

Graph Cut Segmentation with a Global Constraint: Recovering Region Distribution via a Bound of the Bhattacharyya Measure

Ismail Ben Ayed^{1,2}, Hua-mei Chen², Kumaradevan Punithakumar¹, Ian Ross², and Shuo Li^{1,2}

¹GE Healthcare, London, ON, Canada

² University of Western Ontario, London, ON, Canada

Abstract

This study investigates an efficient algorithm for image segmentation with a global constraint based on the Bhattacharyya measure. The problem consists of finding a region consistent with an image distribution learned a priori. We derive an original upper bound of the Bhattacharyya measure by introducing an auxiliary labeling. From this upper bound, we reformulate the problem as an optimization of an auxiliary function by graph cuts. Then, we demonstrate that the proposed procedure converges and give a statistical interpretation of the upper bound. The algorithm requires very few iterations to converge, and finds nearly global optima. Quantitative evaluations and comparisons with state-of-the-art methods on the Microsoft GrabCut segmentation database demonstrated that the proposed algorithm brings improvements in regard to segmentation accuracy, computational efficiency, and optimality. We further demonstrate the flexibility of the algorithm in object tracking.

1. Introduction

This paper addresses image segmentation with a reference distribution. Based on the optimization of a global measure of similarity between distributions, the problem consists of finding a region in an image, so that the distribution of image data within the region most closely matches a given model distribution. As several recent studies have shown [1, 2, 3, 4, 5, 6, 7, 8], the use of global measures outperforms standard techniques based on *pixelwise* information in the contexts of image segmentation and object tracking. Furthermore, segmentation with a reference distribution can yield robust region-based measures for image retrieval [1]. Possible measures include the Kullback–Leibler divergence [6] and the Bhattacharyya measure [2, 3, 4, 7]. However, the latter has shown superior performances over other criteria [3, 7]. Outstanding theoretical properties of the Bhattacharyya measure were also studied in information

theory [14], and demonstrate its wide potential for applications.

Unfortunately, optimization of a global similarity measure, for instance the Bhattacharyya measure, with respect to segmentation is *NP-hard*, and the problem has been commonly addressed with local, stepwise optimization procedures. In this connection, segmentation with a reference distribution has been generally stated as an active contour optimization via partial differential equations [2, 3, 4, 5, 6]. A gradient flow equation of contour evolution is derived in order to increase the similarity between the region within the contour and a given model, thereby leading to a local optimum at convergence. These methods lead to computationally intensive algorithms¹, which may limit significantly their application. Along with an incremental contour evolution, they 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 (cf. table 3). Moreover, 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.

Combinatorial graph cut algorithms [10, 11, 12], which view image segmentation as a label assignment following the discrete optimization of a cost function, have recently been of intense interest because they can guarantee global optima and numerical robustness, in nearly real-time. Several studies have shown that graph cut optimization can be quite effective in various computer vision problems, for instance, image segmentation [16, 17, 18], object tracking [19], motion estimation [20], visual correspondence [21], and restoration [11]. Most of existing graph cut segmentation methods optimize a sum over all pixels of pixel or pixel-neighborhood dependent data and variables. Variables which are global over the segmentation regions have been generally avoided because they cannot be written in a form amenable to graph cut optimization. A notable exception

¹Active contour segmentation with a reference distribution can be very slow in practice; it may require up to several minutes on typical CPUs.

is the *Trust Region Graph Cut (TRGC)* method of Rother *et al.* [1], [9], devoted to co-segmentation of regions of the same size in image pairs, where the problem of finding an image region consistent with a reference histogram arises. This work pioneered optimization of the L_1 norm of the difference between histograms with graph cuts. Inspired by trust region methods from continuous optimization [13], the method computes a sequence of parametric linear approximations of the energy and the corresponding optimal segmentations, so that the energy does not increase. Optimization of the energy is carried out over the approximation parameter rather than the segmentation. The authors used a submodular-supermodular procedure [15] for initialization of the sequence, which requires that the energy is supermodular, and demonstrated that this is the case for the L_1 norm. Moreover, 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, the *unnormalized* histogram depends on the size of the learning region and, as such, is not a complete representation of the class of target regions. For instance, TRGC cannot be readily applied to tracking an object whose size varies over an image sequence. In applications where the size of the target region is different from the size of the learning region, TRGC requires additional optimization/priors with respect to region size [1]. Furthermore, in information theory, it transpires that the L_1 norm is not the best measure of similarity between distributions [14].

