PatchCut: Data-Driven Object Segmentation via Local Shape Transfer

Jimei Yang[1], Brian Price[2], Scott Cohen[2], Zhe Lin[2], and Ming-Hsuan Yang[1]

[1]UC Merced [2]Adobe Research

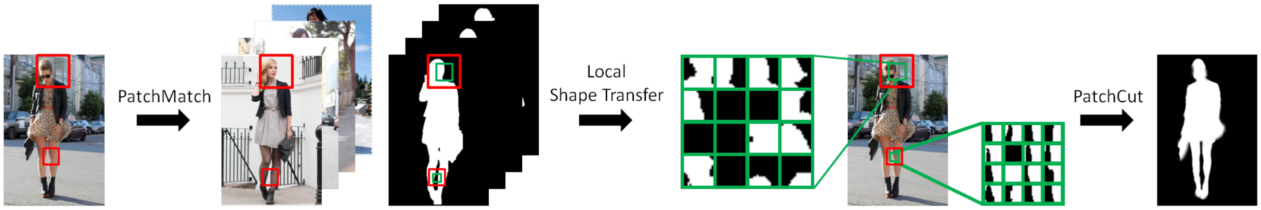

Figure 1: Overview of proposed object segmentation algorithm using examples. Given a test image and a set of segmentation examples, our algorithm first performs multiscale image matching in patches by PatchMatch. The local shape masks within the matched patches are then transferred to represent the patch-wise segmentation candidates for the test image. Finally, local mask candidates are selected based on MRF energy function to produce the segmentation in a coarse-to-fine manner.

Abstract

Object segmentation is highly desirable for image understanding and editing. Current interactive tools require a great deal of user effort while automatic methods are usually limited to images of special object categories or with high color contrast. In this paper, we propose a data-driven algorithm that uses examples to break through these limits. As similar objects tend to share similar local shapes, we match query image patches with example images in multiscale to enable local shape transfer. The transferred local shape masks constitute a patch-level segmentation solution space and we thus develop a novel cascade algorithm, PatchCut, for coarse-to-fine object segmentation. In each stage of the cascade, local shape mask candidates are selected to refine the estimated segmentation of the previous stage iteratively with color models. Experimental results on various datasets (Weizmann Horse, Fashionista, Object Discovery and PASCAL) demonstrate the effectiveness and robustness of our algorithm.

1. Introduction

Object segmentation, separating a foreground object from its background with a clear boundary, has long been an important challenge for computer vision. It not only provides mid-level representations for high-level recognition tasks [7] but also has immediate applications to image editing [1]. Object segmentation is typically formulated as a binary labeling problem on Markov Random Fields (MRFs) with foreground/background appearance models [6].

Recent methods [29, 20] show that object segmentation can be solved efficiently with a carefully prepared bounding box around the target and further refined by user inputs. In these interactive algorithms [6, 29, 20], color is commonly used to separate foreground from background. Although more complex image cues such as textures are shown to be useful to improve segmentation performance [34], a critical source of information, object shape, is clearly missing in these algorithms. Similar situations exist in salient object segmentation [28, 25, 8] in that most of algorithms work well when the images have high foreground-background color contrast, but work poorly in cluttered images. A notable exception is Li *et al.*'s latest work [22] that achieves impressive object segmentation results on the PASCAL images [10] by integrating shape sensitive object proposals [7] with a classic saliency map [11]. On the other hand, in model-based algorithms [5, 19, 18, 4, 36], shape is always the major driving force for segmentation. Category-specific shape models are usually designed [18] based on prior knowledge or learned offline from training data [21, 36]. This category-specific nature limits the generalizability of model based algorithms to handle unseen objects.

In this work, we propose a data-driven object segmentation algorithm that addresses the problems mentioned above by using a database of existing segmentation examples. Our algorithm requires neither offline training of category-specific shape models nor prior knowledge of object shapes. Instead we transfer shape masks from similar segmentation examples to the test images by image retrieval and matching. Compared to user- and saliency-driven algorithms, the transferred shape cues help resolve segmentation ambiguities from appearance models.

Existing data-driven object segmentation algorithms [17, 2, 13, 32] mostly focus on transferring entire shape masks by either window based or local features based image matching. In this paper, however, we investigate a patch-level local shape transfer scheme that finds candidate local shape masks for each patch of a test image in multiple scales through dense correspondences between query and example images built by the PatchMatch algorithm [3]. Those candidate local shape masks indeed constitute an online structured label space where object segmentation solutions can be found. We thereby develop a novel cascade algorithm for coarse-to-fine object segmentation. At each stage, we define a color based MRF energy function with the coarse shape mask estimated in the previous stage, and select the local shape mask for each patch independently with the minimum MRF energy to estimate a new shape mask with finer details. This patch-wise segmentation provides an approximate solution to global energy minimization, but a solution which is easier to solve in parallel. We carry out local shape mask selection iteratively while updating the foreground/background color models. This iterative procedure shares a similar idea with GrabCut [29], but it operates patch-wise in a structured label space. Thus we name our method *PatchCut*. We carry out experiments on various object segmentation benchmark datasets with comparisons to leading example-, learning- and saliency-based algorithms.

The contributions of this paper are summarized below:

- a novel nonparametric high-order MRF model via patch-level label transfer for object segmentation;
- an efficient iterative algorithm (PatchCut) that solves the proposed MRF energy function in patch-level without using graph cuts;
- state-of-the-art performance on various object segmentation benchmark datasets.

