# Learning Image-Specific Parameters for Interactive Segmentation

Zhanghui Kuang[1] Dirk Schnieders[1] Hao Zhou[1]
Kwan-Yee K. Wong[1] Yizhou Yu[1] Bo Peng[2]
[1]The University of Hong Kong [2]The Hong Kong Polytechnic University

## Abstract

*In this paper, we present a novel interactive image segmentation technique that automatically learns segmentation parameters tailored for each and every image. Unlike existing work, our method does not require any offline parameter tuning or training stage, and is capable of determining image-specific parameters according to some simple user interactions with the target image. We formulate the segmentation problem as an inference of a conditional random field (CRF) over a segmentation mask and the target image, and parametrize this CRF by different weights (e.g., color, texture and smoothing). The weight parameters are learned via an energy margin maximization, which is solved using a constraint approximation scheme and the cutting plane method. Experimental results show that our method, by learning image-specific parameters automatically, outperforms other state-of-the-art interactive image segmentation techniques.*

## 1. Introduction

In image segmentation, we aim at separating an object of interest from the rest of the image. This is useful for pasting the object into a new context, and has applications in computational photography, image synthesis, and visual effects for film making. Unfortunately, no fully automatic system has been shown to be accurate, robust, and unambiguous for all sorts of challenging inputs. On the other hand, semi-automatic interactive image segmentation methods [6, 7, 12, 17] have produced very impressive results with a reasonable amount of user interactions. Very often there exist some parameters in a segmentation model, and inappropriate choice of such parameters may result in unsatisfactory segmentations. A common practice is to manually adjust the segmentation parameters until desired segmentations can be achieved on a few representative test images. The underlying assumption is that there exists a parameter setting that works for a variety of images represented by the few test images.

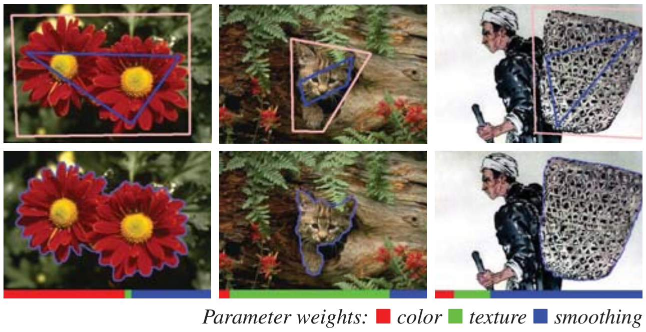


Figure 1. Segmentation results produced by the proposed method. Optimal parameter weights for color, texture, and smoothing are estimated from just a single image.

There do exist some research studies [5, 19, 1] that do not require manual parameter selection. They learn the parameters of an energy minimization from supervised training data using cross validation, pseudolikelihood, or structured support vector machines. Typically, the following assumptions are made in learning the parameters: *i) Ground truth segmentations are available for a large number of training images; ii) Model parameters learned from the training data can be generalized to unseen images.*

Unfortunately, the underlying assumptions for the aforementioned manual parameter selection and supervised training are not necessarily true. In general, different images require distinct sets of parameters that produce optimal segmentation results. For example, in Fig. 1, when segmenting the red flower, color is the main discriminative feature. When segmenting the cat, texture is the main discriminative feature. Yet, when segmenting the rice basket, stronger spatial smoothing is required to avoid over-segmentation. Sometimes, even for the same image, having constant model parameters is undesirable (see the last two rows of Fig. 7). Therefore, existing interactive segmentation methods based on training data cannot be expected to give optimal results because they use a constant set of parameters for all images.

For an interactive image segmentation system to achieve optimal results, it becomes necessary to find a parameter setting tailored for each and every image. When there exists ambiguity in segmentation, it is also necessary to find

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

a parameter setting most consistent with the intention of a specific user. To avoid extensive manual intervention while still meeting these requirements, the system needs to be sufficiently “intelligent” to determine which types of image regions should belong to the foreground according to the hints provided by scarce user interactions. The system should be able to understand the user’s intention and learn the relative importance of different features.

In this paper, we propose an interactive segmentation method based on just a single image (i.e., the image to be segmented). A conditional random field (CRF) is parameterized and iteratively solved to find the optimal parameters (weights for color, texture, and smoothing terms) that globally solve the segmentation. To do this, we first start with the user’s interaction as hard constraints and generalize it to unlabeled regions by maximizing an energy margin. This ambiguous problem becomes tractable with a novel constraint approximation scheme. Since all parameters of the CRF are learned automatically just from the image to be segmented, the system does not require any training images or hand-tuned parameters. As a result, the proposed method can be applied to different image domains and determine the optimal parameters based on a specific image.

## 2. Related Work

An interactive image segmentation technique based on graph cut was first presented in [6]. Since then, a large number of interactive methods have been proposed, including Grabcut [17], lazy snapping [12], random walker [8], geodesic matting [2], and TVSeg [22].

These methods usually require manual setting of some parameters. Researchers have also developed learning methods for estimating such parameters automatically. In [5], Blake *et al*. introduced an adaptive Gaussian Mixture Markov Random Field method that can learn Ising parameters from training images. Their extension to graph cut results in a system that puts less burden on the user. More recently, Szummer *et al*. [19] proposed an efficient learning method for multiple parameters. Their learning is formulated as minimizing a loss function over training images. Both of the above learning methods require ground truth segmentations, and estimate constant parameters that cannot be expected to give optimal results for different classes of images.

