Adaptive Figure-Ground Classification

Yisong Chen[1] Antoni B. Chan[2]

[1]Graphics Laboratory, EECS Department
Key Laboratory of Machine Perception
(MOE), Peking University
chenys@graphics.pku.edu.cn

[2]Video, Image, and Sound Analysis Lab (VISAL)
Department of Computer Science
City University of Hong Kong
abchan@cityu.edu.hk

Abstract

We propose an adaptive figure-ground classification algorithm to automatically extract a foreground region using a user-provided bounding-box. The image is first over-segmented with an adaptive mean-shift algorithm, from which background and foreground priors are estimated. The remaining patches are iteratively assigned based on their distances to the priors, with the foreground prior being updated online. A large set of candidate segmentations are obtained by changing the initial foreground prior. The best candidate is determined by a score function that evaluates the segmentation quality. Rather than using a single distance function or score function, we generate multiple hypothesis segmentations from different combinations of distance measures and score functions. The final segmentation is then automatically obtained with a voting or weighted combination scheme from the multiple hypotheses. 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. In addition, this improvement in accuracy is achieved with low computational cost.

1. Introduction

Foreground extraction in still images plays a key role in vision applications [1]. Popular approaches include interactive graph cut [2], random walk [3], geodesic [4], information theory [5], and variational solutions [6]. On the one hand, we are looking for better interactive approaches that provide a priori knowledge to guide segmentation. Bounding box assignment and seed positioning are two representative methods [7,8]. On the other hand, we desire simple models that free users from troublesome algorithm design. The uncertainty of model selection and goodness evaluation makes robust segmentation difficult [9,10]. Different models lead to different results and there exists no dominant winner [11]. Recent attempts report encouraging results through the aid of reference distributions or multiple hypotheses [12,13], although widely applicable solution in the absence of a priori knowledge remains a big challenge.

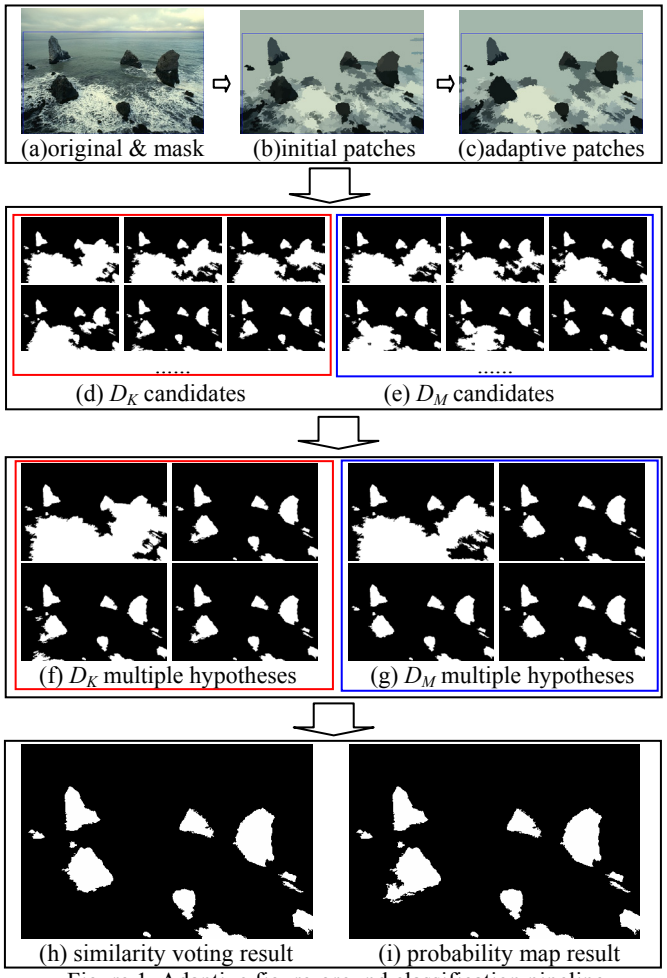


Figure 1. Adaptive figure-ground classification pipeline
1st row block: Adaptive mean-shift patches generation;
2nd row block: Multiple candidates from multiple initializations;
3rd row block: Multiple hypotheses by eight evaluation scores;
4th row block: Two Automatic selection results.

