# Distribution Matching with the Bhattacharyya Similarity: a Bound Optimization Framework

Ismail Ben Ayed[*†§] and Kumaradevan Punithakumar[‡] and Shuo Li[*†]

[*] GE Healthcare, London, ON, Canada

[†] University of Western Ontario, London, ON, Canada

[‡] University of Alberta, Edmonton, AB, Canada

[§] Corresponding author

Email:`[ismail.benayed]@ge.com`

## Abstract

We present efficient graph cut algorithms for three problems: (1) finding a region in an image, so that the histogram (or distribution) of an image feature within the region most closely matches a given model; (2) co-segmentation of image pairs and (3) interactive image segmentation with a user-provided bounding box. Each algorithm seeks the optimum of a global cost function based on the Bhattacharyya measure, a convenient alternative to other matching measures such as the Kullback–Leibler divergence. Our functionals are not directly amenable to graph cut optimization as they contain non-linear functions of *fractional* terms, which make the ensuing optimization problems challenging. We first derive a family of parametric bounds of the Bhattacharyya measure by introducing an auxiliary labeling. Then, we show that these bounds are *auxiliary functions* of the Bhattacharyya measure, a result which allows us to solve each problem efficiently via graph cuts. We show that the proposed optimization procedures converge within very few graph cut iterations. Comprehensive and various experiments, including quantitative and comparative evaluations over two databases, demonstrate the advantages of the proposed algorithms over related works in regard to optimality, computational load, accuracy and flexibility.

## Index Terms

Graph cuts, bound optimization, auxiliary functions, Bhattacharyya measure.

# I. INTRODUCTION

Finding accurately a meaningful region in an image, for instance a person in a photograph or an organ in a medical scan, is a subject of paramount importance in computer vision for its theoretical and methodological challenges, and numerous useful applications. Current major application areas include content-based image retrieval [1], image editing [2], medical image analysis [3], remote sensing [4], surveillance [5] and many others. The problem we tackle in this study consists of segmenting one or several images into two regions (a foreground and a background), so that an image feature (e.g., color, textures, edge orientations, motion) within the segmentation regions follows some available *a priori* information. Such priors are necessary to obtain semantic segmentations that are unattainable with unsupervised algorithms [6], [7], [8]. The following variants of the problem are of broad interest in computer vision:

- *Co-segmentation of image pairs:* Introduced initially in the work of Rother et al. [9], the problem amounts to finding the same object (foreground) in a pair of images. Facilitating segmentation of an image using minimal prior information from another image of the same object, co-segmentation has bestirred several recent investigations [10], [2], [11], [12], [13] and has been very useful in object recognition and image retrieval [14], [1], [15], [9], as well as image editing [2] and summarization [16].

- *Interactive image segmentation:* Of great practical importance in image editing, interactive segmentation uses minimal user interaction, for instance simple scribbles or bounding boxes, to learn prior information from the current image. Embedding clues on user intention facilitates segmentation, and has been intensively researched in recent years [17], [18], [19], [20], [21], [22].

- *Segmentation with offline learning:* Segmenting a class of images with similar patterns occurs in important applications such as medical image analysis. In this case, offline learning of prior information from segmented training images is very useful [23], [24].

- *Tracking:* In the context of tracking a target object throughout an image sequence, one can segment the current frame using image feature cues learned from previously segmented frames [25], [26].

A sub-problem which arises in these variants is the problem of finding a segmentation region consistent with a model distribution of the image feature [27], [28], [25], [29]. This requires optimization of a global measure of similarity (or discrepancy) between distributions (or histograms). In this connection, several recent studies proved that optimizing global measures outperforms standard algorithms based on *pixelwise* information in the contexts of co-segmentation [9], [11], [13], segmentation [30], [23], [31] and tracking [32], [33], [25]. Moreover, region-based image similarity measures can be very useful in image retrieval [9]. The following discusses prior art in this direction and the contributions of this study.

## A. Prior art

*1) Active contours and level sets:* The use of a global similarity measure in image segmentation often leads to challenging optimization problems. The solutions were generally sought following gradient-based optimization via active contour (or level set) partial differential equations [34], [27], [23], [31], [25], [29]. An Euler-Lagrange equation of contour motion is derived so as to increase the consistency between the foreground region enclosed by the active contour and a given model [25], [29] or to maximize the discrepancy between the two segmentation regions [31], thereby reaching a local optimum at convergence. Several measures were studied within the active contour framework, for instance, the Kullback–Leibler divergence [29], the Earth Mover's distance [27] and the Bhattacharyya coefficient [23], [25], [31]. The Bhattacharyya coefficient has a fixed (normalized) range, which affords a conveniently practical appraisal of the similarity, and several other desirable properties [35]. We will discuss some of these properties in the next section.

Along with an incremental gradient-flow evolution, active contours may require a large number of updates of computationally onerous integrals, namely, the distributions of the regions defined by the curve at each iteration and the corresponding measures. This can be very slow in practice: it may require up to several minutes on typical CPUs for a color image of a moderate size [28]. Furthermore, the robustness

of the ensuing algorithms inherently relies on a user initialization of the contour close to the target region and the choice of an approximating numerical scheme of contour evolution.

*2) Graph cuts:* Discrete graph cut optimization [36], [37], [38], which views segmentation as a label assignment, has been of intense interest recently because it can guarantee global optima and numerical robustness, in nearly real-time. It has been effective in various computer vision problems [39], for instance, segmentation [19], [21], [40], [20], tracking [41], [42], motion estimation [43], visual correspondence [44] and restoration [37]. Unfortunately, only a limited class of functions can be directly optimized via graph cuts. Therefore, most of existing graph cut segmentation algorithms optimize a sum of pixel dependent or pixel-neighborhood dependent data and variables. Global measures of similarity between distributions have been generally avoided because they are not directly amenable to graph cut optimization. Notable exceptions include the co-segmentation works in [9], [13], [11], [10] as well as the interactive segmentation algorithms in [45], [46]. For instance, in the context of co-segmentation of a pair of images, the problem consists of finding a region in each image, so that the histograms of the regions are consistent. Rother et al. [9] pioneered optimization of the $L_1$ norm of the difference between histograms with a trust region graph cut (TRGC) method. They have shown that TRGC can improve a wide spectrum of research: it outperformed standard graph cut techniques based on pixelwise information in the contexts of object tracking and image segmentation, and yielded promising results in image retrieval. Unfortunately, TRGC is very sensitive to initializations [10]. In [13], Mukherjee et al. suggested to replace the $L_1$ by the $L_2$ norm, arguing that the latter affords some interesting combinatorial properties that befit graph cut optimization. After linearization of the function, the problem is solved by graph cuts [47], [48] via roof-duality relaxation [49]. However, such relaxation yields only a partial solution with some pixels left unlabeled. How to label these pixels without loosing ties to the initial problem is an important question [10]. Moreover, the optimization in [13] builds a graph whose size is twice the size of the image. In [10], the authors combine dual decomposition [50] and TRGC to solve the $L_p$ optimization problems in [9], [13], yielding an improvement in optimality. Hochbaum and Singh proposed to maximize the dot product between histograms [11], which results in a sub-modular quadratic function optimization solvable with a single graph cut. Unfortunately, the growth of the graph size in [11] behaves quadratically.

The cost functions in [9], [13], [11] are based on the *unnormalized* histogram, which depends on the size (or scale) of the region. Therefore, they do not afford a scale-invariant description of the class of target regions. The ensuing co-segmentation algorithms enforce the number of foreground pixels to be the same in both images. Therefore, they are seriously challenged when the target foregrounds have different sizes [10]. In such difficult co-segmentation cases, or in other applications where the size of the target region is different from the size of the learning region, for instance, tracking an object whose size varies over an image sequence, the unnormalized histogram requires additional optimization/priors with respect to region size [9]. Furthermore, in information theory, it transpires that an $L_p$ measure does not afford the best appraisal of the similarity between distributions [35].

### B. Contributions

This study investigates efficient graph cut algorithms for three problems: (1) finding a region in an image, so that the distribution (kernel density estimate) of an image feature within the region most closely matches a given model distribution; (2) co-segmentation of image pairs and (3) interactive image segmentation with a user-provided bounding box. Each algorithm seeks the optimum of a global functional based on the Bhattacharyya measure, a practical alternative to other matching measures such as the Kullback-Leibler divergence. Our functionals are not directly amenable to graph cut optimization as they contain non-linear functions of *fractional* terms, which make the ensuing optimization problems challenging[1]. We first derive a family of parametric bounds of the Bhattacharyya measure. Then, we show that these bounds are

---

[1]Note that most of related methods use *unnormalized* histograms, e.g., [9], [13], [11], which do not give rise to fractional terms. In our case, the use of distributions is more flexible (e.g., it affords scale invariance), but comes at the price of a more challenging optimization problem (due to fractional terms).

*auxiliary functions* (See Section II-A2) of the Bhattacharyya measure, a result which allows us to solve each problem efficiently via graph cuts. We show that the proposed optimization procedures converge within very few graph cut iterations. Comprehensive and various experiments, including quantitative and comparative evaluations over two data sets, demonstrate the advantages of the proposed algorithms over related works in regard to optimality, computational load, accuracy and flexibility. These advantages are summarized as follows.

- *Computational load:* The proposed bound optimization brings several computational advantages over related methods. First, it builds graphs that have the same size as the image, unlike the graph cut methods in [13], [11]. Second, the ensuing algorithms converge in very few iterations (typically less than 5 iterations). This will be demonstrated in the experiments. Third, the algorithm is robust to initialization and does not require sophisticated initialization procedures as with TRGC [9]. It is possible to use trivial initializations.

- *Accuracy and optimality:* Quantitative comparisons with related recent methods over a several public databases demonstrate that the proposed framework brings improvements in regard to accuracy and solution optimality.

- *Flexibility:* Unlike the unnormalized histogram models in [9], [13], [11], the proposed framework yields co-segmentation and segmentation algorithms, which handle accurately and implicitly variations in the size of the target regions because the Bhattacharyya measure references kernel densities and, therefore, is scale-invariant.

We presented preliminary results of this work at the CVPR conference [28]. This TPAMI version expands significantly on [28]: It contains new theoretical justifications and algorithms, reports new experiments/comparisons and includes new discussions, details and illustrations. The following summarizes the most important differences with the CVPR version:

- The bound in [28] is an approximate (not exact) auxiliary function. Although the approximation in [28] yielded a competitive performance in practice, there is no theoretical guarantee that the energy decreases at each iteration. In this extended version, we re-wrote the bound so as to obtain an exact (not approximate) auxiliary function, and derived a completely new proof based on rigorous analytical arguments. The new arguments guarantee that the energy will not increase during iterations.

