Enhanced Figure-Ground Classification with Background Prior Propagation

Yisong Chen Antoni B. Chan

Abstract

[1] *We present an adaptive figure-ground segmentation algorithm that is capable of extracting foreground objects in a generic environment. Starting from an interactively assigned background mask, an initial background prior is defined and multiple soft-label partitions are generated from different foreground priors by progressive patch merging. These partitions are fused to produce a foreground probability map. The probability map is then binarized via threshold sweeping to create multiple hard-label candidates. A set of segmentation hypotheses is formed using different evaluation scores. From this set, the hypothesis with maximal local stability is propagated as the new background prior, and the segmentation process is repeated until convergence. Similarity voting is used to select a winner set, and the corresponding hypotheses are fused to yield the final segmentation result. Experiments indicate that our method performs at or above the current state-of-the-art on several datasets, with particular success on challenging scenes that contain irregular or multiple-connected foregrounds.*

1. Introduction

Figure-ground segmentation is a fundamental operation with a great potential in many vision applications [1]. It aims at producing a binary segmentation of the image, separating foreground regions from their background. Modern approaches include solutions based on graphs, statistics, information theory, or variational theory [2, 3, 4, 5]. Automatic segmentation in generic conditions is extremely difficult due to the broad diversity of visual cues in a natural image [6]. As a tradeoff, interactive methods [7, 8, 9] have produced impressive results with a reasonable amount of user guidance. The ideas of multiple hypotheses and classifier fusion have also been applied to segmentation studies [10, 11, 12]. Current state-of-the-art interactive segmentation methods suffer from several limitations, including a restrictive assumption about latent distributions [3], an inability to treat complicated scene topologies [9], or an inefficient similarity measure [13].

[1] This work was supported by national science funds and national basic research program of China (61421062, 61232014, 61173080, 2015CB351806). A.B.C. was supported by the Research Grants Council of Hong Kong, China (CityU 11200314).

Yisong Chen is with the EECS department, key laboratory of machine perception, Peking University (e-mail: chenyisong@pku.edu.cn).

Antoni B. Chan is with the department of computer science, the multimedia software engineering research centre (MERC), City University of Hong Kong (e-mail: abchan@cityu.edu.hk), and MERC-Shenzhen, Guangdong, China.

In this paper, we propose an iterative adaptive figure-ground classification method, which gives promising solutions in a broadly applicable environment. Foreground extraction is achieved by first generating a large amount of hypotheses through an iterated background prior propagation routine, then fusing most promising hypotheses to obtain the final result. The algorithm yields good result for challenging scenes in both segmentation accuracy and execution efficiency. It is not sensitive to difficult scene topology or loose bounding box, and reliably treats multi-connected, multi-hole foregrounds. Another advantage of our method is that the spatial smoothness term essential in popular conditional random fields (CRF) approaches is removed, and hence no additional learning algorithm is needed for tuning a smoothness parameter.

The rest of the paper is arranged as follows. Section 2 briefly reviews related work. Section 3 presents our figure-ground classification framework. Section 4 presents experimental results and Section 5 concludes the paper.

2. Related work

The four major aspects of figure-ground segmentation, related to our work, are the definition of prior knowledge, similarity measures, parameter tuning, and goodness evaluation. Prior knowledge can provide information about either the foreground or background, or both [3, 14]. It can be assigned by users in several forms, including bounding boxes or seed points [11, 15], to help define hard or soft constraints [2, 16]. Priors can also dynamically change or propagate throughout the segmentation process [7, 13].

Similarity measures are defined over feature spaces, based on appearance cues like color, shape, texture and gradient [5, 17]. As different features often characterize different aspects and are complementary, recent work has focused on mixed feature spaces [14, 17]. In particular, joint color-spatial feature have been successful in many vision applications [18, 19, 20]. Besides traditional Euclidean distance, similarity measures are often based on statistics or information theory [4, 13].

Regarding parameter tuning, a common practice is to learn the parameters via an energy minimization framework using training data and supervised learning [21, 22]. The underlying assumption is that there exists a parameter setting that works for a variety of images represented by the

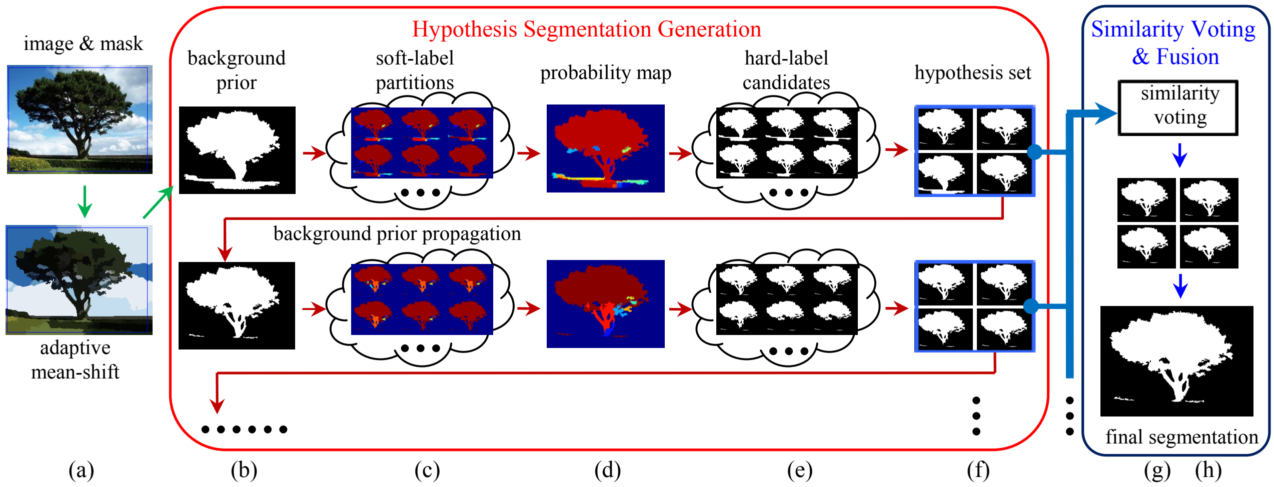


Figure 1. The pipeline of enhanced figure-ground classification with background prior propagation (EFG-BPP): (a) Original image with box mask (Sec. 3.2), and image patches from adaptive mean-shift (Sec. 3.3); The inputs of the "hypothesis segmentation generation" are the image patches. (b) the box mask helps define the initial background prior; (c) Different foreground priors generate multiple soft-label partitions (Sec. 3.5); (d) all soft-labels are combined into one foreground probability map (Sec. 3.6); (e) Thresholding the probability map forms a set of hard-label candidates (Sec. 3.7); (f) A set of hypotheses is selected using evaluations by various score functions (Sec. 3.7). The lower-right hypothesis of (f) is the result using the MSER score function, and is propagated as the background prior for the next segmentation round (Sec 3.8). The inputs of the "similarity voting & fusion" block are multiple hypothesis set. (g) The winning hypothesis set is selected using similarity voting; (h) The final segmentation is obtained by fusing the hypothesis set (Sec. 3.9).

training set, which can then be applied equally well to the test set. Unfortunately, this assumption is not necessarily true, and bears the risk of training set bias and bad generalization performance. Recent works prefer to find parameters that are adaptively set for each image [14, 23].

Finally, it is difficult for an approach that optimizes a single criterion to successfully segment all types of natural images, which contain a broad variety of visual patterns. Hence, recent works have developed good evaluation strategies to judge and combine multiple candidates [24, 25]. Although not fully exploited, many studies have mentioned the power of fusing complementary information from multiple hypotheses [11, 12, 26].

In this paper we propose an algorithm that extracts foreground by merging good hypotheses. Our method is distinct from previous multi-hypothesis approaches in the following two aspects: First, previous works [12, 24, 26] generate multiple hypotheses by varying segmentation parameters or employing different over-segmentation algorithms, resulting in multiple K-way segmentations. In contrast, our task of foreground extraction is a binary segmentation task, and we propose a novel method for generating binary hypotheses using a tree-structured likelihood propagation followed by multiple evaluation. In particular, since it is unknown which part of the bounding box contains the foreground, we generate candidate segmentations by initializing the tree with various regions as the foreground. Second, most previous approaches lack effective mechanisms to choose a best one from multiple hypotheses in general environment and can only resort to some extra learning process [10, 11, 25]. In our method we use the idea of similarity voting, which does not require a learning process, to fuse soft-segmentations into a probability map and hard segmentations into a final segmentation.