2. Related Work

Example-based object segmentation. Our work is closely related to [17, 2] for object segmentation using examples. In [17], the test image is matched with example images by window proposals. By adding up the matched window masks, the estimated segmentation prior contains more information about object location but less information about object shape. As a result, its segmentation performance largely depends on the final iterative GraphCut refinement step. The algorithm in [2] involves two-step image matching. The window proposals of the test image are first localized on example images and then each localized image window is aligned using SIFT flow [24] with its corresponding test window proposal to achieve deformable mask transfer. Although a better shape prior could be obtained this way, running SIFT flow for thousands of window proposals with tens of examples inevitably results in considerable computational cost. Compared to [17, 2], our algorithm performs multiscale dense matching using image patches, which can be solved by PatchMatch efficiently. Instead of adding up the transferred window masks, our algorithm estimates high quality shape priors by selecting local mask candidates in a coarse-to-fine manner so that the segmentation does not fully depend on the refinement step.

Structured label space. Our proposed idea of using local masks as solution space is inspired by recent structured forest based image labeling algorithms [15, 9]. In the training stage, the clustering structure of label patches is exploited in branching functions so that each leaf node stores one example label patch. The label patches in all the leaf nodes constitute a structured label space for edge detection [9] and semantic labeling [15]. In our algorithm, the local shape masks transferred from examples constitute another kind of structured label space for object segmentation. In spirit, both structured forests and our algorithm aim at preserving output structures (local context and shape) when making predictions. However, an important difference is that the structured label space in our algorithm is constructed online by matching with examples, which is more flexible and easier to generalize than offline training in structured forests [15, 9].

3. Our Data-Driven Algorithm

Given a test image $\mathbf{I}$, our goal is to estimate its segmentation $\hat{\mathbf{Y}}$ by using example images $\{\mathbf{I}_m, m = 1, 2, ..., M\}$ and their segmentation ground truth $\{\mathbf{Y}_m, m = 1, 2, ..., M\}$. Figure 1 presents an overview of the proposed algorithm.

3.1. Local Shape Transfer

Our algorithm performs image matching to achieve shape transfer from examples like most data-driven algorithms. However, transferring entire masks in large image windows may result in poor boundary quality [17], while alignment, although improving the boundary quality, significantly increases the computational cost [2]. In this work, we propose transferring local shape masks from multiple scales. We build three-layer image pyramids by downsampling both the test $\{\mathbf{I}^s, s = 1, 2, 3\}$ and example images $\{\mathbf{I}^s_m, \mathbf{Y}^s_m, s = 1, 2, 3\}$. If the size of image $\mathbf{I}$ is $[h, w]$, the size of downsampled image in the s^{th} layer is $[\frac{h}{2^{3-s}}, \frac{w}{2^{3-s}}]$. For all three scales, we use image patches of the same size to perform matching and mask transfer as demonstrated in Figure 2. In each scale $(s = 1, 2, 3)$, we densely sample image patches of 16×16 at every 2 pixels $\{\Delta^s_k, k = 1, 2, ..., K\}$, where $K = \frac{h \times w}{4 \times 2^{6-2s}}$ [1]. For each

[1] Padding is needed to ensure $[\frac{h}{2^{3-s}}, \frac{w}{2^{3-s}}]$ divisible by 2.

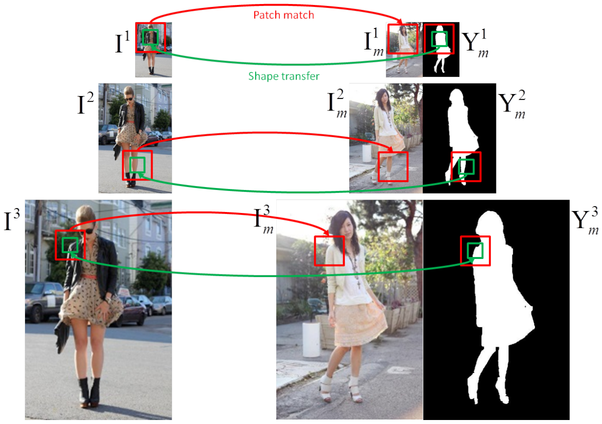

Figure 2: Local shape transfer with multiscale PatchMatch. The left part shows one query image from Fashionista in three scales and the right part shows one example image and its segmentation in image pyramid. For one query image patch in red boxes, we find its best match in the example image (red boxes) and transfer its center segmentation mask in green boxes.

patch of the test image Δ_k^s (green boxes in Figure 2), we extract a SIFT descriptor $\mathbf{x}_k^s$ from its extended 32×32 patch (red boxes in Figure 2). Therefore, the matching problem between the test $\mathbf{I}$ and the m^{th} example $\mathbf{I}_m$ can be described by $\arg\min_{k'} \|\mathbf{x}_k^s - \mathbf{x}_{k'm}^s\|_1, \forall k = 1, 2, ..., K$, where $\mathbf{x}_{k'm}^s$ is the SIFT descriptor extracted from the k'^{th} patch $\Delta_{k'}^s$ of the m^{th} example. This nearest neighbor field problem is solved efficiently by the PatchMatch algorithm [3]. As a result, the test patch Δ_k^s finds its match $\Delta_{k^*}^s$ in the m^{th} example with the cost $d_{km}^s = \|\mathbf{x}_k^s - \mathbf{x}_{k^*m}^s\|_1$.

We denote the local segmentation masks from the matched patches in m^{th} example as $\mathbf{z}_{km}^s = \mathbf{Y}_m^s(\Delta_{k^*}^s)$, which provide location and shape information for segmenting the test image. We argue that those local masks $\mathbf{z}_{km}^s$ constitute a patch-wise segmentation solution space for the test image; in other words, the segmentation mask of test image $\mathbf{Y}$ can be well approximated by $\mathbf{z}_{km}^s$. Note that different methods for image dense correspondences have been explored in [24, 14] to enable pixel-wise label transfer, but their results are either constrained by the local flow [24] or contaminated by relaxation noise [14]. Compared to [24, 14], our method achieves structured label transfer (local masks) through a more flexible matching algorithm.