- The CVPR version addresses the problem of finding a single-image segmentation consistent with a known (fixed) model distribution. This journal submission addresses two other problems where model distributions are unknown variables that have to be estimated with the segmentations: (i) co-segmentation of image pairs; and (ii) interactive image segmentation with a user-provided bounding box.

- All the experiments in [28] are based on the exact knowledge of the ground truth color distribution[2]. However, such assumption is not valid in most of practical scenarios where the actual distribution is not known exactly. In this journal extension, we provide several new sets of experiments. We added a significant number of realistic segmentation/co-segmentation examples along with quantitative evaluations and comparisons with recent methods.

Finally, it is worth mentioning the recent studies in [46], [51], which extended the bound-optimization ideas of our CVPR paper [28] and showed competitive performances in the context of interactive segmentation. Using the Bhattacharyya measure and bound optimization, the authors of [46] stated segmentation as a sequence of distribution-matching processes combined with an additional Bhattacharyya term that maximizes the discrepancy between the foreground and background distributions. Experimentally, the methods in [46], [51] showed improvements over standard algorithms based on pixelwise log-likelihood information [20], [19], [21].

The remainder of this paper is organized as follows. The next section details the cost functions and the bound optimization. First, we start with the problem of finding a region consistent with a known (fixed)

---

[2]In [28], the model is learned from the ground-truth segmentation of the testing image. The purpose of such experiments was to compare the performance of the proposed optimization technique to standard prior-art algorithms (e.g., level sets) in regard to optimality and speed.

model distribution. Then, we extend the formulation to co-segmentation and interactive segmentation, where region models become variables that have to be estimated iteratively with the segmentations. Section III discusses comprehensive experiments, including the application of the algorithms to various scenarios as well as quantitative evaluations and comparisons with other methods. Section IV contains a conclusion.

## II. GRAPH CUTS WITH GLOBAL BHATTACHARYYA TERMS

### A. Finding a region consistent with a known (fixed) model distribution

*1) The cost function:* Let $C = [0,1]^n$ be an $n$-dimensional color space, and $\mathbf{I} = (I_1, I_2, \ldots, I_N)$ a given image, where $I_i \in C$ denotes the color of pixel $i$ and $N$ is the number of pixels in the image. Each segmentation of $\mathbf{I}$ can be identified by a binary vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, with $x_i = 1$ indicating that pixel $i$ belongs to the target region (foreground) and $x_i = 0$ indicating background membership. Each segmentation $\mathbf{x}$ yields a distribution over colors $c \in C$ within the corresponding foreground region:

$$p_{\mathbf{x}}(c) = \frac{\sum_i x_i K_i(c)}{|\mathbf{x}|} \tag{1}$$

where $|\mathbf{x}| = \sum_i x_i$ is the size of the foreground region corresponding to binary vector $\mathbf{x}$. Possible choices of $K_i$ are the Dirac function $\delta(I_i - c) = 1$ if $I_i = c$ and $0$ otherwise, which yields the normalized histogram, or the Gaussian kernel $(2\pi\sigma^2)^{\frac{n}{2}} exp^{-\frac{\|I_i - c\|^2}{2\sigma^2}}$, with $\sigma$ the width of the kernel. The purpose of the algorithm is to seek a segmentation $\mathbf{x}$ so that the corresponding foreground color distribution $p_{\mathbf{x}}$ most closely matches a known target distribution $q$. To achieve this, we use the negative Bhattacharyya coefficient:

$$B(\mathbf{x}|q) = -\int_C \sqrt{p_{\mathbf{x}}(c)q(c)} \tag{2}$$

The range of $B(\mathbf{x}|q)$ is $[-1, 0]$, $0$ corresponding to no overlap between the distributions and $-1$ to a perfect match. Thus, our objective is to minimize $B(\mathbf{x}|q)$ with respect to $\mathbf{x}$. The Bhattacharyya coefficient has the following geometric interpretation. It corresponds to the cosine of the angle between the *unit* vectors $(\sqrt{p_{\mathbf{x}}(c)}, c \in C)^T$ and $(\sqrt{q(c)}, c \in C)^T$ (These vectors are unit if we use the $L_2$ norm). Therefore, it considers explicitly $p_{\mathbf{x}}$ and $q$ as distributions by representing them on the unit hypersphere. Note that the Bhattacharyya coefficient can also be regarded as the normalized correlation between $(\sqrt{p_{\mathbf{x}}(c)}, c \in C)^T$ and $(\sqrt{q(c)}, c \in C)^T$.

The Bhattacharyya coefficient has a fixed (normalized) range, which affords a conveniently practical appraisal of the similarity. This is an important advantage over other usual similarity measures such as the Kullback–Leibler divergence or the $L_p$ norms. It is worth noting that the distribution-matching term is not invariant with respect to illumination changes. This will be demonstrated in the experiments.

To avoid complex segmentations and isolated fragments in the solution, we add a regularization term to our objective function:

$$S(\mathbf{x}) = \sum_{\{i,j\} \in \mathcal{N}} w_{i,j}[1 - \delta(x_i - x_j)] \tag{3}$$

where $\mathcal{N}$ is the set of neighboring pixels in a t-connected grid ($t = 4, 8$ or $16$). Pairwise weights $w_{i,j}$ are typically determined either by the color contrast and/or spatial distance between pixels $i$ and $j$. Our purpose is to minimize the following function with respect to $\mathbf{x}$:

$$E(\mathbf{x}|q) = B(\mathbf{x}|q) + \lambda S(\mathbf{x}), \tag{4}$$

with $\lambda$ a positive constant. As we will eventually use graph cuts in the main step of our algorithm, we assume $w_{i,j} \geq 0$, which means $S(\mathbf{x})$ is a *sub-modular* function of binary segmentation $\mathbf{x}$; See [38].

*2) Efficient bound optimization:* A function $\tilde{h}(\mathbf{x}, \mathbf{y})$ is called auxiliary function of $h$ at $\mathbf{y}$ if it satisfies the following properties:

$$h(\mathbf{x}) \leq \tilde{h}(\mathbf{x}, \mathbf{y}) \quad \forall \mathbf{x} \tag{5}$$

$$h(\mathbf{y}) = \tilde{h}(\mathbf{y}, \mathbf{y}) \tag{6}$$

When cost function $h$ cannot be minimized directly, one can minimize a sequence of auxiliary functions, starting at some initial $\mathbf{y}^{(0)}$. At each iteration $t$, $t = 1, 2, \ldots$, this amounts to solving:

$$\mathbf{y}^{(t+1)} = \underset{\mathbf{x}}{\text{argmin}} \quad \tilde{h}(\mathbf{x}, \mathbf{y}^{(t)}) \tag{7}$$

Properties (5) and (6) guarantee $h(\mathbf{y}^{(t+1)}) \leq h(\mathbf{y}^{(t)})$ and, if $h$ is bounded from below, the auxiliary-function moves in (7) converge to a local minimum of $h$; See [52].
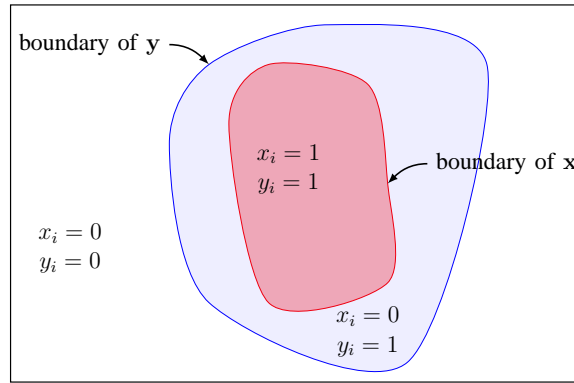


Fig. 1. Derivation of the auxiliary function at $\mathbf{y}$: We assume the foreground region of some fixed $\mathbf{y}$ includes the variable foreground region defined by $\mathbf{x}$. Fixed $\mathbf{y}$ corresponds to the solution obtained at a previous iteration.

To minimize $E(\mathbf{x}|q)$ in a bound optimization framework, we need to design an auxiliary function $\tilde{B}(\mathbf{x}, \mathbf{y}|q)$ for the negative Bhattacharyya coefficient $B(\mathbf{x}|q)$. At each step, we assume the scenario depicted in Fig. 1, where the foreground region of some fixed $\mathbf{y}$ includes the foreground region defined by $\mathbf{x}$, i.e., $\mathbf{x} \leq \mathbf{y}$. Let us start by expressing $p_{\mathbf{x}}$ as a multiple of $p_{\mathbf{y}}$ by choosing[3] some functions $f$ and $g$ such that

$$p_{\mathbf{x}}(c) = \frac{f(c, \mathbf{x}, \mathbf{y})}{g(c, \mathbf{x}, \mathbf{y})} p_{\mathbf{y}}(c) \tag{8}$$

As it will become clear later, the choice of specific forms of $f$ and $g$ will be important in deriving an auxiliary function of $B(\mathbf{x}|q)$. Regardless, plugging (8) into $B(\mathbf{x}|q)$ yields:

$$B(\mathbf{x}|q) = -\int_C \sqrt{p_{\mathbf{y}}(c)q(c)} \sqrt{\frac{f(c, \mathbf{x}, \mathbf{y})}{g(c, \mathbf{x}, \mathbf{y})}} dc \tag{9}$$

The main computational difficulty of (9) comes from the non-linear ratio function $\sqrt{\frac{f(c,\mathbf{x},\mathbf{y})}{g(c,\mathbf{x},\mathbf{y})}}$, which is not directly amenable to powerful optimizers such as graph cuts. In the Lemma that follows, we circumvent this difficulty by showing that this ratio function can be bounded by a linear combination of $f$ and $g$ when these functions are within interval $[0, 1]$.