Finally, it is worth mentioning the mean-shift tracking algorithm of Comaniciu *et al.* [8], where the target object is represented by an ellipsoidal shape. Given the position of the target ellipse in a previous frame, this algorithm seeks a new position in the neighborhood by optimizing a global similarity measure. The latter is defined over the target position, and smoothed with a spatial kernel, so that gradient-based optimization can be carried out in real-time. The current study investigates a related, but different variant of the problem. In our case, the problem consists of finding an accurate segmentation of regions with arbitrary shape and position (cf. the examples in Fig. 1). Apart from clear application in object tracking, this problem is also very useful in image segmentation, editing, and retrieval [1].

This study investigates an efficient algorithm for image segmentation with a reference distribution based on the Bhattacharyya measure. We derive an original upper bound of the Bhattacharyya measure by introducing an auxiliary labeling. From this upper bound, we reformulate image segmentation with a reference distribution as an optimization of an auxiliary function by graph cuts. Then, we formally demonstrate that the proposed procedure converges and give a statistical interpretation of the upper bound. The proposed

algorithm requires very few iterations to converge (less than 5 iterations), and finds nearly global optima. Quantitative evaluations and comparisons with TRGC and active contour optimization on the Microsoft GrabCut segmentation database (50 images with ground truth segmentations) demonstrated that the proposed algorithm brings improvements in regard to segmentation accuracy, computational efficiency, and optimality. We further demonstrate the flexibility of the algorithm in object tracking.

2. Efficient segmentation via an upper bound on the Bhattacharyya measure

Let $I_p = I(p) : \mathcal{P} \subset \mathbb{R}^2 \rightarrow \mathcal{Z} \subset \mathbb{R}^n, n \in \mathbb{N}$, be an image function from a positional array \mathcal{P} to the space \mathcal{Z} of a photometric variable such as intensity or color. The purpose of this study is to seek a region in \mathcal{P} , so that the kernel density estimate of image data within the region most closely matches a learned model distribution \mathcal{M} . We state the problem as the minimization of a discrete cost function with respect to a binary variable (labeling), $\mathcal{L}_p = \mathcal{L}(p) : \mathcal{P} \rightarrow \{0, 1\}$, which defines a variable partition of \mathcal{P} : $\mathbf{R}_1^{\mathcal{L}} = \{p \in \mathcal{P} / \mathcal{L}_p = 1\}$, corresponding to the target region (foreground), and $\mathbf{R}_0^{\mathcal{L}} = \{p \in \mathcal{P} / \mathcal{L}_p = 0\} = \mathcal{P} \setminus \mathbf{R}_1^{\mathcal{L}}$, corresponding to its complement in \mathcal{P} (background). The optimal labeling is sought following the minimization of a global cost function based on the Bhattacharyya measure. Before introducing the cost function, we first consider the following definitions for any binary labeling $\mathcal{L}_p = \mathcal{L}(p) : \mathcal{P} \rightarrow \{0, 1\}$:

- $\mathbf{P}_{\mathcal{L}}$ is the kernel density estimate (KDE) of the distribution of image data within region $\mathbf{R}_1^{\mathcal{L}} = \{p \in \mathcal{P} / \mathcal{L}_p = 1\}$:

$$\forall z \in \mathcal{Z} \quad \mathbf{P}_{\mathcal{L}}(z) = \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}}} K_z(I_p)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}})}, \quad (1)$$

where $\mathbf{A}(\mathbf{R})$ denotes the number of pixels within region \mathbf{R} : $\mathbf{A}(\mathbf{R}) = \sum_{\mathbf{R}} 1$. Possible choices of K_z are the Dirac function, which yields the histogram, or the Gaussian kernel $K_z(I_p) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp^{-\frac{\|z-I_p\|^2}{2\sigma^2}}$, with σ the width of the kernel.

- $\mathcal{B}(f, g)$ is the Bhattacharyya coefficient² measuring the amount of overlap between two distributions f and g :

$$\mathcal{B}(f, g) = \sum_{z \in \mathcal{Z}} \sqrt{f(z)g(z)} \quad (2)$$

The algorithm consists of finding an optimal labeling \mathcal{L}^{opt} that minimizes the following cost function over all possible binary labellings:

$$\mathcal{L}^{opt} = \arg \min_{\mathcal{L} : \mathcal{P} \rightarrow \{0,1\}} \mathcal{F}(\mathcal{L}), \quad (3)$$

²Note that the values of \mathcal{B} are always in $[0, 1]$, where 0 indicates that there is no overlap, and 1 indicates a perfect match.