To examine the quality of local shape masks $\mathbf{z}_{km}^s$, for each patch Δ_k^s we calculate the mean of its local masks $\bar{\mathbf{z}}_k^s = \frac{1}{M}\sum_m \mathbf{z}_{km}^s$, and also find the best possible $\tilde{\mathbf{z}}_k^s$ using the ground truth as reference. Note that $\tilde{\mathbf{z}}_k^s$ actually defines the upper bound for local shape transfer. Obviously, the mean shape prior mask $\bar{\mathbf{Q}}^s$ can be immediately estimated by adding up $\bar{\mathbf{z}}_k^s$ similar to [17]. Similarly, we can estimate the oracle shape prior mask $\tilde{\mathbf{Q}}^s$ from $\tilde{\mathbf{z}}_k^s$.

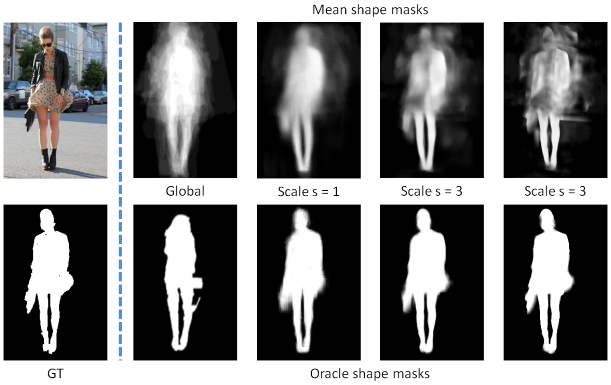

Figure 3: Shape prior masks estimated from mean masks (top row) and best masks (bottom row) at different scales. The masks are upsampled to the size of original image for better visualization.

Figure 3 demonstrates the mean and oracle shape prior masks of different scales. At the coarse scale, the object is well located but its boundary is blurry in the mean shape prior masks. Moving towards finer scales, although some parts of mean shape prior (legs) become clearer, other parts (head and shoulder) turn out to be very noisy. This is because the very local structures of image patches at the finer scales preserve well the edge patterns during matching, but local masks may have inconsistent foreground/background relationship. Meanwhile, both location and boundary qualities of oracle shape prior masks keep getting better from coarse to fine scales. This divergent result indicates that good segmentation solutions can be obtained if we find the right label patches at a fine scale, but without that knowledge, the average results are far from satisfactory. The above observations motivate the coarse-to-fine strategy where we start with a good approximation at the coarse scale which then leads us to choose the right label patches at the fine scale.

3.2. PatchCut

In this section, we introduce a novel algorithm to gradually estimate the shape prior $\hat{\mathbf{Q}}^s$ in a coarse-to-fine manner. In particular, at the s^{th} scale, given the shape prior from the previous scale $\hat{\mathbf{Q}}^{s-1}$, the finer shape prior $\hat{\mathbf{Q}}^s$ is estimated using candidate local shape masks $\mathbf{z}_{km}^s$. At the end, the binary segmentation $\hat{\mathbf{Y}}$ can be computed by thresholding the shape prior at the finest scale.

MRF with shape prior. We start with reviewing a typical object segmentation method based on shape prior, which provides the fundamentals for our algorithm. Note that we temporarily omit the scale index s to keep the description clean. Object segmentation with shape prior is commonly formulated as a MRF energy function [17, 2],

$$E(\mathbf{Y}) = \sum_{i\in\mathcal{V}} U(y_i) + \gamma \sum_{i,j\in\mathcal{E}} V(y_i, y_j) + \lambda \sum_{i\in\mathcal{V}} S(y_i, q_i). \tag{1}$$

where y_i is the binary label at pixel i, q_i is the probability at pixel i of shape prior $\mathbf{Q}$. The unary term for each pixel $U(y_i)$ is the negative log probability of the label y_i given the pixel color $\mathbf{c}_i$ and Gaussian Mixture Models (GMMs) $\mathbf{A}_1$ and $\mathbf{A}_0$ for foreground and background color,

$$U(y_i) = -\log P(y_i|\mathbf{c}_i, \mathbf{A}_1, \mathbf{A}_0). \tag{2}$$

The pairwise term $V(y_i, y_j)$ measures the cost of assigning different labels to two adjacent pixels, which is usually based on their color difference,

$$V(y_i, y_j) = \exp(-\alpha\|\mathbf{c}_i - \mathbf{c}_j\|^2)\mathbb{I}(y_i \neq y_j), \tag{3}$$

where the parameter α is estimated by the mean color difference over the image and $\mathbb{I}(\cdot)$ is an indicator function. The shape term $S(y_i, q_i)$ measures the inconsistency with shape prior $\mathbf{Q}$,

$$S(y_i, q_i) = -\log q_i^{y_i}(1 - q_i)^{1-y_i}. \tag{4}$$

This energy function can be solved by alternating two steps in a way similar to the GrabCut algorithm [29]: 1) updating GMM color models in (2) from the current segmentation $\{\mathbf{A}_1, \mathbf{A}_0\} \leftarrow \mathbf{Y}$; 2) solving the MRF energy function in (1) with updated color models by GraphCut: $\mathbf{Y} \leftarrow \{\mathbf{A}_1, \mathbf{A}_0\}$. However, this method is too sensitive to the parameter λ. On one hand, if the λ is large, the color models cannot correct the mistakes in the shape prior; on the other hand, if the λ is small, the segmentation may deviate from the good shape prior.