**Lemma 1:** $\forall \alpha \in [0, \frac{1}{2}]$ *and if* $f, g \in [0, 1]$, *we have:*

$$-\sqrt{\frac{f}{g}} \leq \alpha g - (1 + \alpha)f \tag{10}$$

[3]For equality to hold, we need to choose $f$ and $g$ so that $(g = 0) \Rightarrow (f = 0 \vee p_{\mathbf{y}} = 0)$.
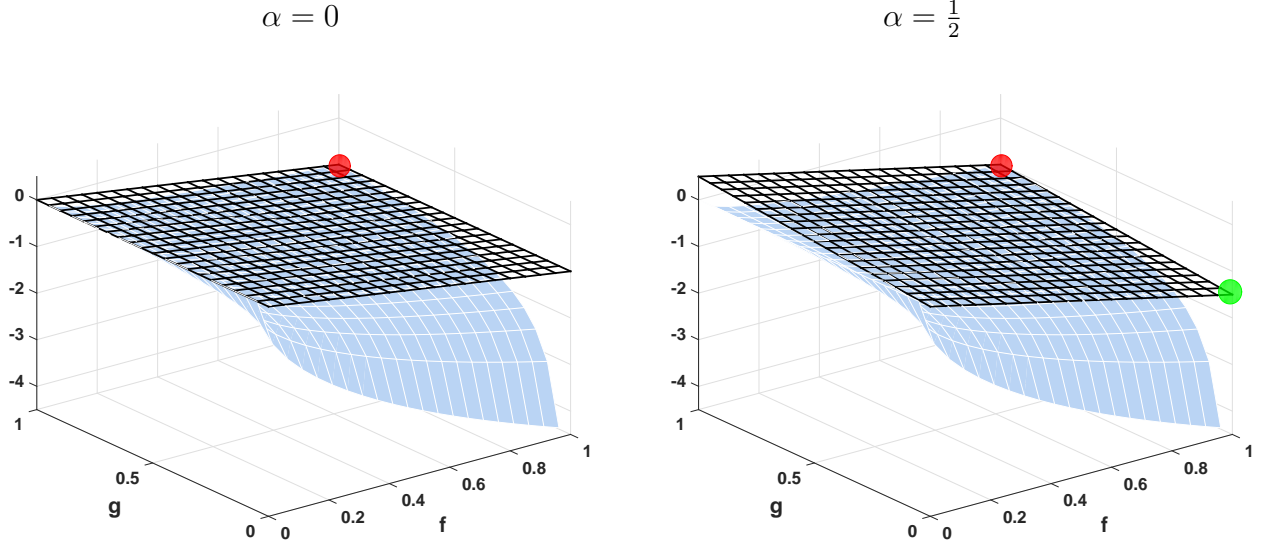
Fig. 2. The geometry of inequality (10) for $\alpha = 0$ (left) and $\alpha = \frac{1}{2}$ (right). The approximating plane (upper bound) is depicted by the wireframe mesh whereas the solid blue surface corresponds to function $-\sqrt{\frac{f}{g}}$. The red dots at (1, 1, -1) correspond to the tightness condition in (12) when $\mathbf{x} = \mathbf{y}$ (specifically, $f = g = 1$). Notice that the neighborhood of the green dot at $(1, 0, -1.5)$ corresponds to lower values of $-\sqrt{\frac{f}{g}}$ and, therefore, lower values of the negative Bhattacharyya coefficient.

*Proof:* See appendix. ∎

Fig. 2 illustrates the geometry of inequality (10), with the upper bound corresponding to $\alpha = 0$ on the left side and the upper bound corresponding to $\alpha = \frac{1}{2}$ on the right side.

If we choose $f$ and $g$ to be within interval $[0, 1]$, then Lemma 1 yields an upper bound on $B(\mathbf{x}|q)$ for some fixed $\mathbf{y}$:

$$B(\mathbf{x}|q) \leq -\int_C \sqrt{p_{\mathbf{y}}(c)q(c)}\Big((1+\alpha)f(c, \mathbf{x}, \mathbf{y}) - \alpha g(c, \mathbf{x}, \mathbf{y})\Big)dc \tag{11}$$

To obtain an auxiliary function that satisfies (6), the choice of $f$ and $g$ should ensure that bound (10) is tight when $\mathbf{x} = \mathbf{y}$, i.e., we should have:

$$\sqrt{\frac{f(c, \mathbf{x}, \mathbf{x})}{g(c, \mathbf{x}, \mathbf{x})}} = (1+\alpha)f(c, \mathbf{x}, \mathbf{x}) - \alpha g(c, \mathbf{x}, \mathbf{x}) \tag{12}$$

We propose the following choices for $f$ and $g$:

$$f(c, \mathbf{x}, \mathbf{y}) = \frac{\sum_i x_i K_i(c)}{\sum_i y_i K_i(c)}$$
$$g(c, \mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x}|}{|\mathbf{y}|} \tag{13}$$

It is easy to verify that these satisfy both the multiplicative form in (8) and also the tightness condition in (12) when $\mathbf{x} = \mathbf{y}$ (specifically, $f = g = 1$, which corresponds to the red dots at (1, 1, -1) in Fig. 2). Plugging this choice of $f$ and $g$ into the upper bound in (11) yields the following auxiliary function

$\tilde{B}(\mathbf{x}, \mathbf{y}|q)$ for the negative Bhattacharyya coefficient $B(\mathbf{x}|q)$ at any $\mathbf{x} \leq \mathbf{y}$ :

$$
\begin{aligned}
\tilde{B}(\mathbf{x}, \mathbf{y}|q) &= -\int_C \sqrt{p_{\mathbf{y}}(c)q(c)}\Big((1+\alpha)\frac{\sum_i x_i K_i(c)}{\sum_i y_i K_i(c)} - \alpha\frac{|\mathbf{x}|}{|\mathbf{y}|}\Big)dc \\
&= \sum_i \Big[-(1+\alpha)\int_C \sqrt{p_{\mathbf{y}}(c)q(c)}\frac{K_i(c)}{\sum_i y_i K_i(c)}dc - \alpha\frac{B(\mathbf{y}|q)}{|\mathbf{y}|}\Big] x_i \\
&= \underbrace{(1+\alpha)B(\mathbf{y}|q)}_{\text{Constant}} + \underbrace{\sum_i \Big[\frac{(1+\alpha)y_i}{|\mathbf{y}|}\int_C \sqrt{\frac{q(c)}{p_{\mathbf{y}}(c)}}K_i(c)dc\Big](1 - x_i)}_{\text{background summation}} + \underbrace{\sum_i \Big[\frac{-\alpha B(\mathbf{y}|q)}{|\mathbf{y}|}\Big] x_i}_{\text{foreground summation}}
\end{aligned}
$$

$$(14)$$

For a fixed $\mathbf{y}$, equation (14) is a *modular* (linear) function of binary variables $x_i$. We have arranged the expression as the sum of a constant (independent of $\mathbf{x}$) and two summations of *unary* coefficients, one over the background region and the other over the foreground. Notice that these unary coefficients are independent of binary variable $\mathbf{x}$. They depend only on fixed $\mathbf{y}$. Condition $\mathbf{x} \leq \mathbf{y}$ can be enforced by adding a very large constant to the unary coefficient of each $i$ within the foreground's summation if $i$ verifies $y_i = 0$.

This development leads us to the following proposition:

**Proposition 1.** For any $\alpha \in [0, \frac{1}{2}]$, function $\tilde{B}$ satisfies

$$B(\mathbf{x}|q) \leq \tilde{B}(\mathbf{x}, \mathbf{y}|q) \quad \forall \mathbf{x} \leq \mathbf{y} \tag{15a}$$

$$B(\mathbf{x}|q) = \tilde{B}(\mathbf{x}, \mathbf{x}|q) \tag{15b}$$

and, therefore, $\tilde{B}(\mathbf{x}, \mathbf{y}|q)$ is an auxiliary function of $B(\mathbf{x}|q)$ at $\mathbf{y}$ for $\mathbf{x} \leq \mathbf{y}$.

*Proof:* The bound condition in (15a) follows directly from the result we obtained in (11). Also, the tightness condition (12), which can be easily verified for our choice of $f$ and $g$, proves (15b). ∎

If $\tilde{B}(\mathbf{x}, \mathbf{y}|q)$ is an auxiliary function of $B(\mathbf{x}|q)$, it is straightforward to see that $\tilde{E}(\mathbf{x}, \mathbf{y}|q) = \tilde{B}(\mathbf{x}, \mathbf{y}|q) + \lambda S(\mathbf{x})$ is an auxiliary function of $E(\mathbf{x}|q)$. Following equation (7), the main step of our algorithm is:

$$\mathbf{y}^{(t+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \tilde{E}(\mathbf{x}, \mathbf{y}^{(t)}|q) \tag{16}$$

A pseudo-code of the algorithm is given in *Algorithm* 1. Since $S(\mathbf{x})$ is submodular and $\tilde{B}(\mathbf{x}, \mathbf{y}|q)$ is modular in $\mathbf{x}$, then auxiliary function $\tilde{E}(\mathbf{x}, \mathbf{y}|q)$ is submodular in $\mathbf{x}$. In combinatorial optimization, a global optimum of such submodular functions can be computed efficiently in low-order polynomial time with a single graph cut by solving an equivalent max-flow problem; In this work, we use the max-flow algorithm of Boykov and Kolmogorov [36].

**Role of parameter $\alpha$:** In this section, we give an interpretation to parameter $\alpha$ using Fig. 2, which illustrates the geometry of inequality (10). Let us first consider the case $\alpha = 0$ depicted by the left side of Fig. 2. In this case, the minimum of the approximating-plane function (i.e., the upper bound depicted by the wireframe mesh) occurs at the red dot at (1, 1, -1). Specifically, $f = g = 1$, i.e., $\mathbf{x} = \mathbf{y}$. This means that the new segmentation obtained at the current iteration is similar to the segmentation recorded at the previous iteration. For $\alpha > 0$, which we illustrate by the right side of the figure for $\alpha = \frac{1}{2}$, the minimum of the upper bound occurs in the neighborhood of the green dot at $(1, 0, -1.5)$. Notice that such a neighborhood corresponds to lower values of $-\sqrt{\frac{f}{g}}$ and, therefore, lower values of the negative Bhattacharyya coefficient; See the lower surface of the figure. In fact, $\alpha$ controls the slope of the upper-bound plane; the higher $\alpha$, the steeper the slope. More importantly, for low values of the negative Bhattacharyya coefficient (Specifically, when function $f$ and $g$ are close to the coordinates of the green dot), increasing $\alpha$ tighten the gap between the bound and the original function. Therefore, when

---

**Algorithm 1:** Finding a region consistent with a model

---

1) Iter. $t = 0$:
   - a) Initialize the fixed labeling to $\mathbf{y}^{(0)}$
   - b) Set $\alpha = \alpha_0 \geq 0$
2) Repeat the following steps until convergence:
   - a) Update the current labeling by optimizing the auxiliary function over $\mathbf{x}$ via a graph cut:

$$\mathbf{y}^{(t+1)} = \underset{\mathbf{x}:\mathbf{x}\leq\mathbf{y}^{(t)}}{\operatorname{argmin}} \quad \tilde{E}(\mathbf{x}, \mathbf{y}^{(t)})$$