A preliminary version of our work was introduced in [23]. The new method presented in this paper improves over the original [23] in the following aspects: 1) we use a soft-label scheme based on foreground likelihoods, which leads to significant improvement in the segmentation quality of fine details; 2) we introduce an iterative scheme to propagate the background prior, which increases the accuracy of the segmentation; 3) maximally stable extremal regions (MSER) are used to define a novel score function for goodness evaluation and background prior propagation, which effectively prevent over-propagation of the background and better handles loose bounding boxes; 4) similarity voting is extended for probability map generation and hypothesis set selection, which yields a robust classifier fusion from multiple hypotheses.

3. Enhanced figure-ground classification with background prior propagation

In this section, we propose an enhanced adaptive figure-ground classification framework with background prior propagation. Our framework is based on fusing multiple candidate segmentations, and is guided by two underlying principles: 1) voting or fusion of multiple candidates often has better chance than optimization of a single score function in classification tasks, as long as the candidates are reasonably generated (even by very weak classifier), and in general, more votes provide higher confidence; 2) regarding the fusion strategy, priority should be given to the candidates sharing more similarities, or the hypothesis set with higher intra-similarity, to yield a result that satisfies more participants. We denote our fusion strategy of multiple candidate segmentations as *similarity voting*. The idea of similarity voting can be seen as an extension of the well-known majority vote principle in classifier fusion [27]. It plays an important role throughout our algorithm.

3.1. Algorithm overview

Fig. 1 shows the pipeline of our figure-ground segmentation algorithm. Our algorithm consists of two main stages: 1) hypothesis segmentation generation, and 2) similarity voting & fusion. In the first stage, the user box specifies the initial background, and a large number of candidate segmentations are created, from which a set of best hypothesis segmentations are selected. By using one of the hypotheses to define the new background prior, the segmentation process is repeated to form several hypothesis sets. In the second stage, the best hypothesis set is automatically selected by intra-similarity comparison, and the corresponding hypotheses are fused to form the final segmentation.

3.2. Bounding box assignment

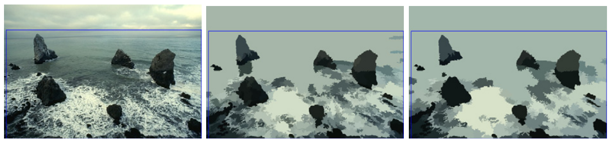
Our algorithm is based on a user-specified mask box that helps define the initial background prior, as in previous approaches [7, 15]. Either inside or outside of the box can be defined as the background mask, which is assumed to only contain background pixels. The complement of the background mask is the foreground mask, which may contain both foreground and background elements. The mask box can flexibly handle various cases of partially-inside or multiply connected foregrounds.

3.3. Image patches by adaptive mean-shift

Defining a segmentation as the grouping of nonoverlapping regions has become popular due to its advantages in information transfer and computational efficiency [28, 29, 30]. In our work, we generate an over-segmented image using an adaptive mean-shift algorithm (Fig. 1a). Mean-shift [28] is a non-parametric clustering method, which is based on finding the modes of the kernel density estimate in the feature space. We choose the mean-shift algorithm because mean-shift patches are better described statistically in comparison to other super-pixel generators [17]. For each pixel we extract a 5D feature vector in a joint color-spatial feature space,

$$f_{x,y} = \left(L(x,y), a(x,y), b(x,y), x, y\right), \tag{1}$$

where (x,y) are the 2D pixel coordinates and $(L(x,y), a(x,y), b(x,y))$ are the corresponding pixel values in the Lab color space. We use the Lab space because it is better modeled by a normal distribution in comparison to RGB [31]. We then apply the mean-shift algorithm to cluster the feature vectors, with pixels in each cluster forming an image patch. The result is a partitioning of the original image I into a set of non-overlapping patches $R_I=\{p_1, p_2, ..., p_n\}$, where p_i is an image patch (Fig. 2). Since we use a joint color-spatial feature space, the image patches tend to be visually similar and spatially compact.



(a) original & mask (b) initial patches (c) adaptive patches

Figure 2. Example of over-segmentation by adaptive mean-shift.

In the mean-shift algorithm we use two bandwidth parameters for the kernel, h_s for the spatial features (x, y) and h_r for the color features (L, a, b). The bandwidth controls the smoothness of the estimate, and ultimately determines the number of mean-shift patches obtained [28]. Different initial settings lead to different image patch sets, only some of which are suitable for the subsequent classification [32]. This is illustrated in Figure 2, where the default setting of h_s=7 and h_r=6 generates cluttered patches and fails to transfer the background prior reliably into the region of interest. Nevertheless, a bandwidth setting h_s=10 and h_r=8.6 (determined by our adaptive scheme) generates more consolidated patches.

Based on the relationship between the bandwidth parameters and the covariance matrix of the multivariate normal distribution [33], we propose the following scheme to adaptively set the bandwidths. First, an initial mean-shift segmentation is performed with the default bandwidths h_s=7 and h_r=6. Next, patches overlapped with the foreground mask region are collected into the set F_0, and the 3x3 covariance matrix $\Sigma_i^{(rr)}$ of the color features, and the 2x2 covariance matrix $\Sigma_i^{(ss)}$ of the spatial features are calculated for each patch p_i. Finally, the adaptive bandwidths are estimated by averaging the color/spatial variances over all collected patches in F_0,

$$h_s = \left\lfloor \sqrt{\frac{1}{|F_0|}\sum_{i\in F_0}\frac{1}{2}\mathrm{trace}(\Sigma_i^{(ss)})} \right\rfloor, \quad h_r = \sqrt{\frac{1}{|F_0|}\sum_{i\in F_0}\max(\mathrm{diag}(\Sigma_i^{(rr)}))}\;. \tag{2}$$

Whereas h_s is estimated from the variance in both x- and y-coordinates, h_r is estimated by averaging the Lab components with largest variance, due to the observation that this component often dominates in the Lab space. The mean-shift algorithm is run again with the adapted bandwidths to obtain the final patches[2]. By using the statistics

[2] The bandwidth could be updated iteratively with multiple runs of mean-shift. However, we did not see any improvement using more than one update, and the iterations sometimes did not converge to a fixed value, but instead oscillated within a small range.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIP.2015.2389612

from the foreground mask, our approach tunes the bandwidth parameters to form better representative patches.

In some cases, when the background contains repetitive cluttered textures, the adaptive mean-shift may still produce too many image patches, and cause the background patches to be mainly distributed along the mask boundary (as in Fig. 3). This will lead to a poor estimate of the background prior and a poor segmentation. We suggest a simple heuristic to identify and circumvent these cases. If the initial mean-shift creates too many patches (>300) within the mask region, we double the bandwidths (h_s=14, h_r=12) to group together pixels in a larger neighborhood to make larger patches. Larger bandwidths merge small patches into bigger ones and extend background prior deeper into the mask region.

(a) (b) (c)

Figure 3. Larger bandwidths for cluttered textures: (a) image; mean-shift patches using (b) small and (c) large bandwidths.

3.4. Similarity measure between patches

In the next stage of the segmentation pipeline, patches are gradually assigned likelihood labels, based on their similarities to the patches labeled earlier. We will represent a region as the set of its patches. Hence, we must first define a suitable dissimilarity measure between two patches, and between a patch and a region.

To remain consistent with the underlying probabilistic framework of the mean-shift algorithm, we model each mean-shift patch p_i as a multivariate normal distribution $N(\mu_i,\Sigma_i)$ in the 5D feature space defined in (1), where the mean vector μ_i and the covariance matrix Σ_i are estimated from the patch. All patches are eroded with a 3x3 structural element to avoid border effects.

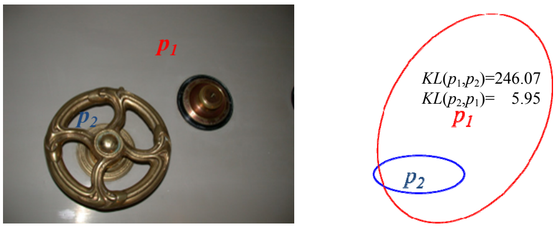
The Kullback-Leibler divergence (KLD) can be used to measure dissimilarity between two distributions, but is not symmetric [34]. Here, we use the minimum KLD between two patches as our dissimilarity measure,

$$D(p_1,p_2)=\min(KL(p_1,p_2),KL(p_2,p_1)), \tag{3}$$

where patches p_1 and p_2 are represented by two Gaussians, with distributions $N(\mu_1,\Sigma_1)$ and $N(\mu_2,\Sigma_2)$, and the KLD between two d-dimensional Gaussians is [34]

$$KL(p_1,p_2)=\frac{1}{2}\left[(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)+Tr(\Sigma_2^{-1}\Sigma_1)+\log\frac{|\Sigma_2|}{|\Sigma_1|}-d\right]. \tag{4}$$

Eq. (3) is a symmetrized version of the KLD in (4), and has an intuitive interpretation that two patches are similar if either of them can be well described by the other. With this dissimilarity the background holes illustrated in Fig. 4 can be reliably identified as similar to the background.