The possibility of foreground extraction from a box input was explored in recent work [7,12,14,15]. However, these methods suffer from restrictive assumption about latent distributions [12], inability to treat complicated scene topologies [14], or inefficient similarity measure [15]. In this paper, we propose a box-based foreground extraction method that gives promising solutions in a broadly appli-

cable environment. The pipeline of our framework is illustrated in Figure 1. Under the assumption that a mask box can provide good statistical information about the background, we design an online figure-ground classification algorithm. Two novel probability distances are defined to measure the similarities between adaptive mean-shift patches in a 5D joint color-spatial feature space. Multiple hypotheses are generated from various initial foreground maps and different score functions. The final segmentation is obtained by a voting scheme or a weighted combination of the multiple hypotheses. This method is not sensitive to difficult scene topology and reliably treats multi-connected, multi-hole foregrounds.

The remainder of this paper is organized as follows. Section 2 presents our figure-ground classification framework. Section 3 gives experimental results on several image datasets. Section 4 concludes the paper.

2. A figure-ground classification framework

Our segmentation algorithm is based on a user-specified mask box that defines the background prior, as in previous approaches [7, 16]. Either side of the box can be defined as the background mask, which contains only background pixels. The complement of the background mask makes the foreground mask, which may contain both foreground and background elements. The mask box can flexibly handle different cases of partially-inside or multiply connected foreground. An example mask box is shown in Figure 1a, with more examples in Figures 5-8.

2.1. Algorithm overview

An overview of our foreground extraction algorithm appears in Figure 1. Under the assumption that a mask box is able to provide sufficient statistical information about the background, we treat the segmentation process as an online figure-ground (f-g) classification task. We first generate an over-segmented image using an adaptive mean-shift algorithm, which partitions the original image, I, into a set of non-overlapping patches $R=\{p_1, p_2, ..., p_n\}$(Figures 1b and 1c). Given this over-segmentation, our objective is to group the patches into a foreground category F and a background category B. That is, for every patch p_i we perform a binary classification so that

$$L(p_i)=\begin{cases}1 & \text{if } p_i \in F \\ 0 & \text{if } p_i \in B\end{cases} . \quad (1)$$

To do this, each patch is modeled as a multivariate normal distribution in a 5D joint color-spatial feature space. Next, patches overlapped with the background mask are used to form the background prior. An initial foreground prior is obtained by selecting patches that are the most dissimilar to the background prior. The remaining patches are iteratively assigned based on their proximity to the foreground or background prior, with the foreground prior being updated with new patches. A large set of candidate segmentations is formed by initializing with different foreground priors, and using different distance functions (Figures 1d and 1e). From this large set of candidates, multiple hypotheses are selected based on several evaluation scores that encourage different types of segmentations (Figures 1f and 1g). The final segmentation is then formed by combining the multiple hypotheses, using either a similarity voting or a weighted sum scheme (Figures 1h and 1i).

2.2. Patch making by adaptive mean-shift

Defining the segmentation as the grouping of non-overlapping regions has become popular due to its advantages in information transfer and computational efficiency [17,18,19]. We choose the mean-shift algorithm [17] to do over-segmentation since mean-shift patches are better described statistically in comparison to other super-pixel generators [20]. To remain consistent with the underlying probabilistic framework of the mean-shift algorithm, we model each mean-shift patch as a multivariate normal distribution. A feature vector in the 5D joint color-spatial feature space is given by

$$f=(L,a,b,x,y), \quad (2)$$

where (x,y) are the 2D pixel coordinates and (L,a,b) are the pixel values in the Lab color space. We use the Lab color space because it is better modeled by a normal distribution in comparison to RGB [21]. Finally, we treat the 3D color features and the 2D spatial features identically and do not give any explicit priority to spatially adjacent patches.

The mean-shift result relies heavily on the two bandwidth parameters, h_s and h_r, corresponding to the 2 spatial coordinates and the 3 color features. Different initial settings lead to different super-pixel sets and only some of them are suitable for the subsequent classification [22]. This is illustrated in the first row of Figure 1, where the default setting of h_s=7 and h_r=6 generates cluttered patches and fails to transfer the background prior into the region of interest, whereas a bandwidth setting h_s=10 and h_r=8.6 (determined by our adaptive scheme) generates much sparser patches and leads to good foreground extraction.

Based on the relationship between the bandwidth parameters and the covariance matrix of the multivariate normal distribution [23], 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 5x5 covariance matrix Σ_i for each patch p_i is calculated,

$$\Sigma_i=\begin{bmatrix}\Sigma_i^{(rr)} & \Sigma_i^{(rs)} \\ \Sigma_i^{(sr)} & \Sigma_i^{(ss)}\end{bmatrix}. \quad (3)$$