   - b) If $\alpha \leq \frac{1}{2}$, go to step d)
   - c) If $\alpha > \frac{1}{2}$ (This step is necessary only when $\alpha_0 > \frac{1}{2}$)
     - If the actual energy does not increase, i.e., $E(\mathbf{y}^{(t+1)}) \leq E(\mathbf{y}^{(t)})$:
       - Go to step d)
     - If the actual energy increases, i.e., $E(\mathbf{y}^{(t+1)}) > E(\mathbf{y}^{(t)})$:
       - Decrease $\alpha : \alpha \leftarrow \rho\alpha$, with $\rho \in [0, 1[$
       - Return to step a)
   - d) $t \leftarrow t + 1$

---

$\alpha$ increases, the bound yields a better approximation of the energy. In other words, higher values of $\alpha$ favor lower values of the negative Bhattacharyya coefficient. Consequently, when we have a strict upper bound ($\alpha \in [0, \frac{1}{2}]$), one expect that $\alpha = \frac{1}{2}$ yields the best solution; We will confirm this experimentally. *In summary, $\alpha$ controls the quality of the approximation for low values of the negative Bhattacharyya coefficient: the higher $\alpha$, the better the approximation.* Recall that one cannot increase arbitrary $\alpha$ as a value of $\alpha > \frac{1}{2}$ does not guarantee anymore that the energy does not increase within each iteration. However, one can see from Fig. 2 that, up to some values of $\alpha > \frac{1}{2}$, most of the blue surface still lies below the upper-bound plane, even though we do not have a strict bound anymore. Therefore, it is natural to introduce in *Algorithm* 1 additional optional steps, which guarantee that the energy does not increase even for an initial choice of $\alpha$ bigger than $\frac{1}{2}$ (Steps 2.c in *Algorithm* 1). These steps allow to choose the best trade off between approximation quality and optimality guarantee; we will confirm experimentally the benefits of such steps. Starting from an $\alpha > \frac{1}{2}$, we verify whether the bound optimization did not increase the energy at the current iteration, i.e., $E(\mathbf{y}^{(t+1)}) \leq E(\mathbf{y}^{(t)})$. If this is the case, we accept the obtained solution and proceed to next iteration $t + 1$, while keeping the same $\alpha > \frac{1}{2}$. Otherwise, we reject the obtained solution and re-optimize the auxiliary function at iteration $t$, but with a smaller value of $\alpha$.

### B. Co-segmentation of image pairs

The co-segmentation problem amounts to finding the same foreground region in a pair of images. It has attracted an impressive research effort recently [11], [13], [9], [10]. Let $\mathbf{I}^1$ and $\mathbf{I}^2$ be two given images. The purpose is to simultaneously segment these images so that the foreground regions have consistent image distributions and smooth boundaries. We formulate the problem as the optimization of the following cost function with respect to two binary variables $\mathbf{u}$ and $\mathbf{v}$, the first encoding a segmentation of $\mathbf{I}^1$ and the second a segmentation of $\mathbf{I}^2$:

$$F(\mathbf{u}, \mathbf{v}) = \underbrace{B(\mathbf{u}|p_{\mathbf{v}})}_{\text{Co-segmentation}} + \underbrace{\lambda\{S(\mathbf{u}) + S(\mathbf{v})\}}_{\text{Regularization}} \tag{17}$$

We adopt an iterative two-step algorithm, with functional $F$ decreasing at each step: the first step fixes $\mathbf{u}$ and minimizes $F$ with respect to $\mathbf{v}$, whereas the second step seeks an optimal $\mathbf{u}$ with $\mathbf{v}$ fixed. Both

steps amount to finding a region consistent with a fixed model distribution and, therefore, can be solved using *Algorithm* 1 presented in the previous section. The principle steps of the proposed co-segmentation procedure are summarized in *Algorithm* 2.

---

**Algorithm 2:** Co-segmentation of image pairs

1) Iter. $l = 0$. Initialize $\mathbf{u}, \mathbf{v}$: $u_i = v_i = 1 \quad \forall i$.
2) Repeat the following steps until convergence.

a) Fix $\mathbf{u}$ and update $\mathbf{v}$ with *Algorithm* 1 as follows:

$$\mathbf{v}^{(l+1)} = \operatorname*{argmin}_{\mathbf{v}} \; F(\mathbf{u}^{(l)}, \mathbf{v})$$

$$= \operatorname*{argmin}_{\mathbf{v}} \; E(\mathbf{v}|p_{\mathbf{u}^{(l)}})$$

b) Fix $\mathbf{v}$ and update $\mathbf{u}$ with *Algorithm* 1 as follows:

$$\mathbf{u}^{(l+1)} = \operatorname*{argmin}_{\mathbf{u}} \; F(\mathbf{u}, \mathbf{v}^{(l+1)})$$

$$= \operatorname*{argmin}_{\mathbf{u}} \; E(\mathbf{u}|p_{\mathbf{v}^{(l+1)}})$$

c) $l \leftarrow l + 1$

---

### C. Image segmentation with a user-provided bounding box

In this section, we extend *Algorithm* 1 to interactive image segmentation. In this case, the model distribution is not assumed known (or fixed). Given a user-provided box bounding the foreground region (cf. the examples in Fig. 11), the background model is updated iteratively along with the segmentation process and is used to find a two-region partition of the image domain. Let $\mathbf{I}$ denote a given image and $\mathbf{R}_{bounding}$ the region (image sub-domain) within the bounding box. A segmentation of $\mathbf{R}_{bounding}$ can be identified by binary labeling $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, with $x_i = 1$ indicating that pixel $i$ belongs to the target region (foreground) and $x_i = 0$ indicating background membership ($N$ is the number of pixels within $\mathbf{R}_{bounding}$). The algorithm consists of optimizing with respect to $\mathbf{x}$ a sequence of cost functions of the following form:

$$G(\mathbf{x}|\mathcal{N}^{(k-1)}) = B(\bar{\mathbf{x}}|\mathcal{N}^{(k-1)}) + \lambda S(\mathbf{x}), \tag{18}$$

where $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N)$, with $\bar{x}_i = 1 - x_i$. Model distribution $\mathcal{N}^{(k-1)}$ is a variable, which has to be updated along with the segmentation. Given an initial $\mathcal{N}^{(0)}$ learned from image data outside the bounding box (i.e., within a region in $\Omega \setminus \mathbf{R}_{bounding}$, where $\Omega$ denotes the image domain), the algorithm iterates two steps. One step seeks in $\mathbf{R}_{bounding}$ a background region consistent with current model $\mathcal{N}^{(k-1)}$ and is similar to *Algorithm* 1, whereas the other step refines the model using the current segmentation. An illustration of this two-step algorithm is depicted in Fig. 3. The principle steps of the proposed interactive segmentation algorithm are summarized in *Algorithm* 3.

Let us examine the iterative behavior of *Algorithm* 3. At the first iteration ($k = 1$), model $\mathcal{N}^{(0)}$ is learned from outside the bounding box. Because the optimization in (18) seeks a relevant region inside the bounding box, the initial labeling is guaranteed to change. At iteration ($k > 2$), we have two cases:

1) $\mathcal{N}^{(k-1)}$ *matches perfectly* $\mathcal{N}^{(k-2)}$: This causes the algorithm to converge because the energies optimized at the current and previous iterations are the same.

2) $\mathcal{N}^{(k-1)}$ *does not exactly match* $\mathcal{N}^{(k-2)}$: In this case, the energy is updated and so is the labeling. As illustrated by the example in Fig. 10, this happens in a few iterations (typically less than 10). In fact, the optimization in (18) ensures only a best possible match between $\mathcal{N}^{(k-1)}$ and $\mathcal{N}^{(k-2)}$, not an exact match. Also, the smoothness constraint influences the solution and, therefore, can cause $\mathcal{N}^{(k-1)}$ to deviate

---

**Algorithm 3:** Segmentation with a user-provided bounding box

---

1) Iter. $k = 0$:
   a) Initialize $\mathbf{R}_{bounding}$ with a user-provided bounding box.
   b) Compute $\mathcal{N}^{(0)}$ with image data outside $\mathbf{R}_{bounding}$, using for instance the distribution of $\mathbf{I}$ in a strip of width $w$ around the bounding box.
2) For each iter. $k, k = 1, 2, \ldots$, repeat the following steps until convergence.
   a) Update $\mathbf{x}$ at iteration $k$ with *Algorithm* 1 as follows

   $$\mathbf{x}^{(k)} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad G(\mathbf{x}|\mathcal{N}^{(k-1)})$$

   b) Update the background model at iteration $k$ as follows:

   $$\mathcal{N}^{(k)}(c) = \frac{\sum_i \bar{x}_i K_i(c)}{|\bar{\mathbf{x}}|} \tag{19}$$

---

slightly from $\mathcal{N}^{(k-2)}$. During a few iterations, the background model is updated, thereby approaching the distribution of image data in the neighborhood of the target-region boundaries.

It is worth noting that *Algorithm* 3 has an important advantage over optimizing pixelwise image-likelihood functions, as is common in the existing interactive segmentation algorithms [20], [18], [17]. These algorithms require learning both foreground and background models. *Algorithm* 3 relaxes the need for estimating the foreground model and, therefore, is less prone to errors in estimating model distributions.
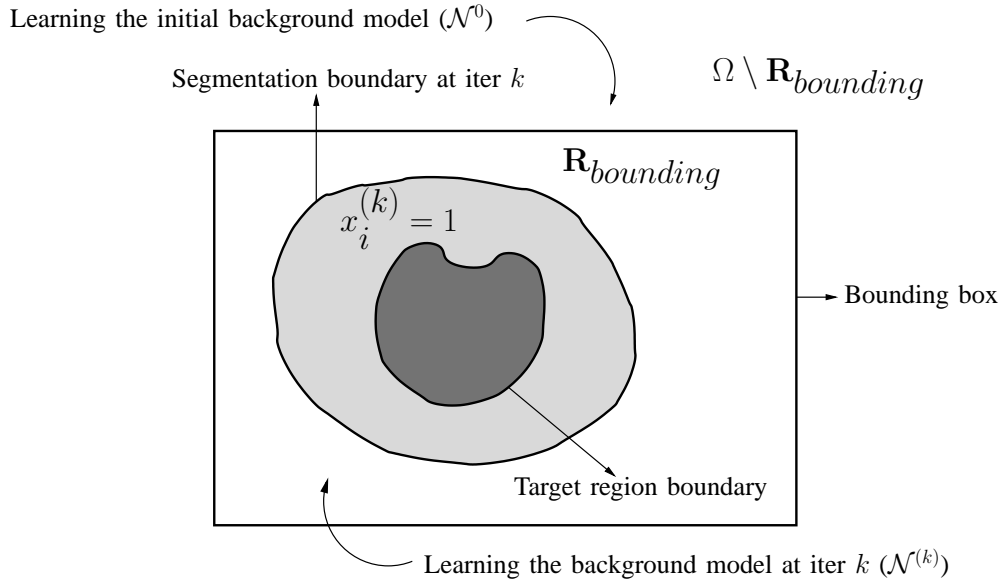


Fig. 3. Illustration of the two-step segmentation with a user-provided bounding box (*Algorithm* 3): background model $\mathcal{N}^{(k)}$ is refined iteratively with the binary labeling.