(a) An example image; (b) the two corresponding distributions

Figure 4. An illustration of minimum KL divergence. p_1 and p_2 are two non-adjacent mean-shift patches modeled by multivariate normals. p_2 is a local sample of p_1. $KL(p_1,p_2)$ is large but $KL(p_2,p_1)$ is small. By using the minimum of the 2 values, the two patches will have low dissimilarity and will likely be grouped together.

A region in an image (e.g., the background) is represented as a set of patches, $R=\{p_{r_1},\ldots,p_{r_K}\}$, where $\{r_k\}$ are the indices of the patches forming the region. Using the dissimilarity between patches in (3), we define the dissimilarity between a patch p and region R as the minimum dissimilarity between the patch p and any patch in R,

$$D(p,R)=\min_{r\in R} D(p,r). \tag{5}$$

We define the dissimilarity between two regions R_1 and R_2 as the minimum dissimilarity between their patches,

$$D(R_1,R_2)=\min_{r\in R_1,p\in R_2} D(r,p)=\min_{r\in R_1} D(r,R_2). \tag{6}$$

Note that both background and foreground can be multi-modal. That is, patches in one region (e.g., background) may have very different distributions (e.g., sky and grass). Therefore, for the patch-region dissimilarity, we use the minimum dissimilarity so as to match the patch to the most similar part in the region. Likewise, the minimum dissimilarity measure between two regions implies that they are similar if they have patches in common (e.g., both contain sky). In our context, using alternatives such as median dissimilarity or max-min dissimilarity may not work well due to the regions being multi-modal.

3.5. Soft-label partitions

With the patch distances defined in Section 3.4 we next describe our foreground extraction algorithm. Under the assumption that the user-specified box provides sufficient background statistics, we first initialize the background and foreground priors (Fig. 1b), and then gradually compute a soft-label partition (Fig. 1c). Formally, our objective is to assign each image patch p_i a likelihood (soft-label) of belonging to the foreground category, denoted by $L(p_i)$.

The partitioning process proceeds as follows. First, all patches p_i overlapping with the background mask form the initial background prior B, and are given zero likelihood,

$$L(p_i)=0,\quad \forall p_i\in B. \tag{7}$$

Next, the initial foreground region F_0 is formed using the set of patches that are sufficiently far from B,

$$F_0=\{p_i|D(p_i,B)>D_t\}, \tag{8}$$

where D_t is a foreground threshold whose value will be discussed at the end of the subsection. The foreground likelihood of these initial foreground patches is set to 1,

$$L(p_i)=1, \quad \forall p_i \in F_0 \tag{9}$$

The remaining unlabeled patches are progressively labeled with patches furthest from the background considered first, i.e., in descending order based on their distances from the background prior B, $D(p_i,B)$. Let Θ be the set of currently labeled patches. For each patch p_i under consideration, a local conditional probability with respect to any labeled patch $p_j \in \Theta$ is computed by comparing the distances from p_i to the background prior B and p_j using the softmax (logistic) function,

$$l(p_i \mid p_j)=\frac{e^{-D(p_i,p_j)}}{e^{-D(p_i,p_j)}+e^{-D(p_i,B)}}. \tag{10}$$

Because the feature space represents both color and location, (10) will give high likelihood when the two patches are both visually similar and spatially close together, while also being dissimilar to B. The overall likelihood of patch p_i being foreground is estimated by calculating the maximum likelihood score over all preceding patches,

$$L(p_i)=\max_{p_j\in\Theta} L(p_j)l(p_i \mid p_j). \tag{11}$$

Eq. (11) considers both the conditional probability of the current patch being foreground given the labeled patch, and the probability of the labeled patch also being foreground. Note that these patches are not explicitly assigned a foreground or background label, but instead assigned a likelihood of being foreground, based on foreground likelihood of preceding labeled patches. After all unlabeled patches are processed with (11), a likelihood L is defined for every patch, resulting in a soft-labeling of foreground regions in the image. The procedure is summarized in Algorithm 1.

Algorithm 1. Soft-label partitioning

Input: Image patches $R_I=\{p_1, p_2, ..., p_n\}$, background mask K, threshold D_t.
Output: Foreground likelihood $L(p_i)$ for each patch p_i.
Initialize background prior: $B=\{p_i \mid p_i \cap K \neq \phi\}$.
Initialize foreground: $F_0=\{p_i \mid D(p_i,B) > D_t\}$.
Initial labels: $L(p_i)=0, \ \forall p_i \in B$; $L(p_i)=1, \ \forall p_i \in F_0$.
Initial labeled set: $\Theta = B \cup F_0$.
Repeat
1. Find furthest patch: $p_i = \operatorname{argmax}_{p_i \notin \Theta} D(p_i,B)$
2. Conditional probabilities: $l(p_i \mid p_j)=\frac{e^{-D(p_i,p_j)}}{e^{-D(p_i,p_j)}+e^{-D(p_i,B)}}, \ p_j \in \Theta$
3. Foreground likelihood: $L(p_i)=\max_{p_j\in\Theta} L(p_j)l(p_i \mid p_j)$
4. Update labeled set: $\Theta = \Theta \cup p_i$

Until no more unlabeled patches.

Overall, the above soft-label method can be interpreted as a likelihood-tree growing procedure, as shown in Fig. 5. The initial foreground F_0 is the root of the tree. The likelihood of being foreground is propagated from node to node as the tree grows in a top-down manner.

Finally, the hard-label partition, used in our preceding work [23], can be obtained by replacing the softmax function in (10) with a hard binary-valued function,

$$l_{hard}(p_i \mid p_j)=\begin{cases} 1 & \text{if } D(p_i,p_j) \le D(p_i,B) \\ 0 & \text{if } D(p_i,p_j) > D(p_i,B) \end{cases}. \tag{12}$$

Using (12), p_i will be marked as foreground only if it is more similar to some other foreground patch than the background B. This corresponds to a greedy labeling method, where the foreground set, $F=\{p_i|L(p_i)=1\}$, is updated when a new patch is assigned to the foreground.

We now turn our attention to the threshold D_t that determines the initial foreground region F_0. The choice of threshold is important since it may lead to different tree structures and hence different soft-label partitions (e.g., see Figs. 5d and 5e). Rather than select a single threshold, we instead consider multiple thresholds, i.e., multiple foreground initializations, and produce various candidate soft-label partitions for consideration. In practice, we use all thresholds D_t between the lower and upper bounds, D_l=5 and D_u=50. This interval allows a large enough set of initial foreground priors but excludes unnecessary initializations[3]. Since there are a finite number of possible $D(p_i,B)$ values (one for each image patch), we only need to try a finite number of thresholds. In particular, we sort all values of $D(p_i,B)$ within the interval $[D_l,D_u]$ in ascending order and use the midpoints between two successive values as the set of thresholds. Running the soft-label partitioning method for each threshold, we obtain a large set of soft-label partitions. The size of the set depends on the number of patches in the image. Simple images will have few patches (<5), whereas cluttered images will have more patches (>100), and thus a larger set of soft-label partitions.