To overcome the above problems, Peng and Veksler [16] proposed an interesting method that learns image-specific parameters for the graph cut segmentation algorithm. Their classifier does not require ground truth segmentations, but instead 10 segmentation results that are manually labeled as either ‘good’ or ‘bad’ for each image in the training data. Their method is computationally expensive (it uses a brute-force search) and does not scale well. In contrast to their work, our method efficiently optimizes a multi-dimensional parameter space and does not require such ’good’ / ’bad’ labels. Kirmizigül and Schlesinger [10] introduced an image-specific method that learned a single smoothing parameter. They proposed an incremental learning technique that produces a set of feasible parameters in each iteration, and requires the user to refine the segmentation in each iteration. Since they used parametric max-flow, their method cannot be extended to multiple dimensions.

Parameter learning is also found in related fields, such as image labeling [3, 9, 11, 14, 15]. In contrast to these works, our method learns the parameters from incomplete ground truth (from simple user interactions) and does not require multiple images. To the best of our knowledge, our proposed method is the first to learn multiple CRF parameters from just a single image.

## 3. Parameter Learning from a Single Image

Image segmentation can be casted as an inference of a conditional random field over an image mask $\mathbf{y} = [y_1, \cdots, y_N]$ and an image $\mathbf{x} = [x_1, \cdots, x_N]$ with the form

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \frac{1}{Z(\mathbf{x}, \mathbf{w})} e^{(-E(\mathbf{y}, \mathbf{x}, \mathbf{w}))}, \tag{1}$$

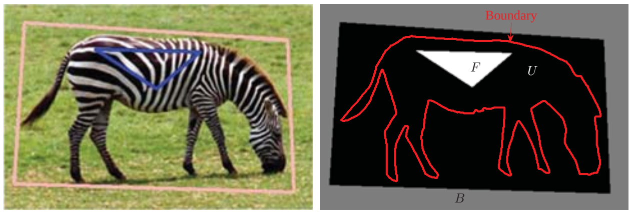
where $\mathbf{w}$ is a parameter vector and $Z(\mathbf{x}, \mathbf{w})$ is a partition function. The energy function is usually given by

$$\begin{aligned} E(\mathbf{y}, \mathbf{x}, \mathbf{w}) &= \sum_{i \in \nu} \sum_{k \in K} w_k^{\mathrm{d}} d_k(y_i, \mathbf{x}) \\ &+ \sum_{(i,j) \in \varepsilon} \sum_{t \in T} w_t^{\mathrm{s}} s_t(y_i, y_j, \mathbf{x}), \end{aligned} \tag{2}$$

where $\nu$ is the index set of graph nodes (image pixels), $\varepsilon$ is the index set of graph edges (adjacent pixel pairs), $K$ is the index set of features (e.g., color and texture), and $T$ is the index set of smoothness regularization terms. $d_k(y_i, \mathbf{x})$ and $s_t(y_i, y_j, \mathbf{x})$ are unary terms and pairwise smoothing terms respectively. We assume that both $d_k(y_i, \mathbf{x})$ and $s_t(y_i, y_j, \mathbf{x})$ are positive, $s_t(y_i, y_j, \mathbf{x}) = s_t(y_j, y_i, \mathbf{x})$, and $s_t(y_i, y_j, \mathbf{x}) = 0$ when $y_i = y_j$. We concatenate $\sum_{i \in \nu} d_k(y_i, \mathbf{x})$ and $\sum_{(i,j) \in \varepsilon} s_t(y_i, y_j, \mathbf{x})$ to form a vector function $\mathbf{\Psi}(\mathbf{x}, \mathbf{y})$. The energy function is therefore linear in the nonnegative parameter $\mathbf{w}$ with the form $E(\mathbf{y}, \mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathrm{T}} \mathbf{\Psi}(\mathbf{x}, \mathbf{y})$.

### 3.1. Maximizing an Energy Margin

In our interactive image segmentation system, we obtain a trimap based on simple user interactions. Let $F$, $B$ and $U$ be the index sets for foreground, background and unknown (unlabeled) pixels, respectively, labeled by the user (see Fig. 2). Let $\mathbf{y}_F$, $\mathbf{y}_B$ and $\mathbf{y}_U$ denote subsets of $\mathbf{y}$ indexed by $F$, $B$ and $U$ respectively. Our method is based on the principle that the final segmentation result should not



(a) (b)

Figure 2. Interactions for image segmentation. (a) Original image with superimposed polygons drawn by a user. (b) Labeled foreground ($F$), background ($B$), and unknown ($U$) regions. The contour in $U$ is the true foreground boundary.

contradict with the user input. Let the ground truth segmentation mask be $\bar{\mathbf{y}}$, with $\bar{\mathbf{y}}_F = \mathbf{1}$, $\bar{\mathbf{y}}_B = \mathbf{0}$, and $\bar{\mathbf{y}}_U$ a binary $|U|$-dimensional vector. The energy for the ground truth segmentation should not be bigger than any other mask $\mathbf{y}$, i.e.,

$$\mathbf{w}^{\mathrm{T}}\mathbf{\Psi}(\mathbf{x},\mathbf{y}) - \mathbf{w}^{\mathrm{T}}\mathbf{\Psi}(\mathbf{x},\bar{\mathbf{y}}) \geq 0 \quad \forall \mathbf{y} \neq \bar{\mathbf{y}}. \tag{3}$$