with

$$\mathcal{F}(\mathcal{L}) = \underbrace{\mathbf{B}(\mathcal{L})}_{\text{Distribution matching}} + \underbrace{\lambda \mathbf{S}(\mathcal{L})}_{\text{Smoothness}}, \quad (4)$$

$$\mathbf{B}(\mathcal{L}) = -\mathcal{B}(\mathbf{P}_{\mathcal{L}}, \mathcal{M}) = -\sum_{z \in \mathcal{Z}} \sqrt{\mathbf{P}_{\mathcal{L}}(z) \mathcal{M}(z)}, \quad (5)$$

and $\mathbf{S}(\mathcal{L})$ is a smoothness prior to minimize the length of the partition boundary [22]:

$$\mathbf{S}(\mathcal{L}) = \sum_{\{p,q\} \in \mathcal{N}} s_{p,q} \delta_{\mathcal{L}_p \neq \mathcal{L}_q}, \quad (6)$$

with

$$\delta_{x \neq y} = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{if } x = y, \end{cases} \quad \text{and } s_{p,q} = \frac{1}{\|p - q\|} \quad (7)$$

\mathcal{N} is some neighborhood system containing all unordered pairs $\{p, q\}$ of neighboring elements of \mathcal{P} . λ is a positive constant that balances the relative contribution of region and boundary terms. \mathcal{L}^{opt} will yield an optimal region, $\mathbf{R}_1^{\mathcal{L}^{opt}} = \{p \in \mathcal{P} / \mathcal{L}_p^{opt} = 1\}$, so that the boundary of $\mathbf{R}_1^{\mathcal{L}^{opt}}$ is smooth and the kernel density estimate of image data within $\mathbf{R}_1^{\mathcal{L}^{opt}}$ most closely matches \mathcal{M} .

The *global* term $\mathbf{B}(\mathcal{L})$ in equation (5) is not directly amenable to graph cut optimization because it does not reference pixel or pixel-neighborhood penalties. It evaluates a global similarity measure between distributions and, therefore, its optimization is a challenging and *NP-hard* problem. To optimize efficiently \mathbf{B} , we first propose an original *upper bound* of \mathbf{B} by introducing an auxiliary labeling. From this upper bound, we reformulate the problem as an optimization of an auxiliary function by graph cut iterations. Then, we formally demonstrate that the proposed procedure converges and give a statistical interpretation of the upper bound. To introduce our formulation, let us first introduce the following proposition:

Proposition 1: *Given a fixed (auxiliary) labeling \mathcal{L}^a , for any labeling \mathcal{L} verifying $\mathbf{R}_1^{\mathcal{L}} \subset \mathbf{R}_1^{\mathcal{L}^a}$, i.e., the foreground region defined by \mathcal{L} is within the foreground region defined by \mathcal{L}^a , and $\forall \alpha \in [0, 1]$, we have the following upper bound of $\mathbf{B}(\mathcal{L})$*

$$\mathbf{B}(\mathcal{L}) \leq \mathcal{J}(\mathcal{L}, \mathcal{L}^a, \alpha) = \sum_{p \in \mathbf{R}_0^{\mathcal{L}^a}} \mathbf{m}_p(\mathbf{0}) + (1-\alpha) \sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} \mathbf{m}_p(\mathbf{1}), \quad (8)$$

with $\mathbf{m}_p(\mathbf{0})$ and $\mathbf{m}_p(\mathbf{1})$ given for each p in \mathcal{P} by

$$\begin{aligned} \mathbf{m}_p(\mathbf{0}) &= \frac{\delta_{\mathcal{L}_p^a \neq 0}}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \left(\mathbf{B}(\mathcal{L}^a) + \sum_{z \in \mathcal{Z}} K_z(I_p) \sqrt{\frac{\mathcal{M}(z)}{\mathbf{P}_{\mathcal{L}^a}(z)}} \right) \\ \mathbf{m}_p(\mathbf{1}) &= \frac{\mathbf{B}(\mathcal{L}^a)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \end{aligned} \quad (9)$$

Proof of proposition 1: Because $\mathbf{R}_1^{\mathcal{L}}$ and $\mathbf{R}_0^{\mathcal{L}}$ are complementary, we can rewrite $\mathbf{R}_1^{\mathcal{L}^a}$ as follows

$$\mathbf{R}_1^{\mathcal{L}^a} = (\mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_1^{\mathcal{L}}) \cup (\mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}) \quad (10)$$