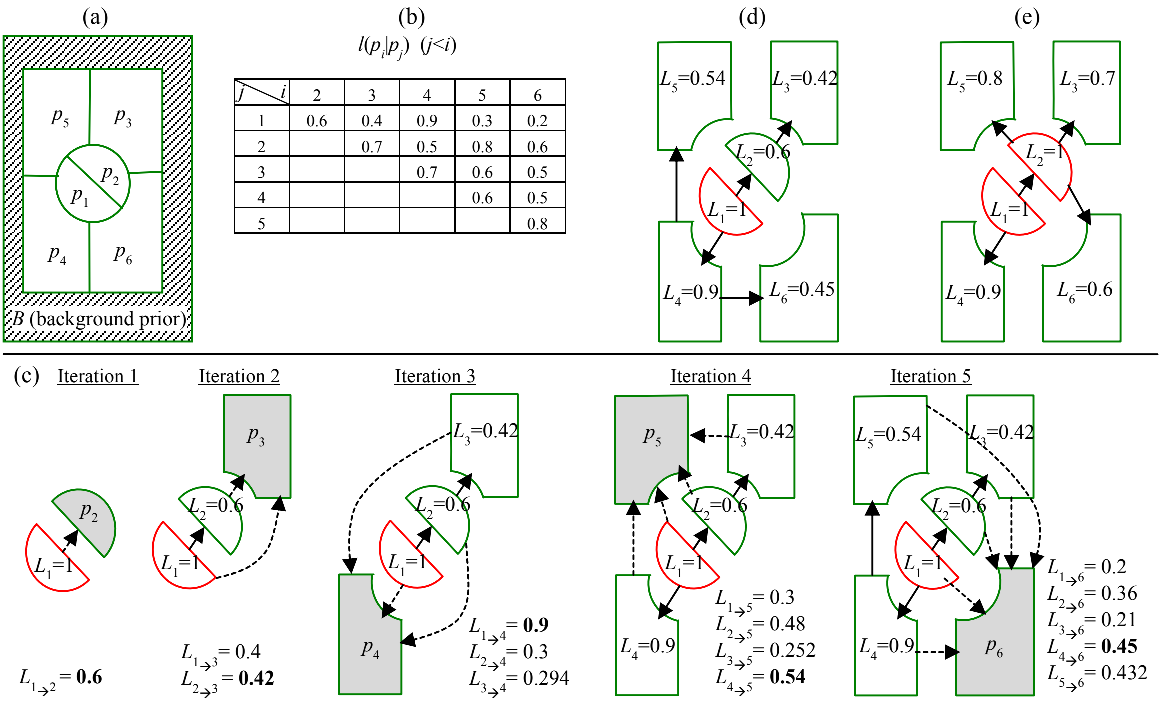
3.6. Foreground probability map

We next build a foreground probability map by fusing all soft-label partitions (Fig. 1d). The fusion is based on the idea of similarity voting. That is, partitions sharing more similarities are given higher influence. Denote F_i as the i-th soft-label partition from the previous stage, and F_i^m as the likelihood value of the m-th pixel in F_i, where pixels take the likelihoods of their corresponding patches. We define the similarity between two soft-label partitions F_i and F_j by

$$d(F_i,F_j)=\left(\sum_{m=1}^{M}\left|F_i^m-F_j^m\right|\right)/\left(\sum_{m=1}^{M}\operatorname{sign}(F_i^m+F_j^m)\right), \tag{13}$$

[3] The KL divergence values calculated in (4) are typically dominated by the Mahalanobis distance term, which follows a 5-dof χ^2 distribution under multivariate Gaussian [35]. For a random sample from B, D_l=5 and D_u=50 set it as foreground with chances p=0.42 and p=10^{-9} respectively.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIP.2015.2389612



j \ i	2	3	4	5	6
1	0.6	0.4	0.9	0.3	0.2
2		0.7	0.5	0.8	0.6
3			0.7	0.6	0.5
4				0.6	0.5
5					0.8

Figure 5. Two examples of tree-structured likelihood propagation as soft-labeling. The demo image in (a) has 7 patches. The background prior is B (striped area). Patches p_1-p_6 are the unlabeled regions (white area), sorted in descending order by their distance to the background prior. (b) Table of conditional probabilities $l(p_i|p_j)$ that p_i is foreground given p_j is foreground. (c) Iterations to generate a tree structure using p_1 as foreground. Foreground likelihood scores are denoted as $L_i = L(p_i)$. White segments and solid arrows form the current tree structure Θ. The gray segment is the current segment under consideration, and dashed lines are candidate connections between the current segment and existing nodes in the tree. In each iteration, the likelihood scores $L_{j\to i}=l(p_i|p_j)L(p_j)$ are calculated for current patch p_i to each existing tree node p_j, and the maximum value (bold) is selected to add the segment to the tree. (d) The tree structure using p_1 as the foreground prior, resulting from iterations in (c). (e) A different tree structure that uses p_1 and p_2 as the foreground prior.

where M is the total number of pixels, and sign(x) is 0 when x=0 and 1 when x>0. When F_i and F_j are hard-label partitions (i.e., sets of foreground pixels), (13) reduces to the scale invariant measure [36],

$$d(F_i,F_j)=|(F_i\oplus F_j)|/|(F_i\cup F_j)|, \tag{14}$$

where ⊕ and ∪ denote symmetric difference and union of two sets, and $|F|$ denotes the cardinality of a set F. We then construct a symmetric affinity matrix A with entries

$$A(i,j)=\exp(-d(F_i,F_j)^2/2\sigma^2), \tag{15}$$

where σ^2 is the variance of the pairwise distances between all partitions $\{F_i\}$ [7, 30]. Finally, a real-valued probability map is calculated as the weighted sum of the soft-label partitions $\{F_i\}$,

$$P=\sum_i w_i^2 F_i\,. \tag{16}$$

The weight vector w is determined using the following constrained optimization problem,

$$\max\ w^T Aw,\ s.t.\ \|w\|^2=1\,. \tag{17}$$

Eq. (17) is a standard Rayleigh quotient problem [37], and the optimal w is given by the top eigenvector of A. Intuitively, the weights found by (17) are higher for partitions sharing more similarities. In short, the probability map is computed as the weighted sum of all soft-label partitions, where larger weights are given to more similar partitions. This corresponds to a similarity voting process leading to a better probability map, compared to [23]. Some example probability maps are given in Figs. 1d and 7a.

3.7. Hypothesis segmentation set

Given the foreground probability map, a set of candidate segmentations is formed by thresholding the probability map P (Fig. 1e). Due to the finite number of patches, the probability map P contains a finite number of values π_i (i=0..n). Therefore, it is easy to create multiple hard-label (binary-valued) candidates from P by brute-force thresholding. In particular, first we sort all values of π_i in ascending order,

$$0=\pi_0<\pi_1<\pi_2<\ldots<\pi_n=1. \tag{18}$$

We have π_0=0 and π_n=1 because there must be some definite foreground and background regions in a valid probability map. We then define a threshold set $T=\{t_i\}_{i=1..n}$ as the midpoints between two successive probability values, $t_i = (\pi_{i-1}+\pi_i)/2$. This threshold set T is used to binarize the probability map P into n hard-label candidates

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIP.2015.2389612

$$C_i = (P > t_i), i = 1 \cdots n. \tag{19}$$

From these hard-label candidates we select promising segmentations according to various evaluation scores, denoted as the *hypothesis set* (Fig. 1f). Taking into account the fact that perceptually meaningful segmentations may correspond to different cost functions, we generate multiple segmentation hypotheses from multiple evaluation scores. In particular, we prefer evaluation scores that encourage different types of segmentations. We consider three score functions from different points of view, which are described below. Other scores could also be used to incorporate any available prior knowledge (like texture or shape).

The *average-cut* (a-cut) score is defined as the average of the distances $D(f,B)$ from each foreground patch f to the background set, i.e. the selected threshold is given by

$$t^{(a)} = \arg\max_{t \in T} \frac{1}{|F(t)|} \sum_{f \in F(t)} D(f, B(t)), \tag{20}$$

where $F(t)$ and $B(t)$ are respectively the foreground and the background groups in the final segmentation map computed from the threshold t. The a-cut score finds a split of foreground and background such that the average distance between the two is large. It is also related to the “average cut” used in spectral partitioning [30], but here we only consider the foreground region when calculating the score.

Inspired by the maximum-margin principles of support vector machines (SVMs), the *maxmin-cut* (m-cut) score maximizes the minimum distance between foreground and background patches,

$$t^{(m)} = \arg\max_{t \in T} D(F(t), B(t)). \tag{21}$$

The m-cut score prefers segmentations where the foreground and background regions have a wide boundary in the feature space, corresponding to the optimization of (6).

The third score is based on the idea of maximally stable extremal regions (MSER) [38], which tries to maximize the local stability of a candidate over the threshold set T. Recent evaluations reveal that the MSER detector [38] exhibits good performance on a variety of benchmarks [39]. The original MSER detector finds regions that are locally stable over a wide range of thresholds. In contrast to previous works, we make a modification by using the full foreground map instead of a local connected region to define MSER, and only considering the global maximum over the whole threshold set T. In our context, the threshold selected by MSER score is defined by

$$t^{(M)} = \arg\min_{t_i \in T} \frac{|F(t_{i-1})| - |F(t_{i+1})|}{|F(t_i)|}, \tag{22}$$

where $\{t_{i\text{-}1}, t_i, t_{i+1}\}$ are three consecutive thresholds in the threshold set T, and $|F|$ denotes the area of the region F. The adoption of the region area as the denominator term in (22) makes the MSER score favor larger foreground regions. Therefore, it is particularly good when a tight mask box is assigned. More importantly, the MSER score is sufficiently robust to allow for the background prior to be updated iteratively, which will be discussed in Sec. 3.8.