The 3x3 submatrix $\Sigma_i^{(rr)}$ is the covariance matrix in the (L,a,b) subspace, the 2x2 submatrix $\Sigma_i^{(ss)}$ is the covariance in the (x,y) subspace, and the 3x2 submatrix $\Sigma_i^{(rs)}$ is the co-

variance between color and location. Finally, the adaptive bandwidths are estimated by averaging the color/spatial variances over all collected patches in F_0,

$$h_s = \left\lfloor \sqrt{\frac{1}{|F_0|}\sum_{i\in F_0}\frac{1}{2}\mathrm{trace}(\Sigma_i^{(ss)})} \right\rfloor, h_r = \sqrt{\frac{1}{|F_0|}\sum_{i\in F_0}\max(\mathrm{diag}(\Sigma_i^{(rr)}))}\,. \quad (4)$$

Whereas h_s is estimated from the variance in both x- and y-directions, h_r is estimated by averaging the Lab components with largest variance, due to the observation that this component is often more dominant in the Lab space. By using the statistics from the foreground mask, our approach tunes the bandwidth parameters to form better representative patches in the foreground mask region. A similar bandwidth estimation technique is adopted in [14].

(a) (b) (c) (d) (e)

Figure 2. Treating cluttered textures with big bandwidths: (a) original image; (b) mean-shift patches under small bandwidths, and (c) the resulting segmentation; (d) mean-shift patches under big bandwidths, and (e) the resulting segmentation.

In some cases, when the background contains repetitive cluttered textures, the adaptive mean-shift may produce too many image patches, and cause the background patches to be mainly distributed along the mask boundary (as in Figure 2). This will lead to a poor estimate of the background prior and a poor segmentation (typically when there are multiple foreground objects). 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 initial bandwidths h_s=14,h_r=12 to create fewer image patches. By employing larger bandwidths we merge small patches into big ones and extend the background prior deeper into the mask region.

2.3. Similarity measure between patches

In the next stage of the segmentation pipeline, patches are gradually assigned to the current background or foreground regions, based on their distances to each region. We will represent a region as the set of its patches. Hence, we must first define a suitable distance measure between two patches, and between a patch and a region, i.e., a set of patches.

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 (2). The mean vector μ_i and the covariance matrix Σ_i are estimated using patch statistics. All patches are eroded with a radius-1 disk structuring element to avoid border effects.

In this paper, we consider two distance measures. The first is the minimum KL-divergence between two patches,

$$D_K(N_1,N_2) = \min(KL(N_1,N_2), KL(N_2,N_1)) \quad (5)$$

where N_1 and N_2 are two Gaussians, $N(\mu_1,\Sigma_1)$ and $N(\mu_2,\Sigma_2)$, and the KL-divergence between two d-dimentional Gaussians is [24]

$$KL(N_1,N_2) = \frac{1}{2}\left[(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2) + Tr(\Sigma_2^{-1}\Sigma_1) + \log(\frac{|\Sigma_2|}{|\Sigma_1|}) - d\right]. \quad (6)$$

(5) is a symmetrized version of the KL divergence (6), and has an intuitive interpretation that the two patches are similar (i.e., should be grouped together) if either of them can be well described by the other.

The second distance measure that we consider is the minimum Mahalanobis distance,

$$D_M(N_1,N_2) = \min((\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2), (\mu_2-\mu_1)^T\Sigma_1^{-1}(\mu_2-\mu_1)) \quad (7)$$

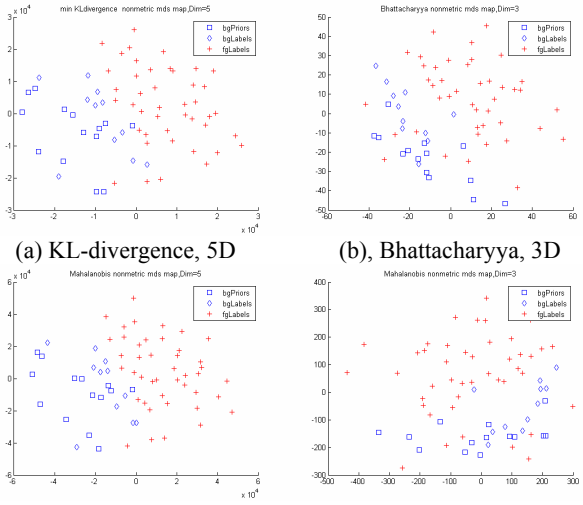
which is a variation of (5) that only considers the distance between the means of the patches. In general, there is no guarantee that one of D_M and D_K is better than the other, but they indeed provide beneficial complements to each other. Therefore, in our framework we will use both D_M and D_K measures and output multiple hypotheses to enhance the chance of obtaining a good segmentation.