For \mathcal{L} verifying $\mathbf{R}_1^{\mathcal{L}} \subset \mathbf{R}_1^{\mathcal{L}^a}$, we have $\mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_1^{\mathcal{L}} = \mathbf{R}_1^{\mathcal{L}}$. Therefore, from equation (10), we can rewrite $\mathbf{R}_1^{\mathcal{L}^a}$ as follows

$$\mathbf{R}_1^{\mathcal{L}^a} = \mathbf{R}_1^{\mathcal{L}} \setminus (\mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}) \quad (11)$$

Using this equation, we rewrite the kernel density estimate in (1) as follows

$$\mathbf{P}_{\mathcal{L}}(z) = \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} K_z(I_p) - \sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a}) - \mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}})} \quad (12)$$

Now because $\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}})$ is nonnegative, we have the following inequality

$$\begin{aligned} \mathbf{P}_{\mathcal{L}}(z) &\geq \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} K_z(I_p) - \sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \\ &= \mathbf{P}_{\mathcal{L}^a}(z) - \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \end{aligned} \quad (13)$$

Using this lower bound in the Bhattacharyya measure, we obtain the following upper bound of $\mathbf{B}(\mathcal{L})$ (z is omitted as argument of the distributions to simplify the equations)

$$\begin{aligned} \mathbf{B}(\mathcal{L}) &\leq -\sum_{z \in \mathcal{Z}} \sqrt{\left(\mathbf{P}_{\mathcal{L}^a} - \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \right) \mathcal{M}} \\ &= -\sum_{z \in \mathcal{Z}} \sqrt{\mathbf{P}_{\mathcal{L}^a} \mathcal{M}} \sqrt{1 - \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\mathbf{P}_{\mathcal{L}^a} \mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})}} \\ &= -\sum_{z \in \mathcal{Z}} \sqrt{\mathbf{P}_{\mathcal{L}^a} \mathcal{M}} \sqrt{1 - \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} K_z(I_p)}} \end{aligned} \quad (14)$$

Now notice the following inequality for any $0 \leq \mathbf{x} \leq 1$

$$\sqrt{1 - \mathbf{x}} \geq 1 - \mathbf{x}. \quad (15)$$

Because $\mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}} \subset \mathbf{R}_1^{\mathcal{L}^a}$, we have

$$0 \leq \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} K_z(I_p)} \leq 1 \quad (16)$$

Thus, applying inequality (15) gives

$$\sqrt{1 - \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} K_z(I_p)}} \geq 1 - \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}}} K_z(I_p)}{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} K_z(I_p)} \quad (17)$$

Finally, combining this inequality with (14) gives the following upper bound of $\mathbf{B}(\mathcal{L})$

$$\begin{aligned}
\mathbf{B}(\mathcal{L}) &\leq - \sum_{z \in \mathcal{Z}} \sqrt{\mathbf{P}_{\mathcal{L}^a} \mathcal{M}} \left(1 - \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}^a}} K_z(I_p)}{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} K_z(I_p)} \right) \\
&= \mathbf{B}(\mathcal{L}^a) + \sum_{z \in \mathcal{Z}} \sqrt{\mathbf{P}_{\mathcal{L}^a} \mathcal{M}} \frac{\sum_{p \in \mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}^a}} K_z(I_p)}{\mathbf{P}_{\mathcal{L}^a} \mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \\
&= \mathbf{B}(\mathcal{L}^a) + \sum_{p \in \mathbf{R}_0^{\mathcal{L}^a}} \frac{\delta_{\mathcal{L}_p^a \neq 0}}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \sum_{z \in \mathcal{Z}} K_z(I_p) \sqrt{\frac{\mathcal{M}(z)}{\mathbf{P}_{\mathcal{L}^a}(z)}}
\end{aligned} \tag{18}$$

Now notice the following inequality $\forall \alpha \in [0, 1]$

$$\begin{aligned}
\mathbf{B}(\mathcal{L}^a) &= \frac{\mathbf{B}(\mathcal{L}^a) \mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a} \cap \mathbf{R}_0^{\mathcal{L}^a})}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} + \frac{\mathbf{B}(\mathcal{L}^a) \mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \\
&\leq \sum_{p \in \mathbf{R}_0^{\mathcal{L}^a}} \frac{\delta_{\mathcal{L}_p^a \neq 0} \mathbf{B}(\mathcal{L}^a)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} + (1 - \alpha) \sum_{p \in \mathbf{R}_1^{\mathcal{L}^a}} \frac{\mathbf{B}(\mathcal{L}^a)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})}
\end{aligned} \tag{19}$$

Combining (18) and (19) proves *proposition 1*.