High order MRF with local shape transfer To use candidate local shape masks to resolve segmentation ambiguities, we can naturally extend (1) to include a patch likelihood $P_{\text{cand}}(\mathbf{Y}(\Delta_k))$ that encourages the label patch $\mathbf{Y}(\Delta_k)$ for image patch $\mathbf{I}(\Delta_k)$ to be similar to some candidate local shape mask $\mathbf{z}_{km} = \mathbf{Y}_m(\Delta_{km})$ for database image patch $\mathbf{I}_m(\Delta_{km})$:

$$E'(\mathbf{Y}) = E(\mathbf{Y}) - \sum_k \log(P_{\text{cand}}(\mathbf{Y}(\Delta_k))). \tag{5}$$

The last term is the negative Expected Patch Log Likelihood (EPLL), a formulation that Zoran and Weiss [37] use for image patches to produce state-of-the-art results on inverse problems such as deblurring. Here we define the patch likelihood on local shape masks by marginalizing out over a hidden variable m_k^* that indicates which database patch Δ_{km} is selected for transfer to the output patch $\mathbf{Y}(\Delta_k)$:

$$\begin{aligned} P_{\text{cand}}(\mathbf{Y}(\Delta_k)) &= \sum_{m=1}^{M} P(\mathbf{Y}(\Delta_k), m_k^* = m) \\ &= \sum_{m=1}^{M} P(\mathbf{Y}(\Delta_k)|m_k^* = m)P(m_k^* = m) \\ &= \sum_{m=1}^{M} \frac{\exp(-\eta\|\mathbf{Y}(\Delta_k) - \mathbf{z}_{km}\|_2^2)}{Z_1} \frac{\exp(-\tau d_{km})}{Z_2}, \end{aligned}$$

where the second term expresses the probability by image appearance that we want to transfer the m^{th} candidate label patch and the first term expresses that the output label patch should be similar to the transferred patch. Note that Z_1, Z_2 are normalization terms, and d_{km} is the match cost introduced in the previous section. We assume that η is large to encourage the output label patches $\mathbf{Y}(\Delta_k)$ to be as similar to the selected candidate patches $\mathbf{z}_{km_k^*}$ as possible. For large η and distinct $\mathbf{z}_{km}$, we have

$$P_{\text{cand}}(\mathbf{Y}(\Delta_k)) \approx \begin{cases} \exp(-\tau d_{km})/Z_2 & \text{if } \mathbf{Y}(\Delta_k) = \mathbf{z}_{km} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

and

$$E'(\mathbf{Y}) \approx E(\mathbf{Y}) + \tau \sum_k H(\mathbf{Y}(\Delta_k)), \tag{7}$$

where

$$H(\mathbf{Y}(\Delta_k)) = \begin{cases} d_{km} & \text{if } \mathbf{Y}(\Delta_k) = \mathbf{z}_{km} \\ \infty & \text{otherwise} \end{cases}. \tag{8}$$

Note that this approximate energy (7) is related to the Nonparametric Higher-order Random Field (NHRF) introduced in [26] that considers top-down local appearance matching (8) but not the bottom-up global image cues (1).

Approximate optimization on patches. The high order term $H(\mathbf{Y}(\Delta_k))$ actually enforces label patch selection among all the local shape masks $\mathbf{z}_{km}$, and thus the solutions of energy function $E'(\mathbf{Y})$ do not exist when selected label patches disagree in any overlapping areas. To address this issue, we introduce an auxiliary variable $\mathbf{z}_k$ to indicate the selected label patch $\mathbf{z}_k \in \{\mathbf{z}_{k1}, \mathbf{z}_{k2}, ..., \mathbf{z}_{kM}\}$ on the k^{th} patch Δ_k and thus rewrite the energy (7) by

$$E'(\mathbf{Y}, \{\mathbf{z}_k\}) \approx E(\mathbf{Y}) + \tau \sum_k H(\mathbf{z}_k), \text{s.t. } \mathbf{Y}(\Delta_k) = \mathbf{z}_k.$$

We notice that the energy $E(\mathbf{Y})$ can be further decomposed into a summation of local energies on $\mathbf{Y}(\Delta_k) = \mathbf{z}_k$

$$E'(\mathbf{Y}, \{\mathbf{z}_k\}) \approx \kappa \sum_k E(\mathbf{z}_k) + \tau \sum_k H(\mathbf{z}_k), \text{s.t. } \mathbf{Y}(\Delta_k) = \mathbf{z}_k,$$

where the constant κ is inversely proportional to the number of patches superimposing on a single pixel. To tolerate the inconsistency between $\mathbf{Y}(\Delta_k)$ and $\mathbf{z}_k$, we convert it into an unconstrained problem by introducing a quadratic penalty on each patch

$$E'(\mathbf{Y},\{\mathbf{z}_k\}) \approx \sum_k (\kappa E(\mathbf{z}_k)+\tau H(\mathbf{z}_k)+\frac{\beta}{2}\|\mathbf{Y}(\Delta_k)-\mathbf{z}_k\|^2). \tag{9}$$

In a similar spirit with the dual decomposition method [33], this quadratic penalty energy function (9) can be minimized by alternatively solving a series of independent slave problems on patch $\mathbf{z}_k$ and a master problem on $\mathbf{Y}$. However, for sufficiently large β, we can approximately solve (9) by a simple two-step minimization:

$$\hat{\mathbf{z}}_k = \arg\min_{\mathbf{z}_k} \kappa E(\mathbf{z}_k) + \tau H(\mathbf{z}_k), \forall k \tag{10}$$

$$\hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \sum_k \frac{1}{2}\|\mathbf{Y}(\Delta_k) - \hat{\mathbf{z}}_k\|^2. \tag{11}$$

Note that (10) can be immediately solved by evaluating the energies of all the local mask candidates in parallel. To solve (11), we need to consider the inconsistency of overlapping $\hat{\mathbf{z}}_k$. By introducing a soft segmentation mask (shape prior) $0 \leq \mathbf{Q} \leq 1$ for $\mathbf{Y}$, we first solve $\hat{\mathbf{Q}} = \arg\min_{\mathbf{Q}} \sum_k \|\mathbf{Q}(\Delta_k) - \hat{\mathbf{z}}_k\|^2$ by averaging all the selected candidates $\hat{\mathbf{z}}_k$ and then compute $\hat{\mathbf{Y}}$ by thresholding $\hat{\mathbf{Q}}$ at 0.5.