The inequalities in (3) enforce valid constraints on $\mathbf{w}$. However, typically there will be multiple solutions for which the set of inequalities is feasible. To specify a unique solution, we maximize the separation margin (suppose to be positive) given by the energy difference between the segmentation mask $\hat{\mathbf{y}} = \arg\min_{\mathbf{y}\neq\bar{\mathbf{y}}}\mathbf{w}^{\mathrm{T}}\mathbf{\Psi}(\mathbf{x},\mathbf{y})$ and the ground truth. This gives

$$\begin{aligned} &\max\ \gamma \\ &s.t. \quad \mathbf{w}^{\mathrm{T}}\delta(\mathbf{x},\mathbf{y},\bar{\mathbf{y}}) \geq \gamma \quad \forall \mathbf{y} \neq \bar{\mathbf{y}}, \|\mathbf{w}\| = 1, \end{aligned} \tag{4}$$

where $\delta(\mathbf{x},\mathbf{y},\bar{\mathbf{y}}) = \mathbf{\Psi}(\mathbf{x},\mathbf{y}) - \mathbf{\Psi}(\mathbf{x},\bar{\mathbf{y}})$. Maximizing the energy margin regularizes the ambiguous problem. Equation (4) can be reformulated into a standard quadratic problem as

$$\begin{aligned} &\min \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &s.t. \quad \mathbf{w}^{\mathrm{T}}\delta(\mathbf{x},\mathbf{y},\bar{\mathbf{y}}) \geq 1 \quad \forall \mathbf{y} \neq \bar{\mathbf{y}}. \end{aligned} \tag{5}$$

To avoid no solutions and to be robust to noise, a positive slack variable $\xi$ is introduced, and the energy function is optimized with a soft-margin criterion

$$\begin{aligned} &\min \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\xi \\ &s.t. \quad \mathbf{w}^{\mathrm{T}}\delta(\mathbf{x},\mathbf{y},\bar{\mathbf{y}}) \geq 1 - \xi \quad \forall \mathbf{y} \neq \bar{\mathbf{y}}, \xi \geq 0, \end{aligned} \tag{6}$$

where $C$ ($C = 5$ for all our experiments) balances the loss and regularization terms. Equation (6) just considers a 0-1 loss function which penalizes all non-ground truth masks $\mathbf{y}$ with 1. To penalize a mask with larger difference to the ground truth more severely than one with a smaller difference, we introduce a re-scale margin inspired by [20]

$$\begin{aligned} &\min \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\xi \\ &s.t. \quad \mathbf{w}^{\mathrm{T}}\delta(\mathbf{x},\mathbf{y},\bar{\mathbf{y}}) \geq \Delta(\mathbf{y},\bar{\mathbf{y}}) - \xi \quad \forall \mathbf{y} \neq \bar{\mathbf{y}}, \xi \geq 0, \end{aligned} \tag{7}$$

where $\Delta(\mathbf{y},\bar{\mathbf{y}})$ is the hamming distance defined as

$$\Delta(\mathbf{y},\bar{\mathbf{y}}) = \sum_{i\in\nu} I(y_i \neq \bar{y}_i), \tag{8}$$

and $I(\cdot)$ is an indicator function. There are two challenges in solving (7), namely

1. $\bar{\mathbf{y}}$ is only partially known, and
2. the number of constraints is huge ($2^N - 1$).

For the first problem, we propose a novel constraint approximation scheme to simplify the original optimization problem. For the second problem, we use the cutting plane algorithm [21] to overcome exponential number of constraints.

### 3.2. Constraint Approximation and Analysis

The ground truth at the unlabeled region $\bar{\mathbf{y}}_U$ is fixed but unknown. It is impossible for us to rewrite the constraints in (7) into inequalities parameterized by $\mathbf{w}$ only. To make the optimization problem tractable, we approximate the constraints and optimize the following problem:

$$\begin{aligned} &\min \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\xi \\ &s.t.\ \mathbf{w}^{\mathrm{T}}\delta(\mathbf{x},\mathbf{y},\bar{\mathbf{y}}) \geq \Delta(\mathbf{y},\bar{\mathbf{y}}) - \xi \ \ \forall \mathbf{y} \neq \bar{\mathbf{y}}, \mathbf{y} \in \Omega, \xi \geq 0, \end{aligned} \tag{9}$$

where $\Omega = \{\mathbf{y}|\mathbf{y}_U = \bar{\mathbf{y}}_U \wedge \mathbf{y}_{J^b} = \bar{\mathbf{y}}_{J^b}\}$ with $J = F \cup B$ and $J^b = \{p|p \in J \wedge \exists p^{'} \in U, (p, p^{'}) \in \varepsilon\}$. Intuitively, $J^b$ is the boundary pixel set of labeled regions and $\Omega$ is a limited segmentation mask space, whose elements are image masks having identical labels with ground truth in the unlabeled regions and along the boundary of labeled regions. For any $\mathbf{y} \in \Omega$, $\delta(\mathbf{x},\mathbf{y},\bar{\mathbf{y}})$ and $\Delta(\mathbf{y},\bar{\mathbf{y}})$ depend only on the mask of labeled regions $\mathbf{y}_J$ and $\bar{\mathbf{y}}_J$, and (9) can be rewritten as

