–
(c) GrabCut (full image)	83.9	84.5	45.1	83.7	82.4	81.5
(d) GrabCut (reference frames)	85.8	88.7	81.4	89.6	82.3	–
(e) ClassCut (init)	84.7	88.4	90.7	85.3	89.2	83.0
(f) ClassCut (final)	**86.2**	**89.8**	**93.1**	**89.0**	**90.3**	**83.6**
(g) upper bound	92.4	95.5	97.2	93.3	94.7	92.9

Chamfer distance measures the average distance of every point on the segmentation outline to its closest point on the ground-truth outline, normalized by the diagonal of the ground-truth bounding-box. Since neither [2, 9] use any such measure we compare to the GrabCut baseline. For Weizmann/Caltech4/Caltech101 datasets the Chamfer distance averaged over all images is 0.09/0.06/0.13 for ClassCut and 0.20/0.27/0.23 for the corresponding GrabCut baselines. This shows that the segmentations obtained using ClassCut are better aligned to the ground-truth segmentation than those from GrabCut.

7 Conclusion

We presented a novel approach to unsupervised class segmentation. Our approach alternates between jointly segmenting the objects in all images and updating a class model, which allows to benefit from the insights gained in interactive segmentation and object class detection. Our model comprises inter-image priors and a comprehensive class model accounting for object appearance, shape, and location w.r.t. an automatically determined reference frame. We demonstrate that the reference frame allows to learn a novel type of shape model and aids the segmentation process.

References

1. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. SIGGRAPH 23, 309–314 (2004)
2. Cao, L., Li, F.F.: Spatially coherent latent topic model for concurrent segmentation and classification of objects and scene. In: ICCV (2007)
3. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV (2001)
4. Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: ICCV (2009)
5. Schroff, F., Criminisi, A., Zisserman, A.: Object class segmentation using random forests. In: Proceedings of the British Machine Vision Conference (2008)

6. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling appearance, shape and context. IJCV 81, 2–23 (2009)
7. Kumar, M.P., Torr, P.H.S., Zisserman, A.: OBJ CUT. In: CVPR (2005)
8. Borenstein, E., Ullman, S.: Learning to segment. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 315–328. Springer, Heidelberg (2004)
9. Arora, H., Loeff, N., Forsyth, D., Ahuja, N.: Unsupervised segmentation of objects using efficient learning. In: CVPR (2007)
10. Winn, J., Jojic, N.: LOCUS: learning object classes with unsupervised segmentation. In: ICCV (2005)
11. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
12. Todorovic, S., Ahuja, N.: Extracting subimages of an unknown category from a set of images. In: CVPR (2006)
13. Galleguillos, C., Babenko, B., Rabinovich, A., Belongie, S.: Weakly supervised object localization with stable segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 193–207. Springer, Heidelberg (2008)
14. Kolmogorov, V., Rother, C.: Minimizing nonsubmodular functions with graph cuts – a review. PAMI 29, 1274–1279 (2007)
15. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. PAMI 28, 1568–1583 (2006)
16. Dalal, N., Triggs, B.: Histogram of Oriented Gradients for Human Detection. In: CVPR (2005)
17. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI (2009) (in press)
18. Alexe, B., Deselaers, T., Ferrari, V.: What is an object?. In: CVPR (2010)
19. Fulkerson, B., Vedaldi, A., Soatto, S.: Localizing objects with smart dictionaries. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 179–192. Springer, Heidelberg (2008)
20. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR, vol. 2, pp. 264–271 (2003)
21. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In: CVPR (2006)
22. Rother, C., Kolmogorov, V., Lempitsky, V., Szummer, M.: Optimizing binary MRFs via extended roof duality. In: CVPR (2007)
23. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: ECCV Workshop on Stat. Learn. in: Comp. Vis. (2004)
24. Bay, H., Ess, A., Tuytelaars, T., van Gool, L.: SURF: Speeded up robust features. CVIU 110, 346–359 (2008)
25. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59, 167–181 (2004)
26. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color and texture cues. PAMI 26, 530–549 (2003)
27. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
28. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In: IEEE CVPR Workshop of Generative Model Based Vision (2004)
29. Lee, Y.J., Grauman, K.: Collect-cut: Segmentation with top-down cues discovered in multi-object images. In: CVPR (2010)