Figure 6 plots an example of the three score functions, while varying the threshold t_i. It is worth mentioning that different segmentations may have the same m-cut score. For example, in Figure 6c within the [82, 84] interval the same m-cut score corresponds to 3 different candidates. This means the solutions to m-cut may not be unique. Different m-cut solutions along the optimal interval tend to have slightly different appearances. Hence, we select two hypotheses from the m-cut score function, corresponding to the left and the right ends of the optimal interval. Thus, in total we have four hypothesis binary segmentations, one by MSER, one by a-cut, and two by m-cut. Figure 7 shows an example of building hypotheses from the probability map.

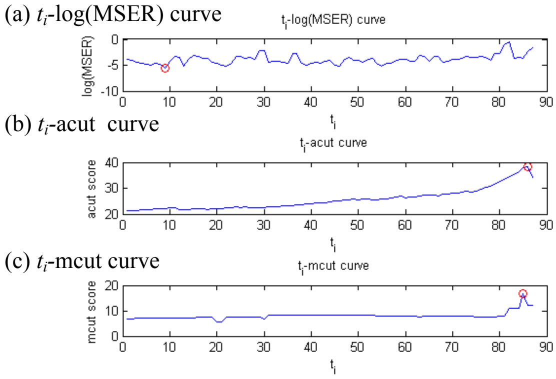


Figure 6. Three t_i-*score* curves for the image of Figure 2. A log-plot is used for the MSER curve to show the minimum clearly. The hypothesis segmentations are selected as the minimum of the log(MSER) curve, and the maxima of the a-cut and m-cut curves. In this example with 87 hard-label candidates, the optimal values are taken at t_9, t_{86}, and t_{85} respectively by the three score functions.

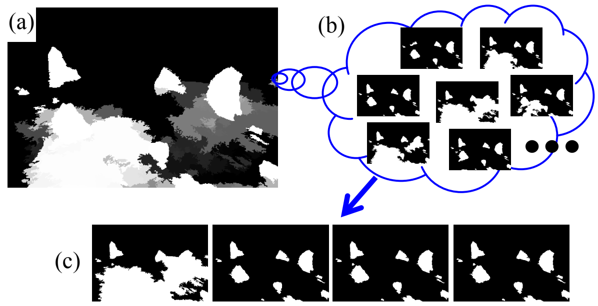


Figure 7. Example probability map and hypothesis set for the image in Figure 2. a) probability map (Sec. 3.6); b) multiple binary candidates by thresholding (Sec. 3.7); c) 4 hypothesis segmentations selected by different evaluation scores, corresponding to the 3 optimal points of Figure 6 (Sec. 3.7).

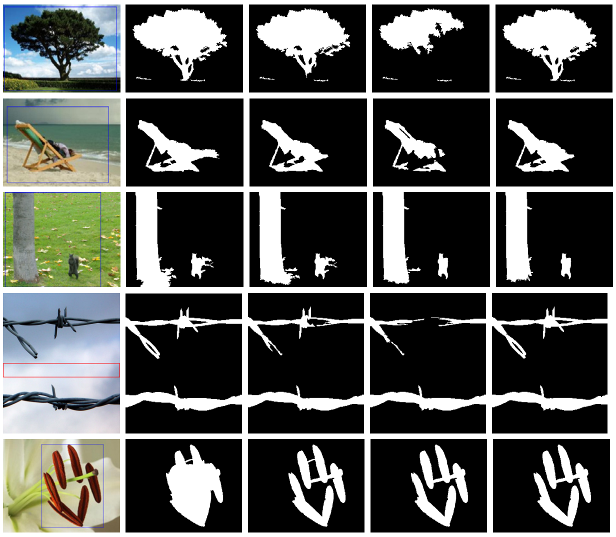
3.8. Iterated background prior propagation

The result of Sec. 3.7 is a set of hypothesis segmenta-

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIP.2015.2389612

tions. The background region of one of these segmentations can be used as the background prior for a new round of segmentation (Algorithm 2), which we call *background prior propagation* (BPP). We use the background from the MSER segmentation for BPP because of its favorable properties mentioned in Sec. 3.7. The process iteratively continues until the background prior stops changing between iterations. The convergence of BPP is guaranteed because image patches can only be added to the background prior in each round. Hence the background prior region can only grow until it reaches a stable point, or, very rarely in our context, covers the full image. The fact that MSER favors bigger foreground regions contributes to the prevention of over-propagation of the background.

Figure 8 shows the background priors after three iterations of BPP for a few example images with complicated foreground topologies (multiple hole, multiply connected, or irregular contours). These examples demonstrate how the background prior gradually propagates into the region of interest and builds multiple hypotheses.



original Iteration 1 Iteration 2 Iteration 3 Final result

Figure 8. Examples of iterative background prior propagation. Columns 2, 3, 4 show the negative background priors after the first three iteration rounds respectively. Column 5 shows the final result by fusing the four hypotheses of the winner round.

Each round of BPP is associated with a hypothesis set. Due to the risk of over-propagation or under-propagation, the best hypothesis set is not necessarily the first or last iterations. In the next subsection, we design a winner selection strategy to automatically select a good hypothesis set. The final results in the last column of Fig. 8 show the effectiveness of our selection method.

3.9. Hypotheses selection and fusion

After convergence, BPP generates several sets of hypothesis binary segmentations, with one set from each BPP iteration round. An automatic mechanism is required to build a final result from the hypotheses sets of all iterations (Figs. 1g and 1h). Direct fusing all these hypotheses is not a good choice due to the risk of including over-propagated backgrounds. Instead, based on the principle of similarity voting, we choose the set with highest intra-similarity for the final fusion. The motivation is that under a good initialization, the results selected by different evaluation scores are generally consistent and correct, whereas under a bad initialization, the results selected by different evaluation scores are generally inconsistent and unreliable. We denote the hypothesis set of 4 binary segmentations in the jth iteration of BPP as $H^j=\{H^j_{MSER}, H^j_{acut}, H^j_{mcut1}, H^j_{mcut2}\}$. For each hypothesis set H^j, we calculate the mean pairwise similarity within the set,

$$s(H^j)=\frac{1}{12}\sum_{\substack{a,b\in\{MSER,acut,mcut1,mcut2\}\\ a\neq b}} s(H^j_a,H^j_b) \tag{23}$$

where the similarity between two binary segmentations H_1 and H_2 is defined as the Jaccard index [16],

$$s(H_1,H_2)=|H_1\cap H_2|/|H_1\cup H_2|. \tag{24}$$

The set with the largest mean similarity $s(H^j)$ is selected as the winner set H. Finally, from the 4 binary-valued hypothesis maps $H=\{H_{MSER}, H_{acut}, H_{mcut1}, H_{mcut2}\}$ of the winner set, we compute the final foreground map F by a simple pixel-wise majority vote,

$$F=((H_{MSER}+H_{acut}+H_{mcut1}+H_{mcut2})\geq 2). \tag{25}$$

Algorithm 2. Enhanced figure-ground classification with background prior propagation (EFG-BPP)

Input: A target image I and a background mask.
Output: foreground segmentation F.
Initialization: Set initial background prior B_0 as the set of image patches overlapping the background mask. Set j=0.
Calculate image patches using adaptive mean shift (Sec 3.3).
Repeat
1. Generate soft-label partitions from B_j (Alg. 1, Sec. 3.5).
2. Compute a real-valued probability map P_j (Sec. 3.6)
3. Generate hard-label candidates from P_j, and obtain a hypothesis set H^j using different score functions (Sec 3.7).
4. Use the MSER segmentation H^j_{MSER} to define a new background prior B_{j+1} (Sec. 3.8)
5. $j = j+1$;

Until ($B_j == B_{j-1}$)
Select the winner hypothesis set from $\{H_j\}$ based on intra-set similarity (Sec. 3.9): $H=\{H_{MSER}, H_{acut}, H_{mcut1}, H_{mcut2}\}$.
Calculate the final segmentation via majority vote (Sec 3.9):
$F=((H_{MSER}+H_{acut}+H_{mcut1}+H_{mcut2})\geq 2)$.