## III. EXPERIMENTS

This section contains three parts, each describing an evaluation of one of the three proposed algorithms. For all the experiments, the photometric variable is color specified in RGB coordinates.

*A. Finding a region consistent with a fixed model distribution (Algorithm 1)*

*1) Examples:* Fig. 4 shows examples of segmentations where the training and testing images are different but depict the same type of target regions. Each row in the figure corresponds to an example of target regions. The target-region categories are very diverse, including animals, cars, monuments and humans in sport scenes. These images were obtained from the iCoseg database introduced in the work of Batra et al. [2]. Given a model learned from a manual delineation of the target region in a single training image, we show how *Algorithm* 1 can delineate different instances of the target region in several other images. The training image and its manual segmentation are shown in the first column of each row. The rest of the columns shows the segmentation obtained with *Algorithm* 1. In these examples, the color distributions, shapes and sizes of the target objects do not match exactly. The target object undergoes significant variations in shape/size in comparison to the learning image, which precludes the use of shape priors to drive the segmentation process. *Algorithm* 1 handles implicitly these variations because no assumptions were made as to the size, shape, or position of the target object. Furthermore, in some cases, the background regions are cluttered and are significantly different from the training-image background. For these scenarios, using a background model as in standard likelihood-based methods [21], [20], [18], [17] would not be helpful. For this set of experiments, we used the following parameters: $\alpha = 0.5$; $\lambda = 1 \times 10^{-4}$, with standard spatial distance pairwise weights [19] and a 4-connected grid. A 3-dimensional histogram based on $96^3$ bins was used as a distribution.

*2) Effect of $\alpha$:* Fig. 5 depicts typical results, which demonstrate the effect of the initial value of $\alpha$ on the obtained solutions. We run several tests and plotted (i) the energy obtained at convergence as a function of the initial value of $\alpha$ (first row, right side); and (ii) the evolution of the energy as a function of the iteration number for different values of $\alpha_0$ (first row, left side). Notice that the energy at convergence is a monotonically decreasing function of $\alpha$, but becomes almost constant starting from some value of $\alpha$ ($\alpha \approx 1$). As expected, for $\alpha \in [0, \frac{1}{2}]$ (i.e., when we have a strict upper bound), $\alpha = \frac{1}{2}$ yielded the best solution (lowest energy). These results are consistent with the interpretation we gave earlier to $\alpha$: $\alpha$ controls how well the bound approximates the energy for high values of the Bhattacharyya coefficient; the higher $\alpha$, the better the approximation. We also observe that $\alpha > \frac{1}{2}$ can improve slightly the obtained solutions but, starting from some value of $\alpha$, the performance of *Algorithm* 1 remains approximately the same. This confirms the benefits of the additional optional steps that we added to our algorithm, and is expected. As discussed earlier, one can see from Fig. 2 that, up to some $\alpha > \frac{1}{2}$, most of the blue plane still lies below the surface, even though it is not a strict lower bound anymore. The second row of Fig. 5 depicts the images we used in this set of tests. The training image and its manual segmentation are shown in the first column of the figure. From the second to fifth column, we show the segmentations obtained for different values of $\alpha_0$. For this set of experiments, we used the following parameters along with standard contrast-sensitive pairwise weights [20] and an 8-connected grid: $\lambda = 1 \times 10^{-4}$; Number of bins: $92^3$; $\rho = 0.8$.

*3) Quantitative evaluations and comparisons in regard to optimality and computational load:* We carried out quantitative evaluations and comparisons on the Microsoft GrabCut database [9], which contains 50 images with ground truth segmentations. This subset of experiments compare the proposed bound optimizer to fast trust region [53], which is an iterative graph cut optimization technique recently proposed to tackle non-linear segmentation functionals. The purpose is to evaluate each optimization technique in regard to solution optimality and computation load. Therefore, similarly to the experiments in [53], [30], [9], we used the ground truth distribution as a target. For each of the algorithms, we computed the following performance measures: (i) the energy obtained at convergence, (ii) the number of graph cuts required to reach convergence and (iii) the average error, i.e., percentage of misclassified pixels in comparison to the ground truth. For both algorithms, we used the same initialization (the initial foreground segment is the whole image domain) and parameters along with standard spatial-distance pairwise weights and an 8-connected grid: $\lambda = 1 \times 10^{-5}$; Number of bins: $92^3$. For *Algorithm* 1, we used $\alpha_0 = 5$ and $\rho = 0.8$. Table I reports the statistics of all the performance measures over the GrabCut data, indicating that *Algorithm* 1
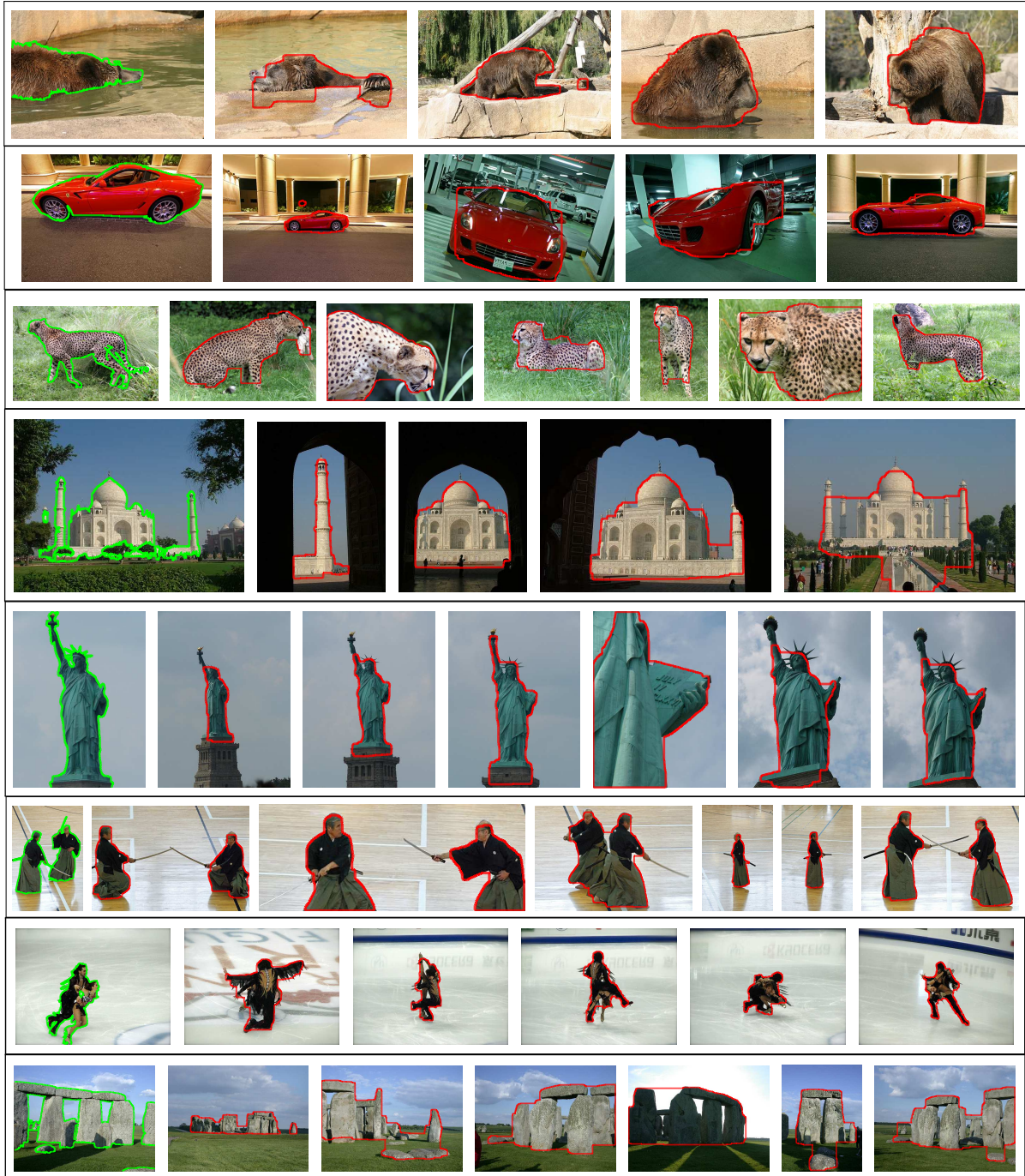
Fig. 4. Examples of segmentations for various target-region categories, including animals, cars, monuments and humans in sport scenes. Given a model learned from a training image in the first column (A manual segmentation is depicted by the green curve), objects of the same category are obtained with *Algorithm* 1 in several other images (red curves). In these examples, the color distributions, shapes and sizes of the target objects do not match exactly. The target objects undergo substantial variations in shape/size in comparison to the learning images, which precludes the use of shape priors to drive the segmentation process. Furthermore, in some cases, the background regions are cluttered and are significantly different from the training-image backgrounds. For these scenarios, the standard log-likelihood criterion, which requires a reliable background model, is not applicable.
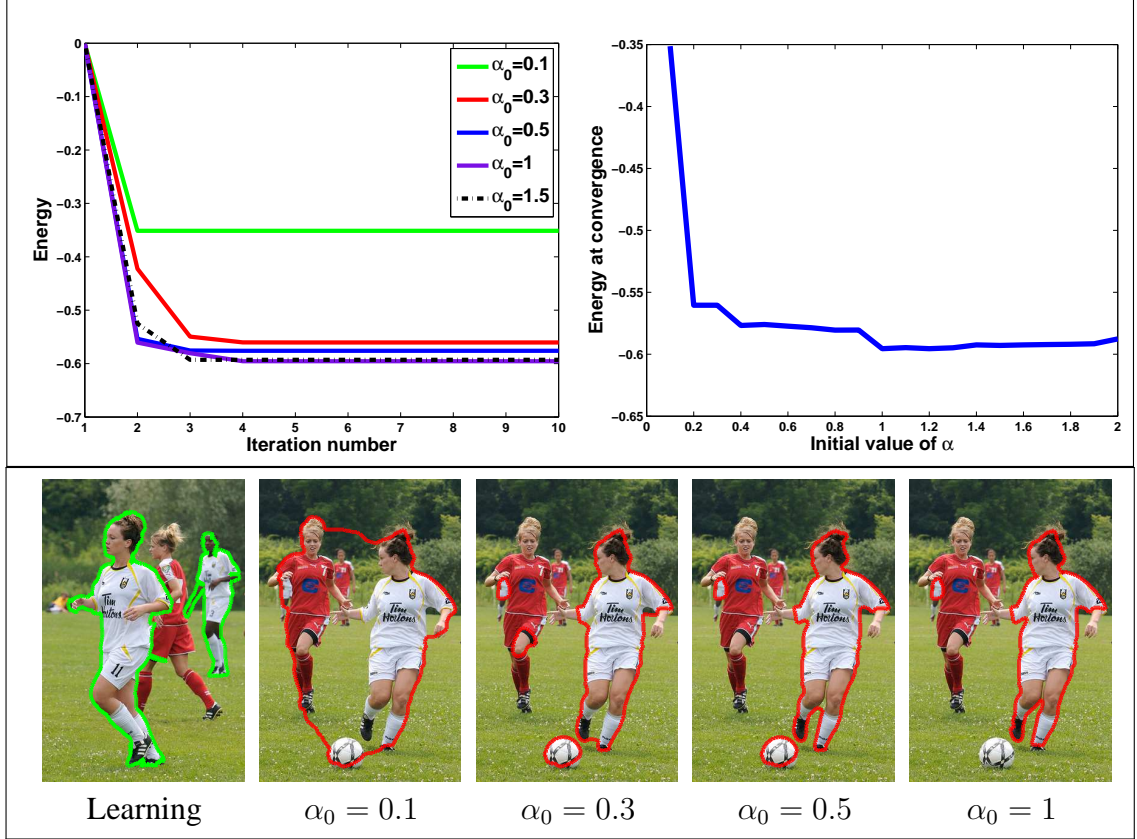
Fig. 5. Illustration of the effect of $\alpha$ on the solutions obtained by *Algorithm* 1. First row, right side: the energy obtained at convergence as a function of the initial value of $\alpha$; First row, left side: the evolution of the energy as a function of the iteration number for different values of $\alpha_0$. Second row, first column: The training image and its manual segmentation. Second row, columns from second to fifth: The segmentations obtained for different values of $\alpha_0$.