$$\begin{aligned} &\min \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\xi \\ &s.t.\ \mathbf{w}^{\mathrm{T}}\delta(\mathbf{x},\mathbf{y}_J,\bar{\mathbf{y}}_J) \geq \Delta(\mathbf{y}_J,\bar{\mathbf{y}}_J) - \xi \\ &\forall \mathbf{y}_J \neq \bar{\mathbf{y}}_J, \mathbf{y}_{J^b} = \bar{\mathbf{y}}_{J^b}, \xi \geq 0, \end{aligned} \tag{10}$$

where

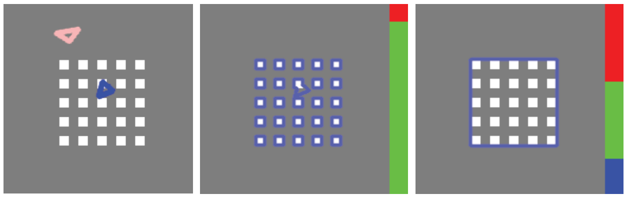
$$\delta(\mathbf{x},\mathbf{y}_J,\bar{\mathbf{y}}_J) = \begin{bmatrix} \sum\limits_{i\in J\setminus J^b} \mathbf{d}(y_i,\mathbf{x}) - \mathbf{d}(\bar{y}_i,\mathbf{x}) \\ \sum\limits_{\substack{i,j\in J \\ (i,j)\in\varepsilon}} \mathbf{s}(y_i,y_j,\mathbf{x}) - \mathbf{s}(\bar{y}_i,\bar{y}_j,\mathbf{x}) \end{bmatrix} \tag{11}$$

and

$$\Delta(\mathbf{y}_J,\bar{\mathbf{y}}_J) = \sum_{i\in J\setminus J^b} I(y_i \neq \bar{y}_i) \tag{12}$$

with $\mathbf{d}()$ and $\mathbf{s}()$ indicating concatenations of $d_k()$ and $s_t()$ in (2) respectively.

The above constraint approximation reduces the possible number of constraints from $2^N - 1$ to $2^{|J\setminus J^b|} - 1$, and leads to an approximation of the feasible domain. Let $\Theta$ and $\Theta^{'}$



*Parameter weights:* 🟥 *color* 🟩 *texture* 🟦 *smoothing*

Figure 3. Synthetic Gestalt example with small labeled regions. From left to right: Image with user input. Segmentation result with initial parameters. Segmentation result with learned parameters.

be the feasible sets of parameter $\mathbf{w}$ defined by (7) and (9) respectively. We have $\Theta \subseteq \Theta^{'}$ since constraints in (9) are a subset of constraints in (7). Note that the approximation error (i.e., the difference of optimal $\mathbf{w}$ between (7) and (9) ) depends on $|U|$ ( i.e., the number of unlabeled pixels). In addition, it depends on image statistics of the labeled and unlabeled regions. We confirmed experimentally on hundreds of images that such an approximation is reasonable to determine a good set of parameters, resulting in segmentations superior to most of the existing methods.

Although the approximation error depends on $|U|$, it does not mean we need to increase user's burden in learning a parameter configuration which can achieve a high quality segmentation due to two reasons. First, users can easily construct simple polygons to cover large areas. Therefore, $|U|$ can be small with limited interactions. Second, as long as feature statistics of the labeled and unlabeled regions are similar (also required by all other global segmentation methods), our algorithm can learn parameters well with small labeled regions. We tested the proposed method on synthetic data (see Fig. 3). It can be seen that our algorithm can learn strong smoothing regularization to produce a segmentation with spatial proximity of small blocks according to small labeled regions.

The constraints in (9) encourage the final segmentation mask $\tilde{\mathbf{y}}$ with learned parameters $\tilde{\mathbf{w}}$ to be consistent with $\bar{\mathbf{y}}$ in regions $J$. Therefore, it can avoid over-segmentation. In addition, the $\mathbf{w}^{\mathrm{T}}\mathbf{w}$ term in the objective function penalizes high smoothness regularization terms to avoid under-segmentation.

### 3.3. Structural Learning

Although the above approximation scheme has greatly reduced the number of constraints, there is still an exponential number of constraints in (9) with respect to $\left|J \setminus J^b\right|$. We incrementally find a small set of most violated constraints that ensures a sufficiently accurate solution based on the cutting plane algorithm [21]. This is summarized in Algorithm 1. Our algorithm converges in less than 25 iterations for most of the images, potentially allowing it to run in realtime [23].