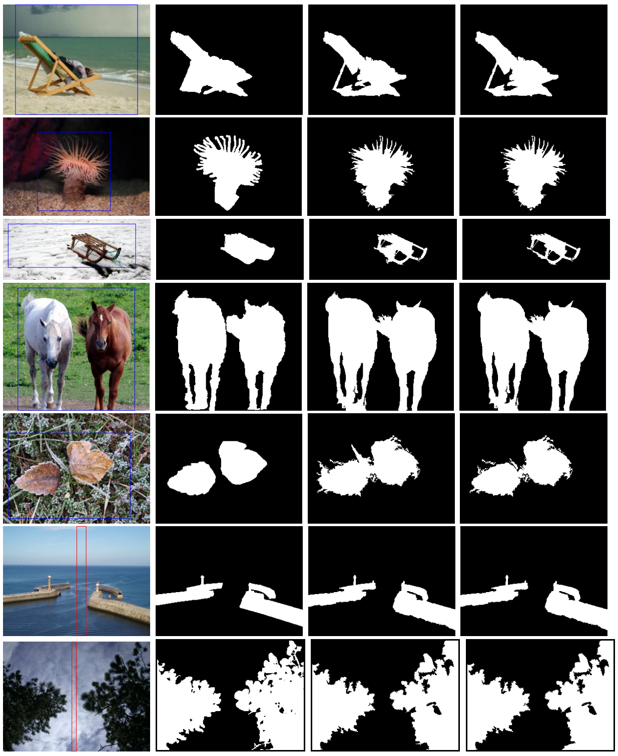
The full framework is summarized in Algorithm 2. Note that most parameters in our system are set automatically based on the image, and our multiple hypotheses framework is based on generating segmentation candidates using all possible thresholds. In addition, because segmentation is based on soft-labeling and multiple hypothesis segmentations are kept, the effects of erroneous outputs in each stage of the pipeline are minimized.

4. Experiments

In this section, we evaluate our algorithm. Experiments are run on a notebook computer with an Intel core-i7 CPU 2.7Ghz processor and 4GB RAM. Our algorithm is implemented in MATLAB and is available online[4].

4.1. Evaluation of segmentation results

We make a comprehensive comparison using four image datasets with ground truths; Weizmann 1-obj (100 images), Weizmann 2-obj (100 images) [40], IVRG [41] (1000 images), and grabcut [42] (50 images). We denote our enhanced figure-ground classification using soft-label partitions and background prior propagation as EFG-BPP. We also test the performance using hard-label partitions with (12), which is denoted as EFG-BPP (hard-label). We also compare against grabcut [7] and other methods [4,11,28,43]. The initial mask box is assigned by the user and is fixed for comparisons between box-based methods.



original & mask ground truth EFG-BPP best candidate

Figure 9. Weizmann examples. Rows 1-3 are 1-obj examples. Rows 4-8 are 2-obj examples. The EFG-BPP result is equally good as the user selection for rows 1, 2, 4, 6, & 7; and slightly worse on the remaining rows. The outside of the blue boxes or the inside of the red boxes define the background masks.

[4] http://www.graphics.pku.edu.cn/members/chenyisong/projects/FigureGroundPuzzle/ FGpuzzle.htm

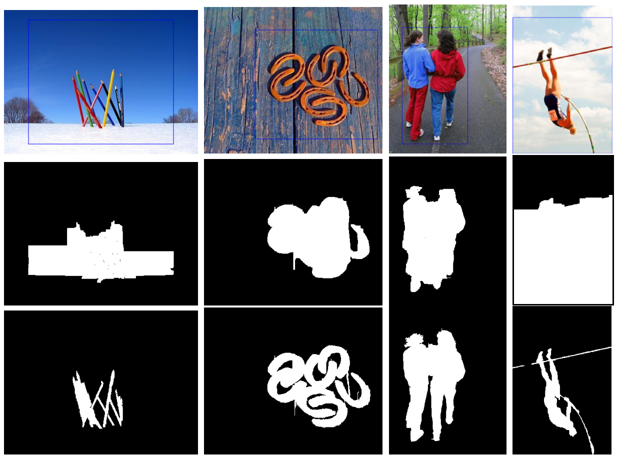


Figure 10. IVRG examples. Top: image & mask; middle: grabcut results; bottom: EFG-BPP results.

We first qualitatively examine the major benefits of our segmentation method on some examples. Figure 9 displays some example segmentations from the Weizmann dataset. EFG-BPP successfully labels background holes and multiply connected components, and identifies many details missed in the manual-made truths. Figure 10 shows some example segmentations from the IVRG dataset. In comparison to grabcut [7], EFG-BPP exhibits qualitatively better performance, mainly when segmenting complicated foreground and background shapes.

The performance on each image is evaluated using F-measure, $F=2PR/(P+R)$, where P and R are the precision and recall values [43]. Table 1 reports the 95% confidence intervals of the average F-scores of both hard-label and soft-label EFG-BPP. We also give the output of the first and the last iteration of BPP for both schemes. We note that even in the absence of background prior propagation, the result of the first iteration is sufficiently good. By employing background prior propagation and automatic hypotheses selection the result becomes better. It is worth noting that the performance of the last iteration slightly degrades for the Weizmann and IVRG datasets. This indicates that the best result is not necessarily reached at the time of convergence, but may instead come in some earlier iteration round. Intelligently selecting and leveraging multiple hypotheses improves the F-measure on *all* the datasets. This validates the principle of similarity voting. That is, *we should encourage more candidates to participate in a multiple hypothesis scheme, and similarity comparison plays an important role in smart hypothesis selection.* Finally, the foreground map closest to the ground truth in all hard-label candidates (last column of Fig. 9) forms an upper bound for the figure-ground classification method. The EFG-BPP result performs close to this upper bound.

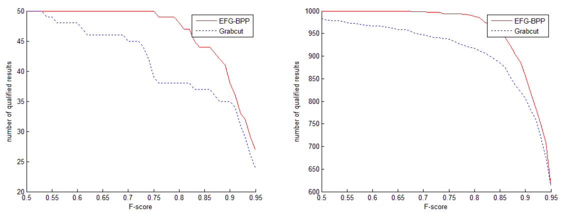
Table 1. F-measures on four image datasets. Bold indicates best accuracy among all methods, excluding user-select.

	Weizmann 1-obj	Weizmann 2-obj	IVRG images	Grabcut images
EFG-BPP (hard-label)	0.93 ± 0.010 (4.77s)	0.89 ± 0.019 (2.12s)	0.94 ± 0.005 (4.02s)	0.93 ± 0.018 (12.09s)
first iteration	0.93 ± 0.011 (3.21s)	0.88 ± 0.021 (1.91s)	0.94 ± 0.004 (3.74s)	0.91 ± 0.027 (10.38s)
last iteration	0.92 ± 0.014	0.88 ± 0.021	0.93 ± 0.006	0.92 ± 0.021
EFG-BPP	**0.94 ± 0.010 (4.82s)**	**0.90 ± 0.017 (3.09s)**	**0.95 ± 0.003 (5.80s)**	**0.93 ± 0.017 (25.99s)**
first iteration	0.93 ± 0.011 (3.28s)	0.89 ± 0.017 (2.34s)	0.94 ± 0.003 (4.90s)	0.92 ± 0.023 (15.90s)
last iteration	0.92 ± 0.014	0.88 ± 0.022	0.93 ± 0.006	0.92 ± 0.021
user-select (upper bound)	0.95 ± 0.009	0.91 ± 0.015	0.96 ± 0.002	0.95 ± 0.013
Nearest competitors	0.85 ± 0.035 [7] (5.67s)	0.81 ± 0.044 [7] (3.95s)	0.93 ± 0.006 [7] (4.96s)	0.89 ± 0.035 [7] (12.95s)
	0.93 ± 0.009 [11]	0.68 ± 0.053 [43]		
	0.87 ± 0.010 [4]	0.66 ± 0.066 [28]		

The last row of Table 1 shows the results of several reference algorithms [4,7,11,28,43]. EFG-BPP performs slightly better than the state-of-the-art techniques for single connected foregrounds (Weizmann 1-obj), and outperforms the state-of-the-art on multiple connected foregrounds (Weizmann 2-obj). For the grabcut image set, we also compare the error rate with the result reported in [13]. The error rate is defined as the percentage of mislabeled pixels within the initial box mask. Table 2 shows that EFG-BPP has lower average error rate than grabcut [7] and iterated distribution matching [13]. Figure 11 compares the performance of grabcut and EFG-BPP for the Grabcut and IVRG image sets. The x-axis is the F-score threshold, and the y-axis is the number of images with an F-score greater than this threshold. The figure shows that EFG-BPP generates fewer poor segmentations.