Algorithm 1 The single scale PatchCut algorithm.

```
1: while not converged do
2:     for each patch $\Delta_k$, select the candidate local shape mask $\hat{\mathbf{z}}_k$ by (10)
3:     estimate the shape prior $\hat{\mathbf{Q}}$ by averaging $\hat{\mathbf{z}}_k$ and the segmentation $\hat{\mathbf{Y}}$ by (11)
4:     update the foreground and background GMM color models $\{\mathbf{A}_1, \mathbf{A}_0\}$ by (2).
5: end while
```

Given the current segmentation $\hat{\mathbf{Y}}$, we further update the color models $\{\mathbf{A}_1, \mathbf{A}_0\}$ in (2). Iteratively, the high-order MRF energy (7) is minimized using local shape mask candidates. We summarize this procedure, dubbed as *PatchCut*, in Algorithm 1.

Cascade. Using the PatchCut algorithm at a single scale, we assemble the cascade object segmentation algorithm in Figure 4. The cascade is initialized by averaging the global shape masks transferred from examples at the coarsest scale $\hat{\mathbf{Q}}^0 = \frac{1}{M}\sum_m \mathbf{Y}^1_m$. Note that other soft segmentation methods can also be used for initialization [11, 22]. At each scale, we run Algorithm 1 with the previously estimated shape prior $\hat{\mathbf{Q}}^{s-1}$, color models $\mathbf{A}_1$, $\mathbf{A}_0$ and candidate local shape masks $\mathbf{z}^s_{km}$. The algorithm proceeds untill the scale $s = 3$ is reached. The final object segmentation is inferred by thresholding the shape prior $\hat{\mathbf{Q}}^3$, denoted as $\hat{\mathbf{Y}}_t$, or further refined by iterative graph cuts in (1), denoted as $\hat{\mathbf{Y}}_r$.

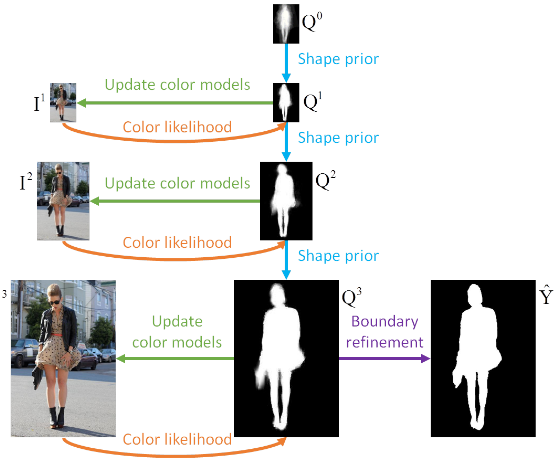

Figure 4: PatchCut cascade for coarse-to-fine object segmentation.

4. Experimental Results

We present experimental results on various object segmentation datasets (Fashionista [35], Weizmann Horse [5], Object Discovery [30], and PASCAL [10]). More results can be found in our website https://eng.ucmerced.edu/people/jyang44. The term *PatchCut_soft* denotes the shape mask $\hat{\mathbf{Q}}^3$, *PatchCut_thres* denotes the binary segmentation after thresholding $\hat{\mathbf{Y}}_t$, and *PatchCut* denotes the binary segmentation after refinement $\hat{\mathbf{Y}}_r$.

Implementation Details. To perform our algorithm, we retrieve relevant examples from a database of existing segmentations. We use Bag-of-Words (BoW) features [31] on category-specific datasets such as Fashionista, Weizmann Horse and Object Discovery, and use image features extracted from the 7th layer of convolutional networks (ConvNet [2]) [12] on the PASCAL dataset for nearest neighbor image search. The number of retrievals is set to $M = 16$. We set $\gamma = 0.5$ for the pairwise term, $\lambda = 0.5$ for the shape term in (1), $\tau = 1.0$ for the match cost term in (7), the number of Gaussian components to 5 for both foreground and background GMM color models in (2). We use the same set of parameters in all the experiments.

4.1. Fashionista

This dataset [35] consists of 700 street shots of fashion models with various poses, cluttered background and complex appearance. All the images have the same size of 600x400 pixels. We run *leave-one-out* tests on this dataset, which means that for each image, we run object segmentation by using the remaining 699 images as the database. We present some segmentation results in Figure 5. In this experiment, we compare our algorithm with the widely used

[2] The ConvNet is pre-trained on the ImageNet dataset [16].

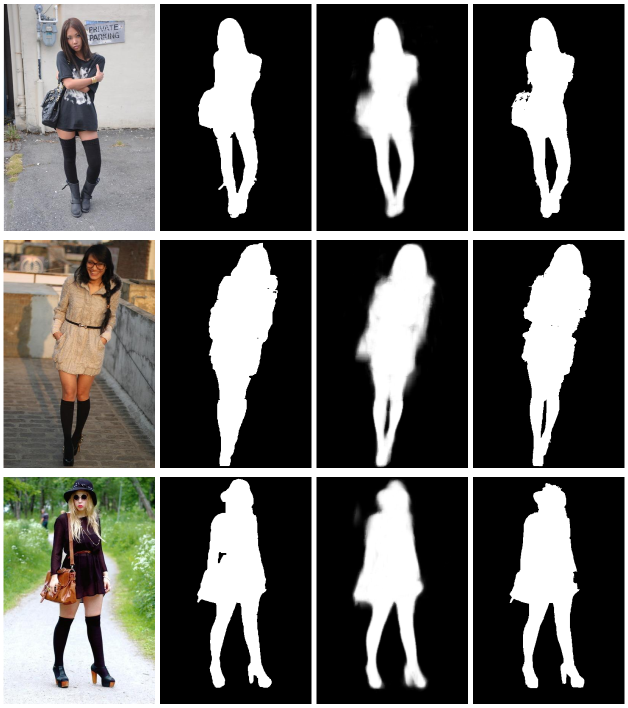
Image GT PatchCut_soft PatchCut

Figure 5: Qualitative results on Fashionista.