**Algorithm 1** Structural learning for (10)

```
1: Input $\mathbf{x}$, $F$ and $B$. Define $\bar{\mathbf{y}}$ according to $F$ and $B$.
2: Empty the most violated constraint set $S = \emptyset$ .
3: Initialize parameters $\mathbf{w}$.
4: repeat
5:     Find the most violated constraint by $\mathbf{y}_J^* \leftarrow \arg\min_{\mathbf{y}_J} \mathbf{w}^{\mathrm{T}}\delta(\mathbf{x}, \mathbf{y}_J, \bar{\mathbf{y}}_J) - \Delta(\mathbf{y}_J, \bar{\mathbf{y}}_J)$ s.t. $\mathbf{y}_{J^b} = \bar{\mathbf{y}}_{J^b}$,
       and set $S = S \cup \{\mathbf{y}_J^*\}$.
6:     Learn parameter $\mathbf{w}$ by solving the following problem:
           $\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\xi$
           $s.t. \quad \mathbf{w}^{\mathrm{T}}\delta(\mathbf{x}, \mathbf{y}_J, \bar{\mathbf{y}}_J) \geq \Delta(\mathbf{y}_J, \bar{\mathbf{y}}_J) - \xi \quad \forall \mathbf{y}_J \in S, \xi \geq 0.$
7: until $\mathbf{w}$ does not change any more.
```

Line 5 of Algorithm 1 searches for the most violated constraint for the current parameter $\mathbf{w}$ by minimizing an energy function. It can be optimized by graph cut. In order to guarantee the constraint $\mathbf{y}_{J^b} = \bar{\mathbf{y}}_{J^b}$ is satisfied, we set the boundary of $F$ to be foreground and that of $B$ to be background as hard seed points. Line 6 updates the parameter $\mathbf{w}$ according to the most violated constraint set $S$.

## 4. Implementation Details

We employ two feature descriptors to describe color and texture respectively (i.e., $|K| = 2$ in (2)). Color is represented by intensity (gray images) or RGB (color images) values for each pixel. We build two global histograms to model background and foreground colors respectively. The number of bins for each feature dimension is fixed at 40. The likelihood based on color $p_c(\mathbf{x}|y_i)$ can be obtained directly from the histogram.

To extract texture features, classical structure tensors [18] are computed. Foreground and background texture models are represented again by two global histograms, and the likelihood for texture $p_t(\mathbf{x}|y_i)$ is estimated in the same way as color.

After computing the likelihoods, the data term in (2) is defined by $d_k(y_i, \mathbf{x}) = -\log(p_k(\mathbf{x}|y_i))$ with $k \in \{c, t\}$. We use contrast-based regularization [17] as the single smoothing term $s_t(y_i, y_j, \mathbf{x})$. Note that all the feature descriptors and smoothing regularization term can be preprocessed.

## 5. Experimental Results

We evaluated our proposed approach by three groups of experiments. Segmentation accuracy was measured by overall pixel accuracy $M_a$ and the foreground overlapping ratio $M_o$ as in [4]. Here, performance of segmentation was measured only in unlabeled regions since the labeled region can always be segmented correctly as hard seed points. We defined $M_{\mathrm{a}} = \frac{1}{|U|}\sum_{i \in U} I(y_i = \bar{y}_i)$ and $M_{\mathrm{o}} = \sum_{i \in U} I(y_i \wedge \bar{y}_i) / \sum_{i \in U} I(y_i \vee \bar{y}_i)$.

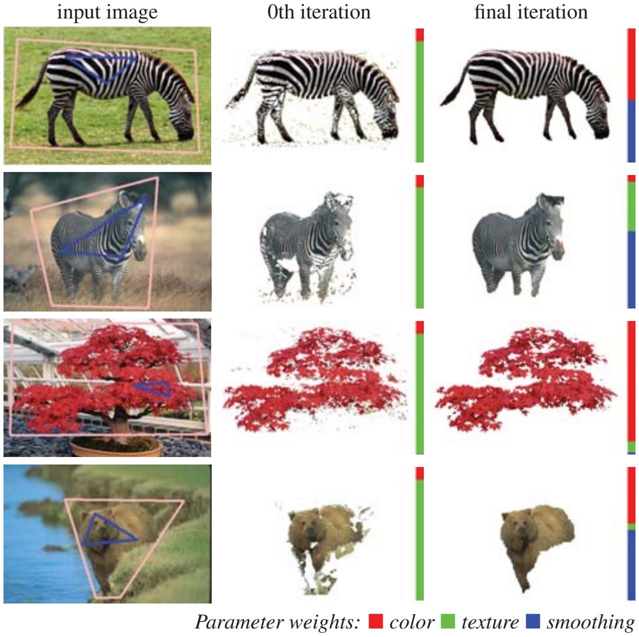
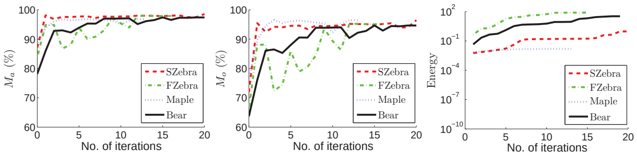


Figure 4. Segmentation results and learned parameters for SZebra, FZebra, Maple, and Bear (from top to bottom).



(a) (b) (c)

Figure 5. Performance of the proposed method. (a) Overall accuracy. (b) Foreground overlapping ratio. (c) Energy of the segmentation result.

## 5.1. Performance of Our Approach

We first tested our algorithm on four images (see Fig. 4) to demonstrate its ability to iteratively learn the parameter according to image contexts and user interactions. Let us name the images in Fig. 4, from top to bottom, as side zebra (SZebra), front zebra (FZebra), Maple and Bear. For both SZebra and FZebra, the initial segmentation results were poor with initialized parameters. This can be seen from the second column, which shows the segmentation results before the first iteration. For SZebra, foreground and background can be easily distinguished by color, which is reflected in the estimated parameter weights in the last column. For Maple, color is important. However, this time smoothing must be weak to avoid under-segmentation. For Bear, at first glance, it may seem that texture should be a more discriminative feature than color. However, our learned parameters indicate that color is the dominant feature. This suggests our intuition can sometimes be wrong, and therefore it is better to determine the parameters automatically.