Table 2. Average error rate comparison on the grabcut dataset

EFG-BPP	EFG-BPP(hard-label)	grabcut[7]	distribution matching[13]
4.3%	5.2%	8.1%	7.1%



(a) Grabcut dataset (b) IVRG dataset

Figure 11. Comparison between EFG-BPP and grabcut.

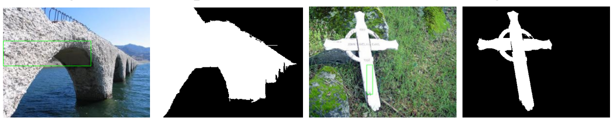


Figure 12. Figure-ground switching examples.
Left: original & mask; right: EFG-BPP result

For some scenes the contour separating foreground and background may take a very complicated shape. This makes it difficult to assign a box mask providing sufficient information about the background. To handle these images we switch the roles of foreground and background. Namely, at the initialization stage we take the foreground region as the background, and assign a bounding box fully enclosed by it (the green boxes in Fig. 12). After the segmentation we reverse the foreground and the background to obtain the final result. Fig. 12 shows two images that can be improved by this switching operation.

We also evaluate EFG-BPP on the Berkeley segmentation dataset [44]. Figure 13 gives the results of some challenging images in the Berkeley dataset (rows 1 & 2) and the grabcut dataset (row 3). The adaptive bandwidth parameters $\{h_s, h_r\}$ computed by (2) are also given. Note that the adaptive bandwidths can vary a lot for different scenes, and our adaptive initialization works well in finding suitable bandwidths, and generates good mean-shift patches. The images in the figure show that EFG-BPP can handle challenging background or foreground topologies. As a typical example, almost all connected components and all holes in image 370036 are successfully identified.

4.2. Evaluation of soft- and hard-label schemes

The soft-label scheme performs better than the hard-label scheme at the cost of slightly slower running time (see Table 1). For simple scenes the outputs of the two schemes are often the same or have only minor differences. For cluttered scenes the two schemes are more likely to output different results. The soft-label scheme has better chance of keeping fine details, due to the soft likelihoods that are transferred to the probability map. Some different outputs are given in Figure 14 for comparison.

image hard-label soft-label image hard-label soft-label

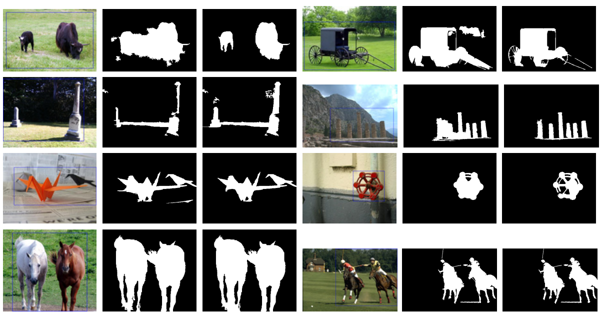


Figure 14. Example results comparing hard-label and soft-label schemes. The soft-label result outperforms hard-label in the first 3 rows, but performs slightly worse for the last row.

Figure 13. Some foreground extraction results for Berkeley and Grabcut datasets.

4.3. Evaluation of background prior propagation

Overall, about 80% of the time, EFG-BPP converges within three iterations of BPP, and most of them select the first round as the winner. This suggests that our method still can perform well without BPP. Nevertheless, we have seen in Table 1 that the output of the first loop is not necessarily the best one. For some cluttered scenes, it may take as long as 10 iteration loops to propagate the background prior deeply into the region of interest. The performance gain of background prior propagation mainly comes from these long iterations. Figure 15 displays some example images that take more than 5 iterations before reaching convergence. Although over-propagation occurs in some trials (row 4 of Fig. 15, and rows 1, 2, & 4 of Fig. 8), similarity voting is able to make a good selection from all iteration rounds and output a satisfactory result.

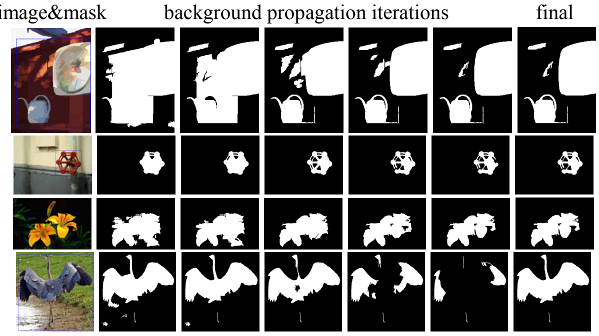


Figure 15. Some examples of long iterations of BPP.

Table 3 compares the three selection strategies from the BPP results: similarity voting (auto), first iteration, and last iteration. Note that multiple strategies can produce the best segmentation at the same time. The three strategies are consistent to a large extent, with each obtaining good results in at least 70% of the trials.

Table 3. The number of times each selection strategy obtained the best segmentation. The value in () is the total number of images in the images set.

	Weiz1(100)	Weiz2(100)	IVRG(1000)	Grabcut(50)	Sum
Auto	**81**	**88**	**811**	**41**	**1021**
First	79	77	725	36	917
Last	74	82	744	38	938

An interesting observation is that the last iteration outputs better results more times than the first iteration whereas the *F*-score is inferior as shown in Table 1. This is due to the higher risk of over-propagation in the last-iteration. Even though over-propagation occurs only in a small number of trials it can cause significant drop of the *F*-score. The auto-choice (EFG-BPP) consistently outperforms the two competitors in both *F*-scores and number of best segmentations. *This provides strong evidence for the power of similarity voting as a winner selection criterion.* Some examples of the three schemes are given in Fig. 16.

4.4. Evaluation of initial mask box

The background prior propagation mechanism makes EFG-BPP tolerant to loose initial mask boxes around the foreground subject, since for each iteration the background region can move further into the initial box. We test 500 IVRG images that allow looser bounding boxes while keeping parts of the background prior. For each image, we manually assign the maximally allowed range of the 4 edges of the mask box such that some common parts of the

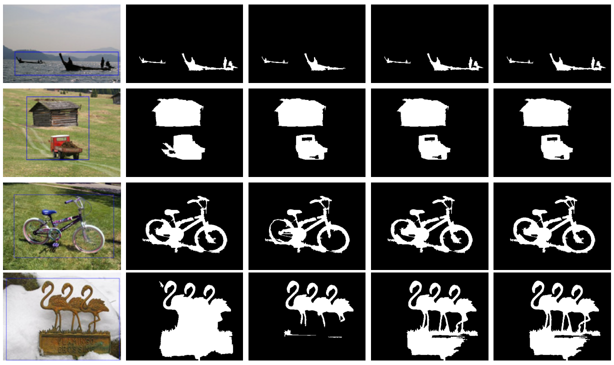


Figure 16. Segmentation results of different iterations of background prior propagation. The result of the first iteration may be better (rows 1 & 4), worse (row 2), or comparable (row 3) to that of the last iteration. EFG-BPP makes a good balance and best candidate gives an upper bound.

background remain, and test various box sizes within the range. Table 4 shows that EFG-BPP is stable and insensitive to looser boxes, and outperforms both the first and last iterations of BPP. Fig. 17 shows example segmentation results by various mask box sizes. Fig. 18 shows that different mask boxes can be used to successfully extract different foreground elements.

Table 4. Results of EFG-BPP for various mask box sizes on 500 IVRG images. Each column shows the average F-measure when randomly expanding the box edges within a range of allowed values (as a percentage of the maximum allowed value).

	0%(Tight)	0-33%	33-67%	67-100%(Loosest)
EFG-BPP	0.94 ± 0.004	0.94 ± 0.005	0.94 ± 0.005	0.94 ± 0.005
First iter.	0.94 ± 0.004	0.93 ± 0.005	0.93 ± 0.006	0.93 ± 0.006
Last iter.	0.93 ± 0.008	0.92 ± 0.008	0.93 ± 0.008	0.93 ± 0.007

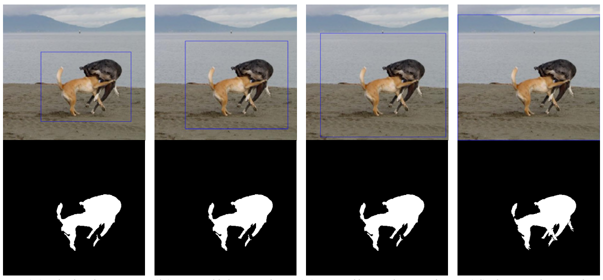


Figure 17. Examples of varying box sizes and the corresponding segmentations.