Given a similarity metric D (either D_M or D_K), we define the distance from a single patch p to a region R as the minimum distance from the patch p to any patch in R,

$$D(p,R) = \min_{r\in R}(D(p,r)) \quad (8)$$

and the distance between two regions R_1 and R_2 as the minimum distance between their patches,

$$D(R_1,R_2) = \min_{r\in R_1}(D(r,R_2))\,. \quad (9)$$



(a) KL-divergence, 5D (b), Bhattacharyya, 3D

(c) Mahalanobis, 5D (d) Mahalanobis, 3D

Figure 3. MDS results of four different feature and distance configurations for the row-1 image of Figure 5 (68 patches).

To compare different feature and distance configurations, we use non-metric multidimensional scaling (MDS) [25] to embed patches from an image into a 2D space. Figure 3 shows the embedded feature-space for various feature/distance combinations, with points marked according to their true foreground/background labels (fgLabels/bgLabels), and inclusion in the background mask

(bgPriors). The foreground and the background patches are better separated with the 5D D_M and D_K measures, compared to the 3D Lab or Bhattacharyya distance [12,15]. In addition, Figures 3a and 3c reveal that, under D_M and D_K, the set of background prior patches is indeed representative of the whole background.

2.4. Binary classification

With the patch distances defined in Section 2.3 we build our foreground extraction algorithm. In light of the assumption that the pre-assigned mask provides sufficient background statistics, we first initialize the background and foreground priors, and then gradually refine the segmentation by merging patches into the foreground. For simplicity, we consider a generic distance D, which can be either D_M or D_K from the previous section.

The first step is to obtain the background prior B by selecting patches overlapped with the background mask. Next, patches are selected for the initial foreground prior F if the distance from patch p_i to the background prior B is greater than a threshold D_t,

$$L(p_i) = 1, \text{ if } D(p_i, B) > D_t . \tag{10}$$

Considering the distance in (8), (10) selects a patch for the foreground prior if it is sufficiently far from all known background patches.

Starting from the initial foreground set F, we use a greedy search to progressively label the patches within the region of interest. First, all unlabeled patches are sorted in descending order by their distances from the background prior B. Patches are then labeled in turn by comparing them with the current background and foreground sets,

$$L(p_i) = \begin{cases} 1 & \text{if } D(p_i,F) \le D(p_i,B) \\ 0 & \text{if } D(p_i,F) > D(p_i,B) \end{cases}. \tag{11}$$

The foreground set F is *updated online*, when a new patch is assigned to the foreground. On the other hand, the background prior B *remains fixed* throughout, in order to avoid error propagation. This is based on the assumption that the background prior B contains sufficient information about the background statistics.

The above routine relies on a predefined threshold D_t. Different thresholds yield different segmentations. Based on the fact that the squared probability distance from a sample of a Gaussian population to the center mode obeys a χ^2 distribution [26], as implied by the Mahalanobis distance terms in (6) and (7), we can convert a confidence interval of the χ^2 distribution to a corresponding distance interval $[D_l,D_u]$, and use it to bound D_t. Then we exhaustively try D_t thresholds over $[D_l,D_u]$, compute an evaluation score from every segmentation result, and output the solution associated with the highest score. In the next section, we show that this 1D brute-force search can be done very efficiently. In practice, we bound D_t loosely with D_l=5.0 and D_u=50.0, which correspond to 0.58 and 1-10^{-9} critical values of the 5-dof χ^2 distribution. This interval allows a big enough initial foreground prior set but excludes unnecessary initializations. Figures 1d and 1e show an example set of candidate segmentations produced by varying the threshold D_t, for both D_M and D_K.

2.5. Generating multiple hypotheses

The segmentation procedure in the previous section generates a large set of candidate segmentations by varying the threshold D_t. Taking into account the fact that perceptually meaningful segmentations may correspond to different cost functions, we generate multiple segmentation hypotheses by selecting the best candidate segmentations according to several evaluation scores. In particular, we consider three score functions that maximize the global distance between background and foreground patches, using the distance measure D: *sum-cut*, *average-cut* and *maxmin-cut* (abbreviated as s-cut, a-cut, and m-cut). Other score functions could also be easily incorporated to enclose any available prior knowledge (e.g., that of [13]).

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

$$D_t^{(s)} = \arg\max_{D_t \in [D_l, D_u]} \sum_{f \in F(D_t)} D(f, B(D_t)), \tag{12}$$

where $F(D_t)$ and $B(D_t)$ are respectively the foreground and the background groups in the final segmentation map computed from the threshold D_t.