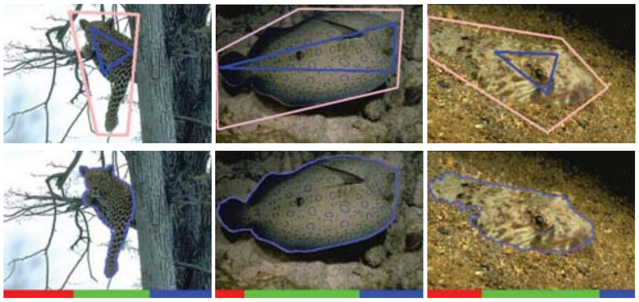


Figure 6. Challenging examples.

Fig. 5 shows the performance of our algorithm against the number of iterations. Both the overall pixel accuracy and overlapping ratio increase with fluctuations. The fluctuations become smaller and smaller as the algorithm converges. Such fluctuations happen because our algorithm always tries to find the most violated constraints. The energy of the segmentation result keeps increasing until convergence as our algorithm adds more and more constraints into the most violated constraint set $S$.

We also tested the proposed method on challenging images (see Fig. 6).

## 5.2. Comparison with Other Parameter Learning Methods

We compared our parameter learning method with three other parameter learning methods, namely the brute force method[1] (BFM) [5], learning CRFs using graph cut (LCGC) [19], and learning segmentation quality measurement[2] (LSQM) [16]. Note that both BFM and LSQM just learn the weight of the smoothing term by searching a discretized parameter space. Here we extend these methods to learning a three dimensional parameter. These extended methods are, however, impractical since they are very time-consuming. For BFM and LSQM, we descretized the parameter space $[0,\ 2] \times [0,\ 2] \times [1,\ 97]$ into $11 \times 11 \times 13$ samples using equal intervals.

The evaluation was performed on a challenging database of 50 images (selected from the Berkeley dataset [13] and various other internet sources) for which ground truth segmentations are available. For a fair comparison, the same user provided segmentation masks and the same vector functions $\mathbf{\Psi}(\mathbf{x}, \mathbf{y})$ were used. For methods that require training images, 5-fold, 10-fold, and 50-fold cross validations were implemented. Table 1 summarizes the average segmentation accuracy using the parameters learned by our proposed method and aforementioned three methods. We also segmented each image with all possible parameter samples to find the 'optimal' parameter that produced the best

[1] Weight of smoothing is determined by brute force search.

[2] Code provided by the authors

Table 1. Average segmentation accuracy compared with other parameter learning methods.

| Methods | | $M_{\rm a}$ (%) | $M_{\rm o}$ (%) |
|---|---|---|---|
| BFM [5] | 5-fold cross-validation | 95.7 | 90.0 |
| | 10-fold cross-validation | 95.8 | 90.0 |
| | 50-fold cross-validation | 95.8 | 90.0 |
| LSQM [16] | 5-fold cross-validation | 92.0 | 83.3 |
| | 10-fold cross-validation | 93.3 | 85.0 |
| | 50-fold cross-validation | 92.6 | 84.0 |
| LCGC [19] | 5-fold cross-validation | 95.7 | 89.8 |
| | 10-fold cross-validation | 95.7 | 90.0 |
| | 50-fold cross-validation | 95.7 | 90.0 |
| *Proposed method* | | *96.5* | *91.6* |
| Optimal parameter | | 97.1 | 92.8 |

segmentation result as compared with the ground truth. As expected, the average accuracy of the proposed method is lower than that with 'optimal' parameter since the feature statistics of the unlabeled and labeled regions are not similar enough for some of the images. However, it is closest to the optimal result compared with the other methods. This shows that each image should have its own set of optimal parameters. Although our method only slightly improves the accuracy, it makes sense since very small differences between segmentation and ground truth may still require much labor to refine in interactive segmentation.

Fig. 7 demonstrates the segmentation results of four examples using the different methods. For the first image, our method learned a weak smoothing regularization so the leg is segmented correctly. For the second image, our method learned strong texture weight and strong smoothing regularization to distinguish face from foreground and enhance spatial coherence. The third and fourth are the same image with different user interactions. It can be seen that our method understands user's intention.

## 5.3. Comparison with Other Interactive Methods

We qualitatively compared our image segmentation method with other state-of-the-art methods using default parameter setting. These methods include (i) graph cut[3] (GC) [6], (ii) random walker[4] (RW) [8], (iii, iv, v) three variations of geodesic matting[5] (with geodesics computed on likelihood image gradients (GM-LIG), image gradient (GM-IG), and smoothed image gradient (GM-SIG)) [2], (vi) lazy snapping[6] (LS) [12], (vii) TVSeg [7] (TS) [22] and (viii) Grabcut[8] [17].