GrabCut baseline [29]. We use the OpenCV implementation for GrabCut by providing a bounding box with 8 pixels from the image borders at each side. We evaluate the object segmentation performance by mean Jaccard (Intersection-Over-Union) score ($|\hat{\mathbf{Y}} \cap \mathbf{Y}|/|\hat{\mathbf{Y}} \cup \mathbf{Y}|$) in Table 1. By simply thresholding the estimated shape masks, the PatchCut algorithm significantly outperforms (by more than 20%) the GrabCut baseline, and the results can be further improved by GrabCut refinement from 86.25% to 88.33%. Figure 5 shows that the refinement mainly occurs around the object contours. To take a closer look at the PatchCut performance, we calculate the segmentation success rate as the percentage of tests that achieve Jaccard scores above certain levels. Figure 6 shows that about 58% of tests achieve more than 90% Jaccard score while about 22% of tests achieve more than 95% Jaccard score.

Table 1: Segmentation performance on Fashionista.

	Jaccard (%)
GrabCut	64.23
PatchCut_thres	**86.25**
PatchCut	**88.33**
PatchCut_thres upper bound	95.72
PatchCut upper bound	95.20

Upper bound performance. We also evaluate the upper bound performance of the PatchCut algorithm. For each test image, we estimate the oracle shape prior masks $\tilde{\mathbf{Q}}$ using the ground truth segmentation $\mathbf{Y}$, and thus produce segmentation results. The mean Jaccard scores from the upper bound segmentation results are as high as 95.72% without refinement (Table 1) and the segmentation success rate at Jaccard score level 95% is near 70% (Figure 6). These upper bound results prove that the transferred local shape masks from examples constitute a valid structured label space for object segmentation.

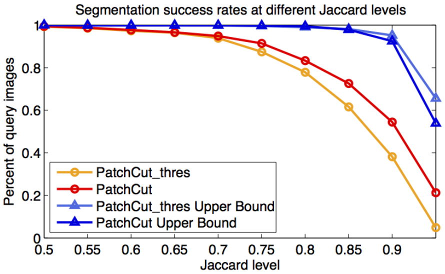

Figure 6: Segmentation success rates on Fashionista.

4.2. Weizmann Horse

Th Weizmann Horse dataset [5] is widely used for benchmarking object segmentation algorithms. This dataset consists of 328 horse images with side views. We follow a typical evaluation protocol that uses 200 images for the database and the remaining 128 for the test set. We present some qualitative results in Figure 7. We evaluate object segmentation performance in terms of mean Jaccard score and overall pixel-wise classification accuracy ($Acc = |\hat{\mathbf{Y}} == \mathbf{Y}|/|\mathbf{Y}|$). In Table 2, we compare PatchCut algorithm with state-of-the-art example based algorithm using Window Mask Transfer [17], and various leading learning based algorithms based on Kernelized Structured SVM [4], CRFs [23, 21] and Margin-Margin Boltzmann Machines (BMs) [36]. Our algorithm performs better in terms of both

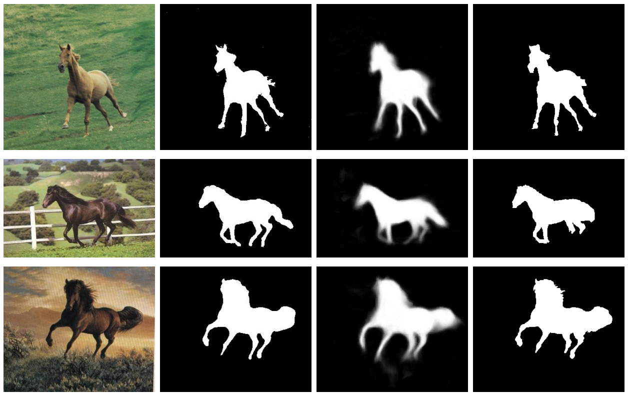
Image GT PatchCut_soft PatchCut

Figure 7: Qualitative results on Weizmanm Horse.

Table 2: Performance evaluation on Weizmann Horse.

	Jaccard (%)	Acc (%)
PatchCut_thres	**80.33**	**94.78**
PatchCut	**84.03**	**95.81**
Kernelized Structured SVM [4]	80.10	94.60
Fragment-based CRFs [21]	N/A	95.0
High-Order CRFs [23]	69.90	N/A
Max-Margin BMs [36]	75.78	90.71
Window Mask Transfer [17]	N/A	94.70

mean Jaccard score and Accuracy. Especially, our algorithm improves about 4% on mean Jaccard score. For horse images, our algorithm usually generates high quality shape prior around legs (Figure 7), but the iterated graph cuts refinement tends to cut off the legs because of the shrinking bias towards shorter boundaries.

4.3. Object Discovery

This dataset consists of three object categories: airplane, car and horse and their images are collected from Internet. It is originally designed for evaluating object co-segmentation [30] and recently used for object segmentation by Ahmed et al. [2]. This dataset is more challenging because the images generally have more complex appearance. Some images include more than one small target and some images are outliers. For each category, we use the same 100 test images as in [30, 2] and the rest as the database. Figure 8 shows some qualitative results.