TABLE I
*Comparisons over the GrabCut data set of the proposed bound optimizer (Algorithm 1) with the fast trust region optimization in [53].*

| Method | Bound optimization (*Algorithm 1*) | *Fast Trust Region [53]* |
|---|---|---|
| Average energy | $-0.9518$ | $-0.9247$ |
| Number of lower energies | 35 | 15 |
| Number of graph cuts (Median, Min, Max) | $(4, 3, 8)$ | $(77, 5, 911)$ |
| Average error | $1.32\%$ | $3.43\%$ |

yields a competitive performance in regard to optimality and speed. In particular, the proposed algorithm obtained lower energy values for $35$ out of the $50$ images while requiring a much lower number of graph cuts. Fig. 6 plots for both algorithms the energy and error at convergence versus the image number.

*4) Failure cases:* Fig. 7 depicts examples of failure of *Algorithm* 1. The first, second and third columns show three instances where *Algorithm* 1 did not succeed to fully detect the target region, given the model learned from the image in the first column. These failures are due to the fact that the Bhattacharyya measure is sensitive to significant variations in color distributions between the learning and testing images. Such variations occur with illumination changes, for instance.

## B. Co-segmentation of image pairs (Algorithm 2)

Fig. 8 illustrates the iterative behavior of the co-segmentation algorithm on a pair of bear images, where the target regions have completely different shapes and sizes. The first three columns depict the images and segmentation boundaries at each iteration, whereas the last two columns display the foreground regions
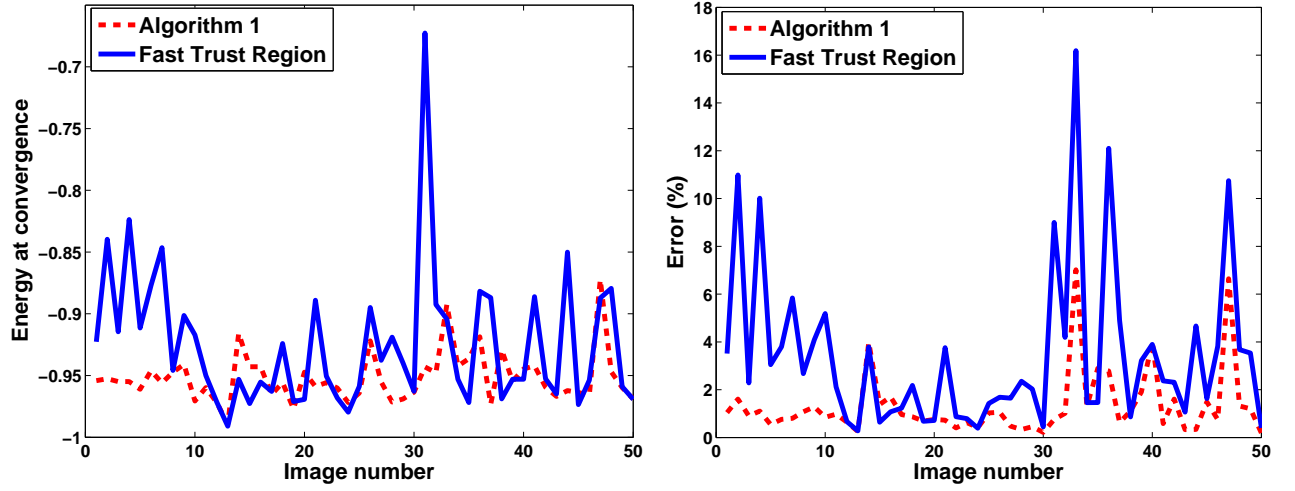
Fig. 6. Comparisons of *Algorithm* 1 with the fast trust region optimization in [53]. Left: Energy at convergence versus the image number; Right: Error at convergence versus the image number.
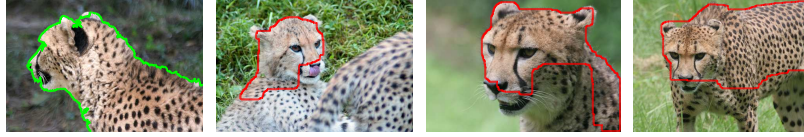


Fig. 7. Examples of failure of *Algorithm* 1. The first column shows the learning image; the rest of the images show three instances where *Algorithm* 1 did not succeed to fully detect the target region. The parameters are $\alpha = 0.5$ and $\lambda = 1 \times 10^{-4}$. We used standard spatial distance pairwise weights [19] and a 4-connected grid. A 3-dimensional histogram based on $96^3$ bins was used as a distribution.

obtained at convergence. At the first iteration, the obtained foregrounds included significant parts from the backgrounds because the initial model distribution was computed over the whole domain of one of the images. Then, *Algorithm* 2 refined the solution at convergence because the model distributions were updated iteratively along with the segmentations.

Fig. 9 depicts several other co-segmentation examples, which illustrate the effectiveness of *Algorithm* 2. For each example, we show the segmentation boundaries and foreground regions obtained at convergence. It is worth noting that, in some of these examples, the foreground regions have significantly different sizes. The proposed co-segmentation algorithm can handle implicitly such variations in the size of the target regions, without the need additional optimization/priors with respect to region size. This is due to the fact that the Bhattacharyya measure does not constrain the target regions to be of equal sizes, which is an important advantage over the co-segmentation models in [13], [9]. Based on *unnormalized* histograms,
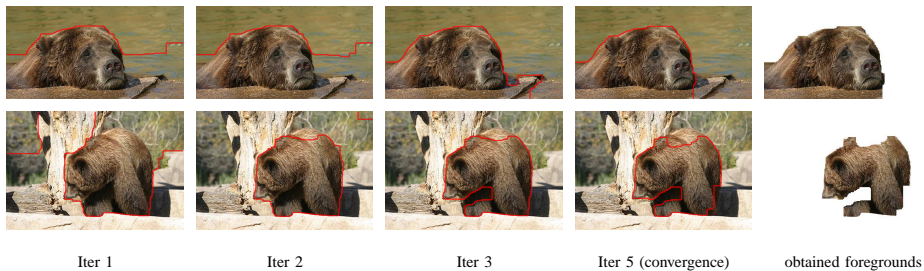


Fig. 8. The iterative behavior of the proposed co-segmentation algorithm (*Algorithm* 2). A 3-dimensional histogram based on $32^3$ bins was used as a distribution. The parameters are $\lambda = 3 \times 10^{-4}$, $\rho = 0.8$ and $\alpha_0 = 5.8$, used in conjunction with standard spatial distance pairwise weights [19] and a 4-connected grid.

Fig. 9. Examples of co-segmentations with *Algorithm* 2. For each example, we show the segmentation boundaries and foreground regions obtained at convergence. A 3-dimensional histogram based on $32^3$ bins was used as a distribution. The parameters are $\lambda = 3 \times 10^{-4}$, $\rho = 0.8$ and $\alpha_0 = 5.8$.

the models in [13], [9] assume the foreground regions have the same size. In these examples, we used standard spatial distance pairwise weights [19] and a 4-connected grid.