Note that methods (i)∼(vii) all used the same user interaction. The evaluation result on a single image is shown in Fig. 8. It can be seen from the learned parameters of the proposed method that color is a dominant feature. We have selected this image to make it fair to those methods that do not use texture features. It can be seen that the proposed method outperforms all the other seven methods. For GC, the segmented foreground has artifacts around the boundary. For the other six methods, some parts of the foreground are not segmented which might be due to weak likelihood estimation or strong spatial smoothing.

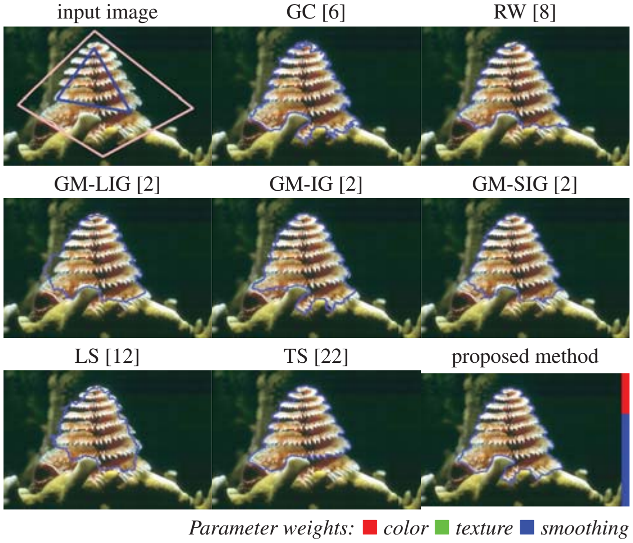


Figure 8. Comparison with other interactive image segmentation methods.

Grabcut uses a more simple user interaction. In Fig. 9, we compare the proposed method with Grabcut. Although the proposed method requires slightly more user interactions, it achieves significantly better results. In fact, one may find drawing two simple polygons (as in our method) takes less time than fitting a tight rectangle around an object (as in Grabcut). There are three reasons why our method outperforms Grabcut. First, our method uses both color and texture to distinguish foreground from background while Grabcut only uses color. Second, histograms (nonparametric models) are used in our approach while GMMs (parametric model) are employed in Grabcut. It is well-known in machine learning that non-parametric models in general can achieve better classification performance than parametric models in low dimension space. Third, our polygon interaction is more flexible than a rectangle.

# 6. Conclusions

In this paper, we have introduced a novel technique that automatically and simultaneously determines the optimal image segmentation and its associated segmentation parameters. Users are only required to draw two simple polygons on the target image to provide examples of their desired foreground and background regions. The energy function is parameterized with multiple weights (e.g., color, texture

[3] http://www.robots.ox.ac.uk/ vgg/research/iseg/
[4] http://cns.bu.edu/ lgrady/software.html
[5] http://www.robots.ox.ac.uk/ vgg/research/iseg/
[6] http://www.cs.cmu.edu/ mohitg/segmentation.htm
[7] http://gpu4vision.icg.tugraz.at/
[8] OpenCV 2.3