Taking the average instead of the sum yields the *average-cut* score function:

$$D_t^{(a)} = \arg\max_{D_t \in [D_l, D_u]} \frac{1}{|F(D_t)|} \sum_{f \in F(D_t)} D(f, B(D_t)) . \tag{13}$$

Finally, the *maxmin-cut* score maximizes the minimum distance between foreground and background patches,

$$D_t^{(m)} = \arg\max_{D_t \in [D_l, D_u]} D(F(D_t), B(D_t)) . \tag{14}$$

Each score function favors a different type of segmentation. m-cut prefers segmentations where the foreground and background pieces have a clear boundary in the feature space, corresponding to the global optimization of (9). a-cut yields segmentations where the global distance from the foreground to the background is sufficiently far, corresponding to the global optimization of (8). s-cut gives a conservative estimate of the solution near D_l, which is particularly good when a tight mask box is assigned.

The combinatorial property of the distance functions in (8) and (9) ensures that the score functions (12-14) are not smooth within the interval $[D_l,D_u]$, but piecewise constant. In other words, the optimal value occurs in an interval of D_t, instead of at a single point. This fact simplifies the exhaustive search to a discrete one, and we need only check some key points within the interval $[D_l,D_u]$. We propose the

following D_t-solving algorithm. First we sort all values of $D(p,B)$ in ascending order,

$$D_l = d_0 < d_1 < d_2 < \ldots < d_n < d_{n+1} = D_u, \quad (15)$$

where n is the number of distances within $[D_l, D_u]$ in $D(p,B)$ and d_i is the i-th smallest value. We then define a set of test D_t thresholds as the midpoints between two successive distances, $S_{D_t} = \{(d_i + d_{i+1})/2\}_{i=0}^{n}$. The best scoring segmentation is determined by a discrete search, by running the figure-ground segmentation for each threshold in S_{D_t}. This method greatly reduces the computational cost in comparison to an exhaustive search.

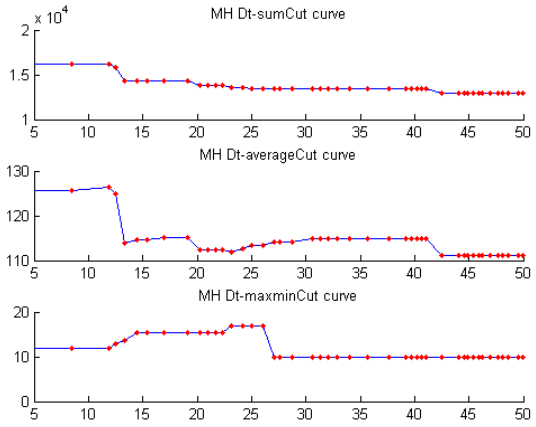


Figure 4. Three D_t-*score* curves for the row-3 image of Figure 5.

Figure 4 shows an example of the three D_t-score functions sampled at all key D_t points (plotted as red dots). It is worth mentioning that the solutions to m-cut may not be unique. For example, the optimal m-cut interval spans 3 different candidates in Figure 4. We note that 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. In total, we generate 8 hypothesis segmentations for each image (one s-cut, one a-cut, two m-cuts for each distance function D_M and D_K), as illustrated in Figures 1f and 1g.

Overall, the labeling algorithm under one evaluation score can be deemed as a hill-climbing optimization with multiple initializations. Each threshold D_t starts a greedy searching procedure from an initial position and ends at a local peak. The adoption of multiple initializations reduces the risk of getting stuck to local optimum. The strategy of multiple evaluation scores greatly adds the chance of success and provides a representative candidate group for the final automatic or manual selection.

2.6. Automatic hypotheses selection

Finally, given the multiple hypotheses, we borrow the idea of classifier fusion to automatically obtain the final segmentation [27]. In particular, we propose two methods for determining the final segmentation by pooling over all the hypotheses.

The first scheme, which we denote as *similarity voting*, is based on the assumption that a good solution is likely to be selected by different score functions. Therefore, we let the eight hypotheses vote and output the winning segmentation. In the case of multiple winners, we select the one most similar to the other hypotheses. The similarity between two foreground areas F_1 and F_2 is defined by a scale invariant measure $S(F_1, F_2) = (F_1 \cap F_2)/(F_1 \cup F_2)$ [28].