*1) Quantitative evaluations and comparisons:* We carried out a quantitative accuracy evaluation of *Algorithm* 2 on the co-segmentation database introduced in [10], which includes 20 pairs of images. The experiments in [10] used this database to evaluate several co-segmentation models, including the $L_1$ model in [9], the $L_2$ model in [13] and the dot product model in [11], as well as several optimization techniques, including trust region graph cut [9] and dual decomposition [10], among others. Due to the difficulty of obtaining a ground truth for co-segmentation, the data is based on composites of 40 different backgrounds with 20 foregrounds. We followed the same experimental setting as [10]: we run *Algorithm* 2 not only on the original images but also on foreground regions of different sizes by rescaling one of the images to 70, 80, 90 and 200 percent of the original size. In particular, [10] showed that the performances of [13], [9] degrade when the foreground regions have different sizes. Table II lists the average errors for *Algorithm* 2 and the errors reported in the comparisons in [10]. Except [11], all the methods yielded approximately the same performance for the original images. However, when the foreground regions have different sizes, the accuracies of [13], [9] degraded significantly. On the contrary, the performance of *Algorithm* 2 is stable. For more difficult co-segmentation examples where the foregrounds have different sizes, the models in [13], [9] remove incorrectly some parts of the foregrounds [10]. Based on unnormalized histograms, these models assume the foreground regions have the same size. *Algorithm 2* handles accurately variations in the size of the target regions, and does so implicitly, i.e., without additional optimization/priors with respect to region size. This is an important advantage over the models in [13], [9]. For this set of experiments,

TABLE II
*Accuracy evaluations on the co-segmentation database introduced in [10]: average error for* Algorithm *2 and the co-segmentation algorithms in [9], [13], [11]. The performances of [9], [13], [11] were reported in the experimental-comparison study in [10].*

| Method | *Algorithm 2* | *Method in [9]* | *Method in [13]* | *Method in [11]* |
|---|---|---|---|---|
| average error (original) | 2.8% | 3.2% | 2.9% | 8.8% |
| average error (Re-sized to 90%) | 2.9% | 4.2% | 4% | 8.1% |
| average error (Re-sized to 80%) | 2.6% | 5.2% | 6.2% | 7% |
| average error (Re-sized to 70%) | 2.8% | – | – | – |
| average error (Re-sized to 200%) | 3.3% | – | – | – |

TABLE III
*Accuracy evaluations on the GrabCut database: average error for* Algorithm *3 and two other algorithms optimizing the image log-likelihood cost, one based on Dual Decomposition (DD) [18] and the other on Expectation-Maximization (EM) [20].* $\lambda = 1.5 \times 10^{-4}$*; Number of bins:* $96^3$*;* $\alpha_0 = 5$*;* $\rho = 0.8$*. The initial background model is estimated from the image within a strip of width* 20 *pixels around the bounding box.*

| Method | *Algorithm 3* | DD + image likelihood [18] | EM + image likelihood [20] |
|---|---|---|---|
| average error | 7.49% | 10.5% (reported in [18]) | 8.1% (reported in [18]) |
| Run time (seconds) | 14.03 | 576 (reported in [18]) | – |

we used the following parameters along with standard contrast-dependent pairwise weights [20] and a 16-connected grid: $\lambda = 1 \times 10^{-4}$; Number of bins: $32^3$; $\alpha_0 = 5$; $\rho = 0.8$.

## C. Interactive segmentation with a user-provided bounding box (Algorithm 3)

*1) Quantitative evaluations:* We carried out quantitative and comparative evaluations of *Algorithm* 3 on the Microsoft GrabCut database [9], which contains 50 images with ground truth segmentations. Each image comes with a bounding box that has been automatically computed from the ground truth [18]. Similar experiments on the same data[4] were reported in [18] to evaluate two other algorithms optimizing the log-likelihood cost, one based on Dual Decomposition (DD) optimization [18] and the other on an Expectation-Maximization (EM) procedure [20]. The third and fourth rows of Fig. 11 depicts two examples from the GrabCut data. The error is computed as the average percentage of misclassified pixels inside the bounding box. The errors reported in Table III demonstrate that *Algorithm* 3 can yield a competitive performance in the context of interactive image segmentation. For this quantitative evaluation, the parameters of *Algorithm* 3 were fixed for all the images in the data set as follows: $\lambda = 1.5 \times 10^{-4}$; Number of bins: $96^3$; $\alpha_0 = 5$; $\rho = 0.8$. The initial background model is estimated from the image within a strip of width 20 pixels around the bounding box. We used standard spatial distance pairwise weights [19] and a 4-connected grid.

*2) Examples:* In this section, we show several examples that illustrate how *Algorithm* 3 can delineate target foreground regions using only a bounding box. The typical example in Fig. 10 illustrates the fast convergence of *Algorithm* 3. The first column depicts the image and the bounding box, whereas the remaining columns show the segmentation boundary obtained at iterations 1, 3, 5, 7, 9 and 11. Fig. 11 depicts several other examples. For each example, we show the bounding box and the segmentation boundary/foreground region obtained at convergence. Learning iteratively the background model from the current image was sufficient to accurately delineate the target regions in most of these examples. This is an important advantage over optimizing the image likelihood cost, as is common in the existing interactive segmentation algorithms [20], [18], [17]. The image likelihood requires learning both foreground and background models. *Algorithm* 3 relaxes the need for estimating the foreground model and, therefore, is less prone to errors in estimating model distributions. The last row of Fig. 11 shows a failure example where *Algorithm* 3 included a part from the background in the obtained target region. This is due to the

[4]Similar to [18], we used 49 images; the "cross" image was excluded because the bounding box corresponds to the whole image domain.
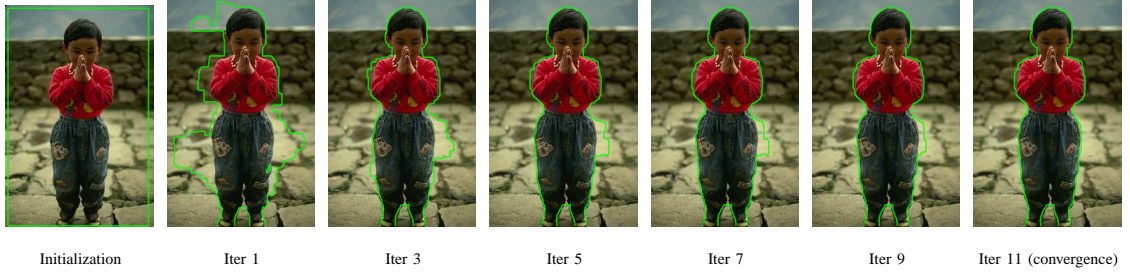
Fig. 10. An example showing the fast convergence of *Algorithm* 3. The initial background model is estimated from the image within a strip of width 10 pixels around the bounding box. $\lambda = 1.5 \times 10^{-4}$; number of bins: $96^3$; $\alpha_0 = 5.8$; $\rho = 0.8$.

similarity in color profiles between the background and the target region. In these examples, we used standard spatial distance pairwise weights [19] and a 4-connected grid.

## IV. CONCLUSION

We proposed efficient graph cut algorithms for three problems: (1) finding a region in an image, so that the distribution of image data within the region most closely matches a given model distribution; (2) co-segmentation of image pairs and (3) interactive image segmentation with a user-provided bounding box. Following the computation of an original bound of the Bhattacharyya measure, we reformulated each problem as an auxiliary function optimization via graph cuts. Various realistic examples along with quantitative and comparative evaluations demonstrated the performances, speed and flexibility of the proposed algorithms.

## APPENDIX A

In this appendix, we give a proof of Lemma 1.

*Proof:* Consider the following parametric function $H_\alpha : [0, 1] \to \mathbb{R}^+$:

$$H_\alpha(g) = \frac{1}{\sqrt{g}} + \alpha g - (1 + \alpha) \tag{A-1}$$

with $\alpha \in \mathbb{R}^+$. The first derivative of $H_\alpha$ is:

$$\frac{dH_\alpha}{dg} = -\frac{1}{2g^{\frac{3}{2}}} + \alpha \tag{A-2}$$

The second derivative of $H_\alpha$ is strictly positive:

$$\frac{d^2 H_\alpha}{dg^2} = \frac{3}{4g^{\frac{5}{2}}} > 0 \quad \forall g \in [0, 1] \tag{A-3}$$

Therefore, $\frac{dH_\alpha}{dg}$ is strictly increasing in $[0, 1]$, which yields the following inequality:

$$\frac{dH_\alpha}{dg} < \frac{dH_\alpha}{dg}(1) = \alpha - \frac{1}{2} \tag{A-4}$$

From (A-4) one can see that, $\forall \alpha \leq \frac{1}{2}$, $\frac{dH_\alpha}{dg} \leq 0$, i.e., $H_\alpha$ is strictly decreasing in $[0, 1]$. This yields:

$$\forall \alpha \in [0, \frac{1}{2}] \text{ and } \forall g \in [0, 1] \quad H_\alpha(g) \geq H_\alpha(1) = 0, \text{ i.e., } \frac{1}{\sqrt{g}} \geq 1 + \alpha - \alpha g \tag{A-5}$$
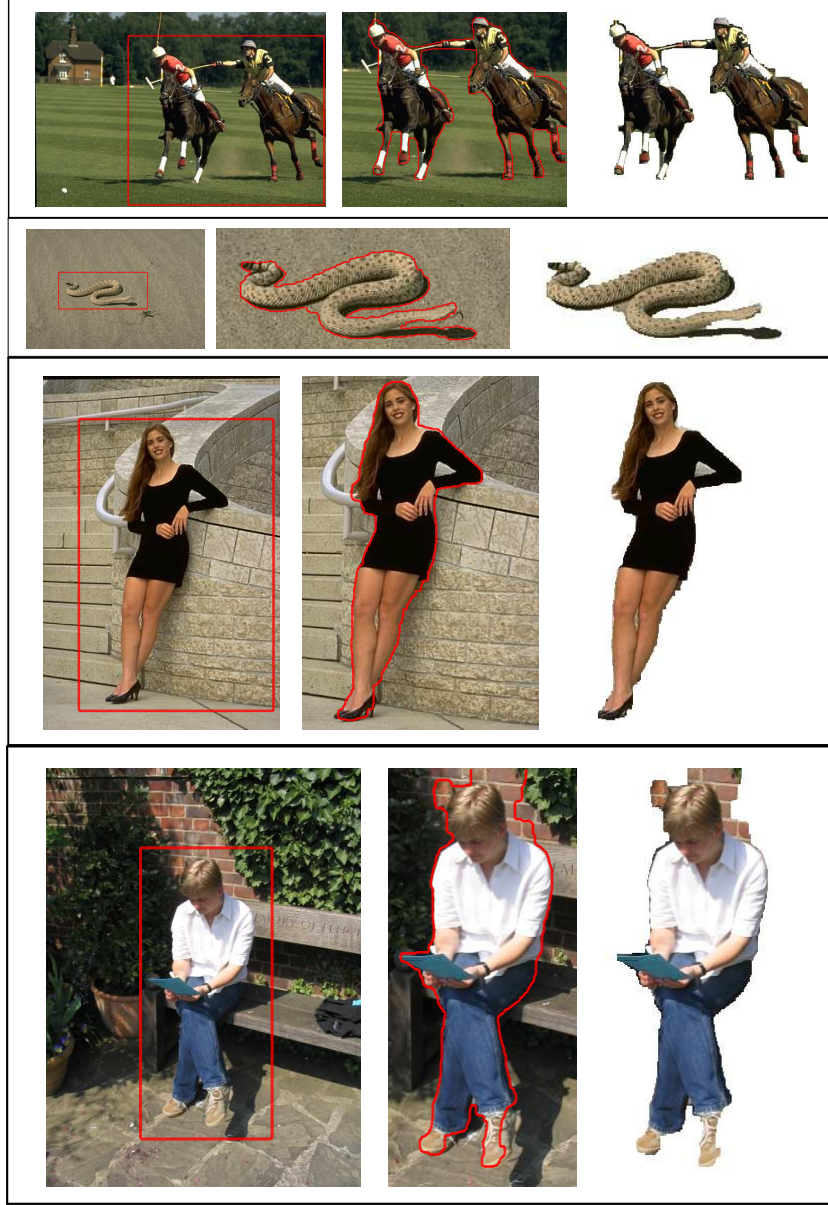
Fig. 11. Examples of segmentations with a user-provided bounding box (*Algorithm* 3). For each example, we show the bounding box and the segmentation boundary/foreground region obtained at convergence. The initial background model is estimated from the image within a strip of width 10 pixels around the bounding box. $\lambda = 1.5 \times 10^{-4}$; Number of bins: $96^3$; $\alpha_0 = 5.8$; $\rho = 0.8$.