We compare our algorithm with the GrabCut baseline (same implementation as Fashionista), a state-of-the-art co-segmentation algorithm [30] and the latest example based method [2] in Table 3. In the Airplane and Horse experiments the refinement step improve the results only slightly while in the Car experiment our algorithm achieves better results without using refinement. The possible reason is that the pixel-wise color models may confuse the shadows with the bottom of the car while the shape prior estimated from local masks better preserves high-level structures.

Table 3: Jaccard scores on Object Discovery.

Jaccard (%)	Airplane	Car	Horse
GrabCut	63.29	67.63	50.32
Co-segmentation [30]	55.81	64.42	51.65
Ahmed et al. [2]	64.27	71.84	55.08
PatchCut_thres	**70.44**	**86.40**	**63.19**
PatchCut	**70.49**	**84.52**	**64.80**

4.4. PASCAL

In this experiment, we present results for salient object segmentation using the PASCAL VOC 2010 dataset [10]. This dataset is more challenging because the images are from 20 object classes with large pose, shape and appearance variations and occlusions. Li *et al*. [22] collect salient object segmentation masks from human subjects for 850 images in the validation set. We use these images as the test set. Note that salient object segmentation masks may not be binary as subjects may disagree on the choice of salient objects as shown in Figure 10.

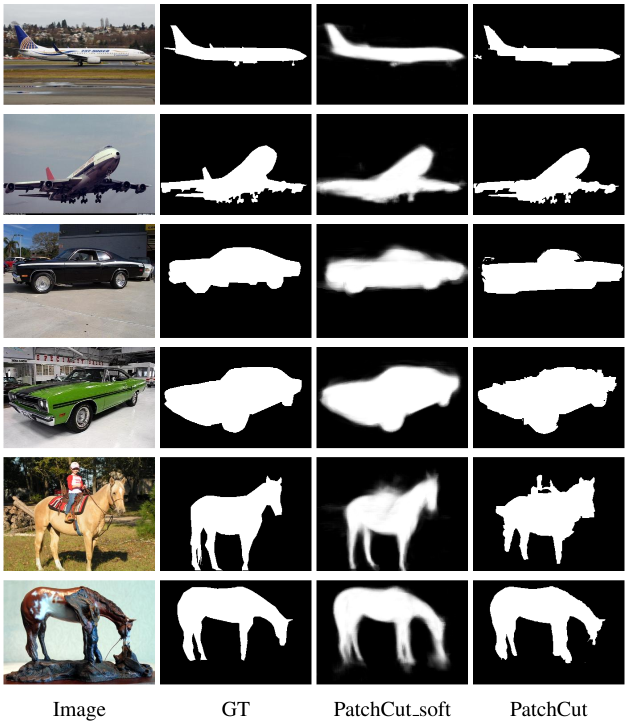

Figure 8: Qualitative results on Object Discovery.

On the other hand, we use all the images in the training set to build our example database, and collect salient object segmentation ground truth in a similar way as in [22]. Basically, for each image, we use the semantic labeling provided by [27] as full segmentation, and ask 6 subjects to select the salient object regions by clicking on them, so the saliency value for each segment is defined as the number of clicks it receives divided by the number of subjects. Differently from previous experiments, we initialize the PatchCut algorithm with the saliency maps generated by the GBVS algorithm [11], and its results are denoted as GBVS_PatchCut_soft, GBVS_PatchCut_thres and GBVS_PatchCut. We mainly compare with the state-of-the-art algorithm, CPMC_GBVS, presented in [22] which also uses the GBVS saliency maps. Figure 10 shows some qualitative results from PatchCut and CPMC_GBVS for comparisons.

Quantitative evaluation. We convert the ground truth segmentation saliency maps into binary masks with three thresholds: 0.1, 0.3, 0.5. Larger threshold means that

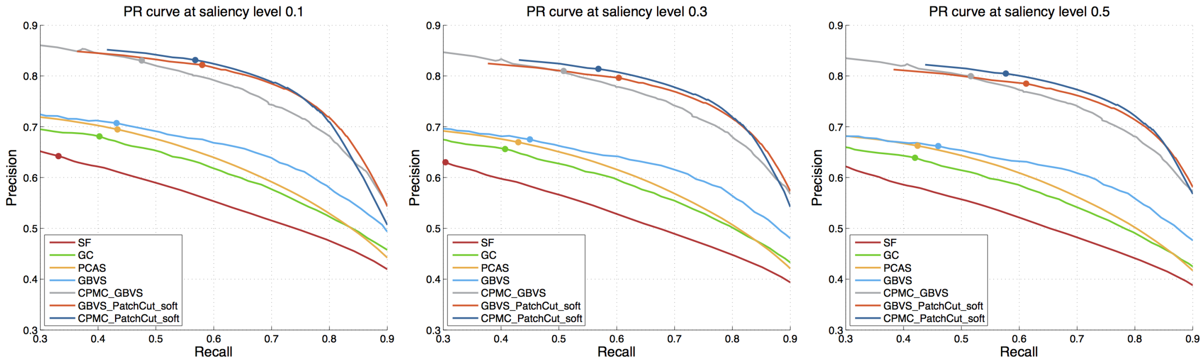

Figure 9: Comparing soft segmentation results at different saliency levels in terms of precision-recall curves. The dot on each curve indicates the operating point that gives the best F-score.