The second scheme, which we denote as *probability map*, computes the final segmentation from the weighted sum of the eight hypotheses. We first define the dissimilarity between two hypothesis segmentations by $D(F_1, F_2) = 1 - S(F_1, F_2)$, and construct a symmetric affinity matrix A with entries $A(i,j) = A(F_i, F_j) = \exp(-D(F_i, F_j)^2 / 2\sigma^2)$, where the parameter $\sigma^2 = Var(D)$ is determined from the pairwise distance value set [7,19]. Next, we build a probability map as the weighted sum of the hypothesis segmentations,

$$P = \sum_i w_i^2 F_i \cdot \quad (16)$$

The weights are determined using the following constrained optimization,

$$\max\ w^T A w, s.t.\ \|w\|^2 = 1. \quad (17)$$

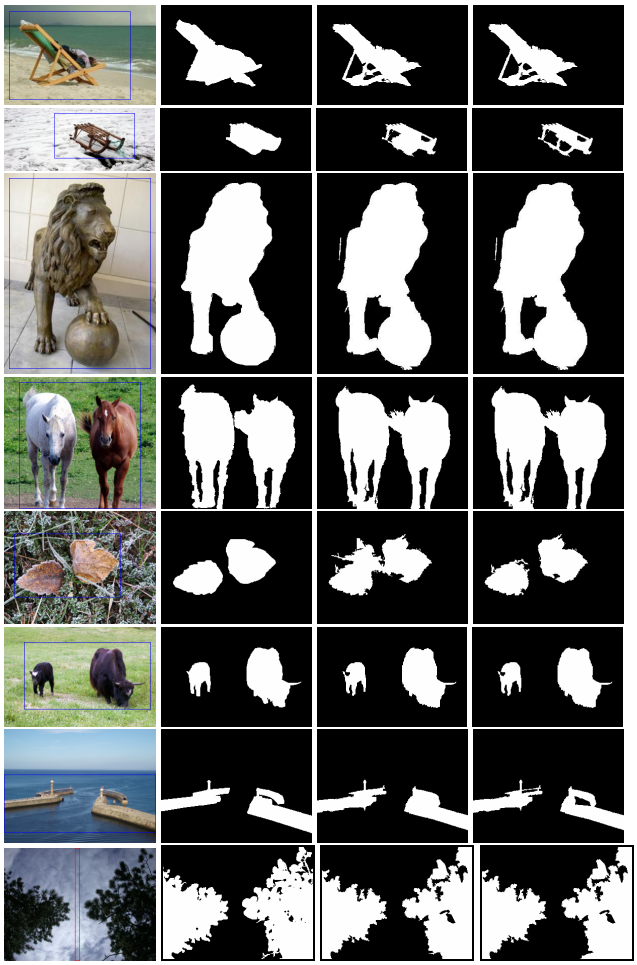
(17) finds the weights that maximize the total similarity among all weighted hypotheses. It is a standard Rayleigh quotient maximization problem, and the optimal w is given by the top eigenvector of A. (16) coarsely characterizes the probability that a patch belongs to the foreground. The final segmentation is obtained by binarizing P with a threshold P_t=0.5. Figures 1h and 1i show the final result produced by the similarity voting scheme and the probability map method.

3. Experiments

In this section, we perform experiments on several popular datasets and report both quantitative and qualitative results. To illustrate the advantage of multiple hypotheses generation, we also consider a user-selection procedure, where the user selects the best segmentation among the eight hypotheses. Experiments were run on a desktop PC with an Intel core-i5 CPU 2.8Ghz dual core processor and 8GB RAM.

3.1. Results on four datasets with ground truths

In our experiments, we consider 4 datasets. The first two datasets are from the Weizmann evaluation dataset containing 100 1-obj images and 100 2-obj images [29]. We also use the IVRG dataset [30] (1000 images) and the grabcut dataset [31] (50 images).



original & mask ground truth similarity voting user selection

Figure 5. Weizmann test examples. Rows 1-3 are 1-obj examples. Rows 4-8 are 2-obj examples. The auto-selection is equally good as the user selection for rows 1,3,4,6,8; and slightly worse for rows 2,5,7. The outsides of the blue boxes or the inside of the red box define the background masks.

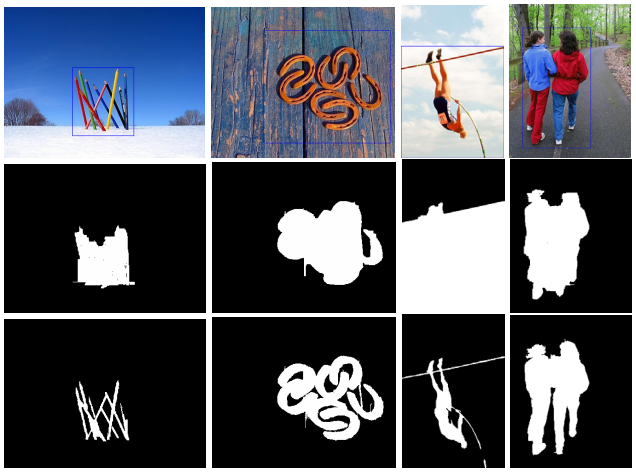


Figure 6. IVRG test example results. Top: image & mask; middle: grabcut results; bottom: f-g classification auto-selection results.