Now notice the following inequality:

$$\sqrt{f} \geq f \quad \forall f \in [0,1] \tag{A-6}$$

Combining (A-5) and (A-6) gives:

$$\begin{aligned}
\sqrt{\frac{f}{g}} &\geq f(1 + \alpha - \alpha g) \\
&= (1+\alpha)f - \alpha f g \\
&\geq (1+\alpha)f - \alpha g
\end{aligned} \tag{A-7}$$

The last inequality in (A-7) is due to the fact that $-\alpha f g \geq -\alpha g$ (because $f \in [0,1]$). Multiplying each side in (A-7) by $-1$ and inverting the inequality proves the Lemma. ∎

## References

[1] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1605–1614.

[2] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "icoseg: Interactive co-segmentation with intelligent scribble guidance," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3169–3176.

[3] K. Punithakumar, I. Ben Ayed, I. Ross, A. Islam, J. Chong, and S. Li, "Detection of left ventricular motion abnormality via information measures and bayesian filtering," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 4, pp. 1106–1113, 2010.

[4] C. Benedek, T. Sziranyi, Z. Kato, and J. Zerubia, "Detection of object motion regions in aerial image pairs with a multilayer markovian model," *IEEE Transactions on Image Processing*, vol. 18, no. 10, pp. 2303–2315, 2009.

[5] N. Ohta, "A statistical approach to background subtraction for surveillance systems," in *IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 481–486.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[7] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[8] M. Mignotte, "A label field fusion bayesian model and its penalized maximum rand estimator for image segmentation," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1610–1624, 2010.

[9] C. Rother, T. P. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 993–1000.

[10] S. Vicente, V. Kolmogorov, and C. Rother, "Cosegmentation revisited: Models and optimization," in *European Conference on Computer Vision (ECCV)*, no. 2, 2010, pp. 465–479.

[11] D. S. Hochbaum and V. Singh, "An efficient algorithm for co-segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 269–276.

[12] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1943–1950.

[13] L. Mukherjee, V. Singh, and C. R. Dyer, "Half-integrality based algorithms for cosegmentation of images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 2028–2035.

[14] A. C. Gallagher and T. Chen, "Clothing cosegmentation for recognizing people," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[15] M. Cho, Y. M. Shin, and K. M. Lee, "Co-recognition of image pairs by data-driven monte carlo image exploration," in *European Conference on Computer Vision (ECCV)*, no. 4, 2008, pp. 144–157.

[16] J. Cui, Q. Yang, F. Wen, Q. Wu, C. Zhang, L. V. Gool, and X. Tang, "Transductive object cutout," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[17] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 277–284.

[18] S. Vicente, V. Kolmogorov, and C. Rother, "Joint optimization of segmentation and appearance models," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 755–762.

[19] Y. Boykov and G. Funka Lea, "Graph cuts and efficient n-d image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[20] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, pp. 309–314, 2004.

[21] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 105–112.

[22] I. Ben Ayed, K. Punithakumar, G. J. Garvin, W. Romano, and S. Li, "Graph cuts with invariant object-interaction priors: Application to intervertebral disc segmentation," in *Information Processing in Medical Imaging (IPMI)*, 2011, pp. 221–232.

[23] I. Ben Ayed, S. Li, and I. Ross, "A statistical overlap prior for variational image segmentation," *International Journal of Computer Vision*, vol. 85, no. 1, pp. 115–132, 2009.

[24] S. Chen and R. J. Radke, "Level set segmentation with both shape and intensity priors," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 763–770.

[25] D. Freedman and T. Zhang, "Active contours for tracking distributions," *IEEE Transactions on Image Processing*, vol. 13, pp. 518–526, 2004.

[26] I. Ben Ayed, Y. Lu, S. Li, and I. G. Ross, "Left ventricle tracking using overlap priors," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI), Part I*, 2008, pp. 1025–1033.

[27] A. Adam, R. Kimmel, and E. Rivlin, "On scene segmentation and histograms-based curve evolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1708–1714, 2009.

[28] I. Ben Ayed, H.-M. 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3288–3295.

[29] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson, "Image segmentation using active contours: Calculus of variations or shape gradients?" *SIAM Journal of Applied Mathematics*, vol. 63, no. 6, pp. 2128–2154, 2003.

[30] L. Gorelick, F. R. Schmidt, Y. Boykov, A. Delong, and A. Ward, "Segmentation with non-linear regional constraints via line-search cuts," in *European Conference on Computer Vision (ECCV)*, no. 1, 2012, pp. 583–597.

[31] O. V. Michailovich, Y. Rathi, and A. Tannenbaum, "Image segmentation using active contours driven by the bhattacharyya gradient flow," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2787–2801, 2007.

[32] H. Jiang, "Linear solution to scale invariant global figure ground separation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 678–685.

[33] I. Ben Ayed, S. Li, and I. Ross, "Embedding overlap priors in variational left ventricle tracking," *IEEE Transactions on Medical Imaging*, vol. 28, no. 12, pp. 1902–1913, 2009.

[34] A. Mitiche and I. Ben Ayed, *Variational and Level Set Methods in Image Segmentation*, 1st ed. Springer, 2010.

[35] F. Aherne, N. Thacker, and P. Rockett, "The bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 32, no. 4, pp. 1–7, 1997.

[36] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[37] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[38] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.

[39] Y. Boykov and O. Veksler, *Graph cuts in vision and graphics: Theories and applications*. Springer, 2006, ch. Handbook of Mathematical Models in Computer Vision, pp. 79–96.

[40] P. Kohli, J. Rihan, M. Bray, and P. H. S. Torr, "Simultaneous segmentation and pose estimation of humans using dynamic graph cuts," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 285–298, 2008.

[41] A. Bugeau and P. Pérez, "Detection and segmentation of moving objects in highly dynamic scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[42] J. G. Malcolm, Y. Rathi, and A. Tannenbaum, "Multi-object tracking through clutter using graph cuts," in *IEEE International Conference on Computer Vision (ICCV)*, 2007, the Workshop on Non-rigid Registration and Tracking Through Learning.

[43] V. S. Lempitsky, S. Roth, and C. Rother, "Fusionflow: Discrete-continuous optimization for optical flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[44] J. Kim, V. Kolmogorov, and R. Zabih, "Visual correspondence using energy minimization and mutual information," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 1033–1040.

[45] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, "Grabcut in one cut," in *International Conference on Computer Vision (ICCV)*, 2013.

[46] V.-Q. Pham, K. Takahashi, and T. Naemura, "Foreground-background segmentation using iterated distribution matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2113–2120.

[47] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[48] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts-a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1274–1279, 2007.

[49] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 155–225, 2002.

[50] N. Komodakis, N. Paragios, and G. Tziritas, "Mrf energy minimization and beyond via dual decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 531–552, 2011.

[51] T. Taniai, V.-Q. Pham, K. Takahashi, and T. Naemura, "Image segmentation using dual distribution matching," in *British Machine Vision Conference (BMVC)*, 2012, pp. 74.1–74.11.

[52] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Neural Information Processing Systems (NIPS)*, 2000, pp. 556–562.

[53] L. Gorelick, F. R. Schmidt, and Y. Boykov, "Fast trust region for segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1714–1721.

**Ismail Ben Ayed** received the Ph.D. degree (with the highest honor) in computer science from the National Institute of Scientific Research (INRS-EMT), University of Quebec, Montreal, QC, Canada, in May 2007. Since then, he has been a research scientist with GE Healthcare, London, ON, Canada. He is also an adjunct research professor at the University of Western Ontario. His research interests are in computer vision, optimization, machine learning and their applications in medical image analysis. He co-authored a book on image segmentation, over forty papers in reputable journals and conferences, and six patent applications. He received a GE innovation award and two NSERC fellowships. Ismail serves regularly as reviewer for the major vision and medical imaging venues.

**Kumaradevan Punithakumar** received the B.Sc.Eng. degree (with first class honors) in electronic and telecommunication engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 2001, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from McMaster University, Hamilton, Ontario, Canada, in 2003 and 2007, respectively. He worked as a research scientist at GE Healthcare, London, ON, Canada, from 2008 to 2012. Since 2012, he serves as the operational and computational director of the Servier Virtual Cardiac Centre, department of Radiology and Diagnostic Imaging, University of Alberta, Edmonton, AB, Canada. His research interests include medical image analysis, target tracking, information fusion, and computer vision.
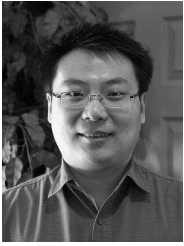
**Shuo Li** received the Ph.D. degree in computer science from Concordia University in 2006, Montreal, QC, Canada. He is currently a research scientist at GE Healthcare, London, ON, Canada. He is also an adjunct research professor at the University of Western Ontario and adjunct scientist at the Lawson Health Research Institute. He received the doctoral prize at Concordia University and several GE awards. His current research interests are in medial image analysis.