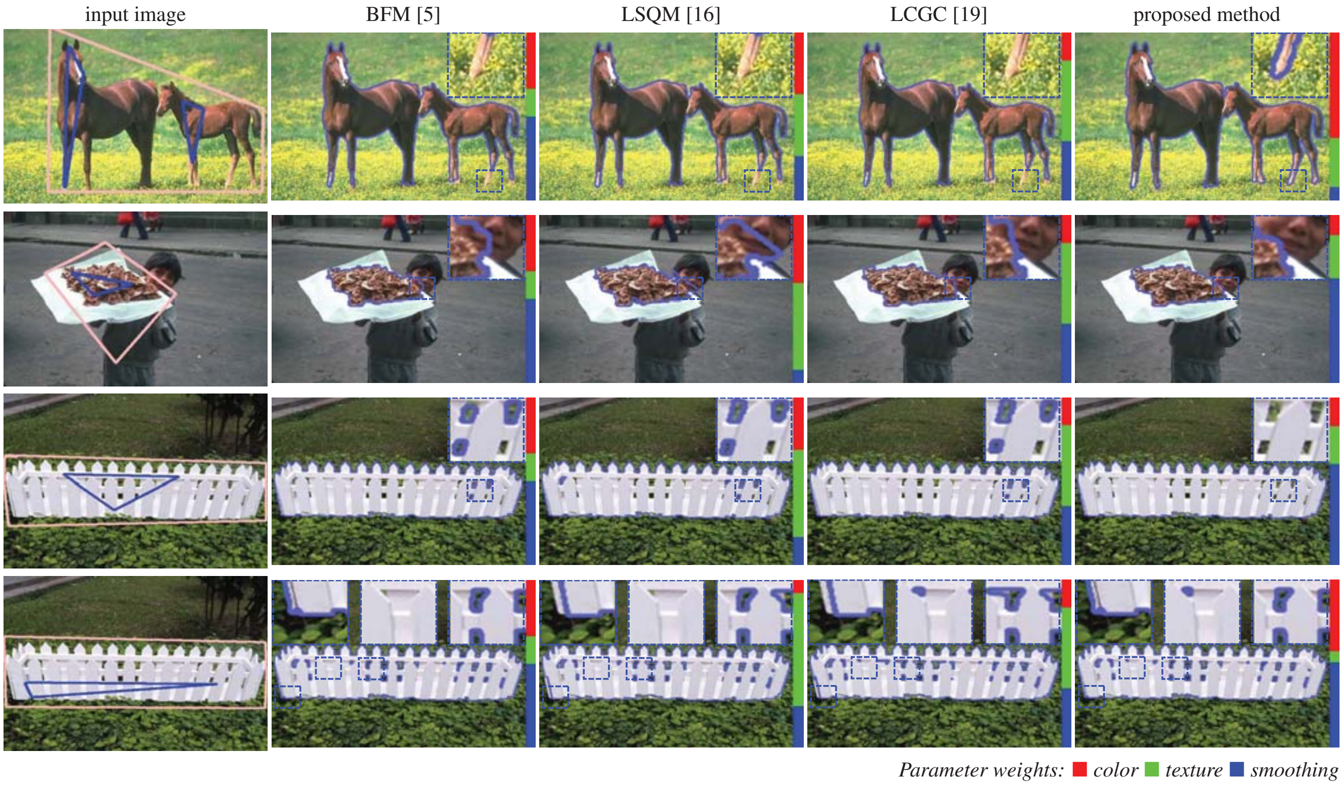


Figure 7. Comparison with other parameter learning methods.

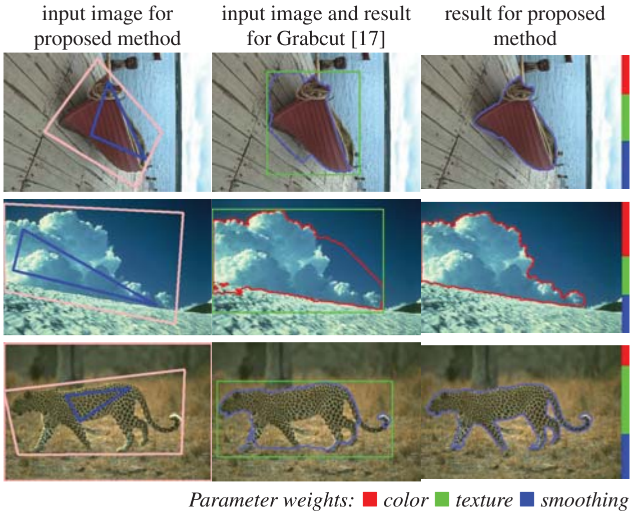

input image for proposed method
input image and result for Grabcut [17]
result for proposed method
Parameter weights: 🟥 color 🟩 texture 🟦 smoothing

Figure 9. Comparison with Grabcut.

and smoothing) and the segmentation mask is then solved by maximizing the energy margin iteratively. Thanks to the capability of learning image-specific parameters, our method demonstrates superior performance in segmentation quality compared with other state-of-the-art methods. Note that we have only used two global features (histograms of colors and textures) and one smoothing term in our experiments for demonstration purpose. Theoretically, energy terms of any feature (local or global) and any number of energy terms can be plugged into our framework easily to further improve the discriminative power of the segmentation method.

## 7. Acknowledgements

The authors would like to thank Lifeng Shang, Xiaolong Zhu and Miaomiao Liu for valuable comments on the paper preparation.

## References

[1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3D scan data. In *CVPR*, pages 169 – 176, 2005.

[2] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *IJCV*, 82:113–132, 2009.

[3] D. Batra, R. Sukthankar, and T. Chen. Learning class-specific affinities for image labelling. In *CVPR*, pages 1–8, 2008.

[4] L. Bertelli, T. Yu, D. Vu, and B. Gokturk. Kernelized structural SVM learning for supervised object segmentation. In *CVPR*, pages 2153–2160, 2011.

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

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

[7] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, and F. Segonne. Segmentation by transduction. In *CVPR*, pages 1–8, 2008.

[8] L. Grady. Random walks for image segmentation. *TPAMI*, 28(11):1768–1783, 2006.

[9] X. He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, pages 695–702, 2004.

[10] D. Kirmizigül and D. Schlesinger. Incremental learning in the energy minimisation framework for interactive segmentation. *Pattern Recognition*, pages 323–332, 2010.

[11] S. Kumar and M. Hebert. Discriminative random fields: a discriminative framework for contextual interaction in classification. In *ICCV*, pages 1150–1157, 2003.

[12] Y. Li, J. Sun, C. Tang, and H. Shum. Lazy snapping. *Transactions on Graphics*, 23(3):303–308, 2004.

[13] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423, 2001.

[14] T. Mensink, J. Verbeek, and G. Csurka. Learning structured prediction models for interactive image labeling. In *CVPR*, pages 833–840, 2011.

[15] S. Nowozin, P. Gehler, and C. Lampert. On parameter learning in CRF-based approaches to object class image segmentation. In *ECCV*, pages 98–111, 2010.

[16] B. Peng and O. Veksler. Parameter selection for graph cut based image segmentation. In *BMVC*, pages 160–170, 2008.

[17] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[18] M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *CVPR*, pages 699–704, 2003.

[19] M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In *ECCV*, pages 582–595, 2008.

[20] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, pages 94–104, 2004.

[21] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[22] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof. TVSeg - interactive total variation based image segmentation. In *BMVC*, pages 335–354, 2008.

[23] V. Vineet and P. J. Narayanan. Cuda cuts: Fast graph cuts on the GPU. In *CVPR Workshops*, pages 1–8, 2008.