Figure 5 displays some example segmentations from the Weizmann dataset. Our adaptive f-g classification successfully labels background holes and multiply connected components. It even identifies many details missed in the manual-made truths (rows 1, 2, 7 of Figure 5). Figure 6 shows some example segmentations from the IVRG dataset. Our method better segments foregrounds with strongly irregular contours, compared to grabcut [7].

The performance on each image is evaluated using F-measure, $F=2PR/(P+R)$, where P and R are the precision and recall values. Table 1 reports the 95% confidence intervals of the average F-measure on each of the 4 datasets. First, looking at the *single* hypothesis segmentations produced by each of the four score functions and 2 distances, we note that there is no clear best score function or distance measure for all scenarios. For example, s-cut is best for Weizmann 1-obj, but does poorly on Weizmann 2-obj. The performances using the two distance functions, D_M and D_K, are also mixed, with the best function depending on the images. Leveraging multiple hypotheses, using our *similarity voting* or *probability map* scheme, improves the F-measure on *all* the datasets (e.g., from 0.91 to 0.93 on Weizmann 1-obj, or 0.92 to 0.94 on IVRG). This improvement in performance indicates that the different score functions and distances *complement* each other, and that the automatic selection scheme is capable of identifying good segmentation among the multiple hypotheses.

Table 1. F-measures on four image datasets.
*: number of times auto-selection at least as good as user-selection.

	Weizmann 1-obj	Weizmann 2-obj	ivrg images	Grabcut images
s-cut (D_M)	0.91 ± 0.014	0.83 ± 0.028	0.91 ± 0.005	0.87 ± 0.034
s-cut (D_K)	0.91 ± 0.014	0.82 ± 0.030	0.91 ± 0.005	0.86 ± 0.035
a-cut (D_M)	0.89 ± 0.029	0.86 ± 0.031	0.92 ± 0.008	0.88 ± 0.056
a-cut (D_K)	0.89 ± 0.031	0.84 ± 0.036	0.91 ± 0.009	0.90 ± 0.044
m_1-cut (D_M)	0.89 ± 0.031	0.88 ± 0.028	0.91 ± 0.009	0.88 ± 0.056
m_1-cut (D_K)	0.89 ± 0.033	0.87 ± 0.029	0.91 ± 0.010	0.91 ± 0.044
m_2-cut (D_M)	0.89 ± 0.032	0.87 ± 0.031	0.92 ± 0.009	0.88 ± 0.056
m_2-cut (D_K)	0.89 ± 0.034	0.86 ± 0.034	0.91 ± 0.010	0.91 ± 0.054
similarity voting	**0.93 ± 0.010(49*)**	**0.88 ± 0.022(64)**	**0.94 ± 0.004(456)**	**0.93 ± 0.018(22)**
probability map	**0.93 ± 0.010(48)**	**0.88 ± 0.022(58)**	**0.94 ± 0.004(371)**	**0.93 ± 0.016(21)**
user selection	**0.94 ± 0.009**	**0.89 ± 0.021**	**0.96 ± 0.003**	**0.94 ± 0.015**
references	0.85 ± 0.035 [7]	0.80 ± 0.046 [7]	0.93 ± 0.006 [7]	0.89 ± 0.033 [7]
	0.93 ± 0.009[13]	0.68 ± 0.053[9]		
	0.87 ± 0.010 [5]	0.66 ± 0.066[17]		

Comparing the two selection schemes, there is no statistical difference in the F-measure between similarity voting and probability map. In general, the weights computed by probability map are strongly biased towards the similarity voting winners, and hence the results are similar. Similarity voting is easier to use, whereas probability map provides more flexibility by tuning the threshold P_t., e.g., as in Figure 7. Adaptive selection of P_t is a topic of future work.

original similarity voting P_t=0.5 P_t=0.8 P_t=0.99

Figure 7. Probability map threshold control

Some images have a cluttered background but a relatively simple foreground. This may hinder the background prior assumption. To treat 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 box in Figure 8). After the segmentation we reverse the foreground and the background to obtain the final result. In our experiments this switch operation is executed for 4 images in the grabcut dataset and 6 images in the IVRG dataset. In all cases the method improves both grabcut and f-g classification. Figure 8 shows an image that can be improved by the switch operation. This example gives a rule of thumb for mask selection: the background mask should be statistically simple and easily characterized by a bounding box.

(a) original & mask (b) f-g classification result (c) grabcut result

Figure 8. A figure-ground switching example.