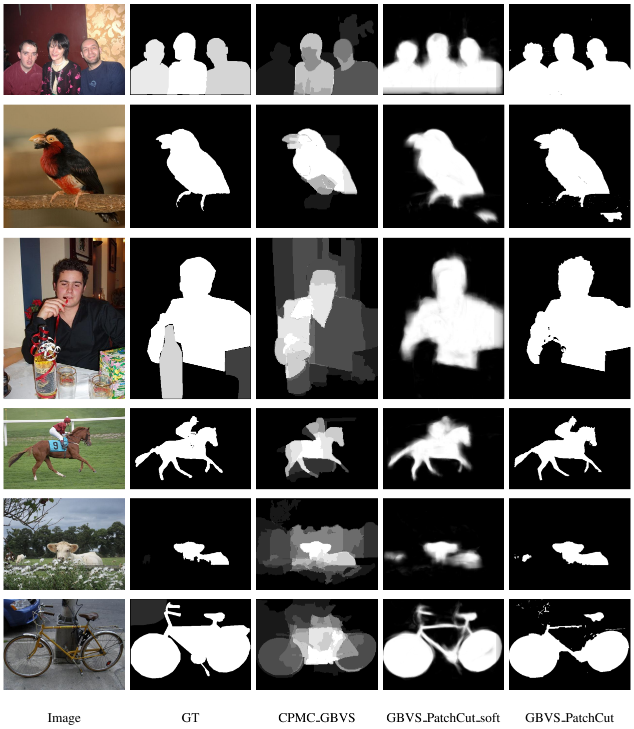

Figure 10: Comparing salient object segmentation results on PASCAL.

less objects with higher saliency values are selected in the ground truth. We first evaluate the soft segmentation masks (saliency maps) in terms of precision-recall curves. We compare the GBVS_PatchCut_soft results with CPMC_GBVS and three recent saliency algorithms, SF [28], GC [8] and PCAS [25]) in Figure. 9. Our algorithm (GBVS_PatchCut_soft) performs favorably against CPMC_GBVS and clearly above other saliency algorithms. Second, we evaluate binary segmentation results in terms of mean Jaccard scores. We convert the CPMC_GBVS results into binary segmentation by tuning the threshold, and find that its best mean Jaccard scores are obtained at the threshold 0.3. Table 4 shows that our algorithm (GBVS_PatchCut) performs slightly better than CPMC_GBVS, especially at low saliency levels. This result also means that our algorithm tends to select more objects than CPMC_GBVS (see examples in Figure 10).

Table 4: Jaccard scores on PASCAL.

Saliency level	0.1	0.3	0.5
GBVS_GrabCut	45.84	45.25	44.90
CPMC_GBVS [22]	59.43	60.58	60.75
GBVS_PatchCut_thres	60.08	60.22	59.27
GBVS_PatchCut	**62.02**	**62.15**	**61.14**
CPMC_PatchCut_thres	61.37	62.64	62.76
CPMC_PatchCut	**63.74**	**64.92**	**64.97**

We also initialize our PatchCut algorithm with the soft segmentation masks generated by CPMC_GBVS, and its results are denoted as CPMC_PatchCut_soft, CPMC_PatchCut_thres and CPMC_PatchCut. With this high quality initialization, PatchCut clearly outperforms the state-of-the-art in terms of both precision-recall curve (CPMC_PatchCut_soft in Figure 9) and the mean Jaccard scores (CPMC_PatchCut in Table 4).

5. Conclusions

In this paper, we present a data-driven object segmentation algorithm using examples, which requires no offline training of category-specific models and generalizes well to novel objects. Our algorithm constructs an online structured label space for object segmentation by transferring local shape mask candidates from examples. The MRF labeling problem is decomposed into a set of independent label patch selection sub-problems that are easier to solve in parallel. Our algorithm operates in a coarse-to-fine manner and achieves leading results in many object segmentation benchmarks with low computational cost (about 10 seconds for segmenting a 200x200 image with unoptimized MATLAB code on a typical desktop).

Acknowledgements

This work is done partially when the first author was an intern at Adobe. The work is supported in part by NSF CAREER Grant #1149783 and NSF IIS Grant #1152576, and a gift from Adobe.

References

[1] Adobe Systems Inc. Photoshop. Creative Cloud, 2014.

[2] E. Ahmed, S. Cohen, and B. Price. Semantic object selection. In *CVPR*, 2014.

[3] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *ECCV*, 2010.

[4] L. Bertelli, T. Yu, D. Vu, and B. Gokturk. Kernelized structural svm learning for supervised object segmentation. In *CVPR*, 2011.

[5] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, 2002.

[6] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV*, 2001.

[7] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010.

[8] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *PAMI*, 2014.

[9] P. Dollár and C. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. ”http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html”.

[11] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[13] J. Kim and K. Grauman. Shape sharing for object segmentation. In *ECCV*, 2012.

[14] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013.

[15] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo. Structured class-labels in random forests for semantic image labelling. In *ICCV*, 2011.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[17] D. Kuettel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*, 2012.

[18] M. P. Kumar, P. Torr, and A. Zisserman. Obj cut. In *CVPR*, 2005.

[19] D. Larlus and F. Jurie. Combining appearance models and markov random fields for category level object segmentation. In *CVPR*, 2008.

[20] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009.

[21] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, 2006.

[22] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014.

[23] Y. Li, D. Tarlow, and R. Zemel. Exploring compositional high order pattern potentials for structured output learning. In *CVPR*, 2013.

[24] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *PAMI*, 33(12):2368–2382, 2011.

[25] R. Margolin, A. Tal, and L. Zelnik-Manor. What makes a patch distinct? In *CVPR*, 2013.

[26] P. Marquez-Neila, P. Kohli, C. Rother, and L. Baumela. Nonparametric higher-order random fields for image segmentation. In *ECCV*, 2014.

[27] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.

[28] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012.

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

[30] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[31] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn reranking. In *CVPR*, 2012.

[32] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013.

[33] H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *CVPR*, 2011.

[34] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, and Z. Tu. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*, 2014.

[35] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012.

[36] J. Yang, S. Safar, and M.-H. Yang. Max-margin boltzmann machines for object segmentation. In *CVPR*, 2014.

[37] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011.