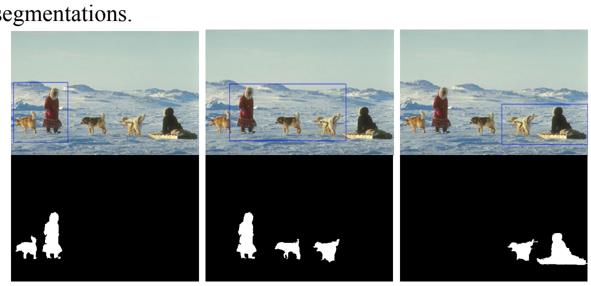


Figure 18. Example segmentations from different mask boxes

4.5. Failure cases

Figure 19 shows two failure cases of EFG-BPP. In general, the method fails if the background prior does not match true background well. This can be caused by similar foreground and background appearances (Fig. 19a), or too cluttered background which prevents successful background prior propagation (Fig. 19b). These can be improved by employing a more flexible initial mask.

Figure 19. Example failure cases of EFG-BPP.

5. Conclusion

We have proposed an enhanced figure-ground classification algorithm. Our framework is based on the principles of generating multiple candidate segmentations, selecting the most promising using several scoring functions, and then fusing them with similarity voting. Specifically, an adaptive mean-shift algorithm is used to generate image patches, and soft-segmentations are produced using tree-structured likelihood propagation. We put forward the idea of similarity voting to guide the generation of multiple foreground map hypotheses, and use several score functions to select the most promising ones. To improve robustness we iteratively propagate the background prior and generate multiple hypothesis sets. The most promising hypothesis set is automatically determined by similarity voting, and the corresponding hypotheses are fused to yield the final foreground map. Our method produces state-of-the-art results on challenging datasets, and is able to segment the fine details in the segmentation, as well as background holes and multiply-connected foreground components.

Future work includes more intelligent schemes with multiple background prior hypotheses, as well as extensions to box-based segmentation in video. Finally, our segmentation algorithm could be applied to other computer vision tasks like tracking, recognition and retrieval.

Acknowledgement

The authors would like to thank Prof. Kenichi Kanatani for beneficial discussions, and the reviewers for helpful comments.

References

[1] Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: Unifying segmentation, detection, and recognition, IJCV, 63(2):113–140, 2005.

[2] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In Proc. ICCV2001, volume 1, pp. 105-112.

[3] I. Ayed, H. Chen, K. Punithakumar, I. Ross, and S. Li, Graph cut segmentation with a global constraint: Recovering region distribution via a bound of the Bhattacharyya measure, in Proc. CVPR, 2010, pp. 3288–3295.

[4] S. Bagon, O. Boiman, and M. Irani, What is a good image segment? a unified approach to segment extraction. In ECCV, pages 30–44, 2008.

[5] Y. Liu and Y. Yu, Interactive Image Segmentation Based on Level Sets of Probabilities, IEEE Transactions on Visualization and Computer Graphics, 18(2), pp.202-213,2012.

[6] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, A comparative study of energy minimization methods for markov random fields with smoothness-based priors. PAMI, 30(6):1068–1080, 2008.

[7] C. Rother, V. Kolmogorov, and A. Blake, "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3):309–314, 2004.

[8] B. Micusik and A. Hanbury. Automatic image segmentation by positioning a seed. ECCV2006, Vol. 2, 468–480.

[9] L. Grady, M. Jolly, A. Seitz, Segmentation from a box, ICCV2011, pp. 367-374.

[10] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman, Using multiple segmentations to discover objects and their extent in image collections. In CVPR2006, 1605–1614.

[11] J. Carreira and C. Sminchisescu. Constrained parametric min cuts for automatic object segmentation. CVPR2010, pp. 3241-3248.

[12] M. Mignotte, A Label Field Fusion Bayesian Model and Its Penalized Maximum Rand Estimator for Image Segmentation, IEEE Trans. IP, 19(6), 2012, 1610-1624.

[13] V. Pham, K. Takahashi, T. Naemura, Foreground- Background Segmentation using Iterated Distribution Matching, CVPR2011, pp. 2113-2120.

[14] Z. Kuang, D. Schnieders, H. Zhou, K. Wong, Y. Yu, Bo Peng, Learning Image-Specific Parameters for Interactive Segmentation, cvpr2012.

[15] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, Image segmentation with a bounding box prior. ICCV2009, pp. 277-284.

[16] D. Kuettel, V. Ferrari, Figure-ground segmentation by transferring window masks, cvpr2012.

[17] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. In ICCV2009, pp. 817 - 824.

[18] M. Narayana, Background modeling using adaptive pixelwise kernel variances in a hybrid feature space, CVPR2012.

[19] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In ICCV, pages 39–846,1998.

[20] K.-J. Yoon and I.-S. Kweon. Adaptive supportweight approach for correspondence search. PAMI, 28(4):650–656.

[21] A. Criminisi, T. Sharp, and A. Blake. GeoS: Geodesic image segmentation. In ECCV, pages 99–112, 2008.

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

[23] Y. Chen, A. Chan, G. Wang, Adaptive Figure-ground Classification, cvpr2012, pp. 654-661.

[24] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. Proc. CVPR workshop perceptual organization in computer vision, 2004.

[25] K. Hsu, Y. Lin, Y. Yu, Augmented Multiple Instance Regression for Inferring Object Contours in Bounding Boxes, IEEE Trans. IP, 23(4), 2014, 1722-1736.

[26] Z. Li, X. Wu, S. Chang, Segmentation Using Superpixels: A Bipartite Graph Partitioning Approach, cvpr2012.

[27] J. Kittler, M. Hatef, R. Duin, and J. Matas, On combining classifiers, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 3, pp. 226–239, 1998.

[28] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. PAMI, 24(5):603–619, 2002.

[29] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. IJCV, 59(2):167–181, 2004.

[30] J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI, 22(8), pp. 888-905, 2000.

[31] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma. Natural image segmentation with adaptive texture and boundary encoding, ACCV2009, pp. 135-146.

[32] T. Malisiewicz and A. Efros. Improving spatial support for objects via multiple segmentations. In BMVC, 2007.

[33] D. Comaniciu, An algorithm for data-driven bandwidth selection, PAMI, 25(2), 2003, pp. 281-288.

[34] T. Cover and J. Thomas, Elements of information theory. Wiley Series in Telecommunications, John Wiley and Sons, New-York, USA, 1991.

[35] K. Kanatani, Statistical Optimization for Geometric Computation: Theory and Practice, Elsevier 1996.

[36] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm, ICCV2007, pp.1-8.

[37] R. A. Horn and C. A. Johnson, Matrix Analysis, Cambridge University Press, 1985.

[38] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In Proc. of British Machine Vision Conf., 384–393, 2002.

[39] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. Int. Journal of Comp. Vision, 65(1-2):43–72, 2005.

[40] http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/scores.html, Weizmann dataset webpage.

[41] http://ivrg.epfl.ch/supplementary_material/RK_CVPR09/index.html, ivrg dataset webpage.

[42] http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm.

[43] R. B. S. Alpert, M. Galun and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration, CVPR2007, pp. 1–8.

[44] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/,Berkeley segmentation dataset page.

Yisong Chen received the B.S. degrees in information engineering from Xi'an Jiaotong University in 1996, and the Ph.D. degree in computer science from Nanjing University in 2001. From 2001 to 2003, he was a Postdoctoral researcher with the HCI & Multimedia Laboratory in Peking University. From 2003 to 2005, he was a research fellow in the Image computing group, City University of Hong Kong. From 2005 to 2006, he was a senior research fellow with the HEUDIASYC laboratory of CNRS, in the University of Technology, Compiegne, France. In 2007, he joined the Department of Computer Science, Peking University, as an Associate Professor. His research interests include computer vision, image computing, and pattern recognition.

Antoni B. Chan received the B.S. and M.Eng. degrees in electrical engineering from Cornell University, Ithaca, NY, in 2000 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, San Diego (UCSD), San Diego, in 2008. From 2001 to 2003, he was a Visiting Scientist with the Vision and Image Analysis Laboratory, Cornell University, Ithaca, NY, and in 2009, he was a Postdoctoral Researcher with the Statistical Visual Computing Laboratory, UCSD. In 2009, he joined the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, as an Assistant Professor. His research interests include computer vision, machine learning, pattern recognition, and music analysis. Dr. Chan was the recipient of an NSF IGERT Fellowship from 2006 to 2008, and an Early Career Award in 2012 from the Research Grants Council of the Hong Kong SAR, China.