The last row of Table 1 shows the results of some reference algorithms ([13,5,9,17,7]), in particular the grabcut algorithm using the same mask box initializations [7]. Our algorithm is comparable to 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). Finally, user selection provides an upper-bound on the performance of the multiple hypothesis approach. The automatic selection schemes perform close to the user selection. Improving auto-selection to match user-selection better is a topic of future work.

For the grabcut image set, we also compare the error rate with the result reported in [15]. The error rate is defined as the percentage of mislabeled pixels within the foreground mask. Table 2 shows that f-g classification outperforms grabcut [7] and iterated distribution matching [15].

Table 2. Average error rate comparison on the grabcut image set

Similarity voting	Probability map	grabcut[7]	distribution matching[15]
5.7%	**5.4%**	8.1%	7.1%

Finally, Table 3 compares the average running time and the average F-measures between the grabcut algorithm and the f-g classification auto-selection for the four datasets. The adaptive f-g classification improves on both execution speed and segmentation quality.

Table 3. Performance comparison of grabcut and f-g classification

		Weizmann 1	Weizmann 2	IVRG	Grabcut
time	grabcut	5.43 s	4.64 s	5.51 s	16.54 s
	f-g	**2.78 s**	**2.19 s**	**3.79 s**	**9.87 s**
$\bar{F}$	grabcut	0.85 ± 0.035	0.80 ± 0.046	0.93 ± 0.006	0.89 ± 0.033
	f-g	$\mathbf{0.93 \pm 0.010}$	$\mathbf{0.88 \pm 0.022}$	$\mathbf{0.94 \pm 0.004}$	$\mathbf{0.93 \pm 0.018}$

3.2. The Berkeley segmentation dataset

In a final experiment, we evaluate our method on the Berkeley segmentation dataset [32]. Figure 9 gives the manual selections 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 (4) are also given for the Berkeley examples. The adaptive initialization works well and generates good mean-shift patches. The two distance functions D_K and D_M complement each other, and improve the chance of producing good segmentations. The images in the figure show the advantages of multiple hypotheses and manual selection. That is, although automatic selection is quite powerful, it may still miss some better choices from the multiple hypotheses. Manual selection becomes important in such cases if high quality is demanded, especially for challenging background or foreground topologies. As a typical example, almost all connected components and all holes in image 370036 are successfully identified.

4. Conclusion

In this paper, we have proposed an adaptive figure-ground classification algorithm to automatically extract foreground region using a bounding-box based background prior. The image is first over-segmented by adaptive mean-shift. Then the background and foreground regions are gradually refined using the background and foreground priors. Multiple hypotheses are generated from different distance measures and evaluation score functions. The best segmentation is automatically selected with a voting or weighted combination scheme. Our method achieves great success for challenging scenes, particularly when there are irregular or multiple-connected foregrounds.

Acknowledgement

The authors would like to thank Prof. Kenichi Kanatani for very beneficial discussions, and the anonymous reviewers for their helpful comments. The work in this paper was supported by national natural science foundation of China (60973052).

References

[1] V. Gulshan, C. Rother, A. Criminisi, A. Blake and A. Zisserman, Geodesic star convexity for interactive image segmentation. In CVPR2010, pp. 3129-2136.

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

[4] X. Bai and G. Sapiro, A geodesic framework for fast interactive image and video segmentation and matting. In ICCV2007, pp. 1-8.

[5] 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.

Figure 9. Example user-selected segmentations (*: coincide with auto-selection; s: voted by s-cut; a: voted by a-cut; m: voted by m-cut) The auto-selections (in supplemental materials) often coincide with the user-selections and are in general slightly worse in other cases.

[6] G. Hua, Z. Liu, Z. Zhang, and Y. Wu, Iterative local global energy minimization for automatic extraction of objects of interest, PAMI, 28(10), 1701-1706, 2006.

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

[10] R. Szeliski, *et al.*, A comparative study of energy minimization methods for markov random fields with smoothness-based priors. PAMI, 30(6):1068–1080, 2008.

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

[12] 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, CVPR2010, pp. 3288–3295.

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

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

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

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

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

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

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

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

[21] 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.

[22] T. Malisiewicz and A. Efros. Improving spatial support for objects via multiple segmentations, BMVC2007.

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

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

[25] J. M. Lattin, J. D. Carrol, P. E. Green, Analyzing multivariate data, Thomson Brooks/Cole, 2003.

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

[27] J. Kittler, M. Hatef, R. Duin, and J. Matas, On combining classifiers, IEEE Trans. PAMI, 20(3), pp. 226–239, 1998.

[28] 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.

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

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

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

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