Definition 1: $\mathcal{A}(\mathcal{L}, \hat{\mathcal{L}})$ is called auxiliary function of cost function $\mathcal{F}(\mathcal{L})$ if it satisfies the following conditions:

$$\mathcal{F}(\mathcal{L}) \leq \mathcal{A}(\mathcal{L}, \hat{\mathcal{L}}) \tag{20}$$

$$\mathcal{A}(\mathcal{L}, \mathcal{L}) = \mathcal{F}(\mathcal{L}) \tag{21}$$

Auxiliary functions are commonly used in the Nonnegative Matrix Factorization (NMF) literature for optimization [23]. Rather than optimizing the cost function, one can optimize iteratively an auxiliary function of the cost function. At each iteration t , this amounts at optimizing over the first variable

$$\mathcal{L}^{(t+1)} = \arg \min_{\mathcal{L}} \mathcal{A}(\mathcal{L}, \mathcal{L}^{(t)}) \tag{22}$$

Thus, by definition of auxiliary function and minimum, we obtain the following monotonically decreasing sequence of the cost function

$$\mathcal{F}(\mathcal{L}^{(t)}) = \mathcal{A}(\mathcal{L}^{(t)}, \mathcal{L}^{(t)}) \geq \mathcal{A}(\mathcal{L}^{(t+1)}, \mathcal{L}^{(t)}) \geq \mathcal{F}(\mathcal{L}^{(t+1)}) \tag{23}$$

Proposition 2: For $\alpha = 0$, the following function is an auxiliary function of $\mathcal{F}(\mathcal{L})$

$$\mathcal{A}(\mathcal{L}, \mathcal{L}^a, \alpha) = \mathcal{J}(\mathcal{L}, \mathcal{L}^a, \alpha) + \mathbf{S}(\mathcal{L}) \tag{24}$$

Proof of proposition 2: To prove *proposition 2*, it suffices to verify conditions (20) and (21) for \mathcal{A} and \mathcal{F} . Condition (20) follows directly from *proposition 1*. For Condition (21), it suffices to see that when $\mathcal{L}^a = \mathcal{L}$, $\delta_{\mathcal{L}_p^a \neq 0} = 0 \forall p \in$

$\mathbf{R}_0^{\mathcal{L}}$, i.e., $\mathbf{m}_p(\mathbf{0}) = 0 \forall p \in \mathbf{R}_0^{\mathcal{L}}$. In this case, we have $\sum_{p \in \mathbf{R}_0^{\mathcal{L}}} \mathbf{m}_p(\mathbf{0}) = 0$. Therefore, we have for $\alpha = 0$

$$\begin{aligned}
\mathcal{A}(\mathcal{L}, \mathcal{L}, 0) &= \mathcal{J}(\mathcal{L}, \mathcal{L}, 0) + \mathbf{S}(\mathcal{L}) = \sum_{p \in \mathbf{R}_1^{\mathcal{L}}} \mathbf{m}_p(\mathbf{1}) + \mathbf{S}(\mathcal{L}) \\
&= \sum_{p \in \mathbf{R}_1^{\mathcal{L}}} \frac{\mathbf{B}(\mathcal{L})}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}})} + \mathbf{S}(\mathcal{L}) = \mathbf{B}(\mathcal{L}) + \mathbf{S}(\mathcal{L}) \\
&= \mathcal{F}(\mathcal{L})
\end{aligned} \tag{25}$$

which verifies condition (21).

Proposition 2 instructs us to consider the following procedure for minimizing functional \mathcal{F} .

Minimization procedure:

begin

- Initialize the auxiliary labeling so that the initial region corresponds to the hole image domain

$$\mathcal{L}^a(p) = \mathcal{L}^{(0)}(p) = 1 \quad \forall p \in \mathcal{P}$$

- Initialize α : $\alpha = \alpha_0$ with $0 < \alpha_0 < 1$

repeat

1. Optimize the auxiliary function over \mathcal{L}

$$\mathcal{L}^{(t)} = \arg \min_{\mathcal{L}: \mathbf{R}_1^{\mathcal{L}} \subset \mathbf{R}_1^{\mathcal{L}^a}} \mathcal{A}(\mathcal{L}, \mathcal{L}^a, \alpha)$$

2. Update \mathcal{L}^a by $\mathcal{L}^a = \mathcal{L}^{(t)}$
3. Decrease α : $\alpha = \alpha^\rho$ with $\rho > 1$

until Convergence ;

end

Convergence proof: When α approaches zero, $\mathcal{A}(\mathcal{L}, \mathcal{L}^a, \alpha)$ approaches an auxiliary function of cost function \mathcal{F} and, therefore, the above procedure leads to a monotonically decreasing sequence of \mathcal{F} . This comes directly from (23). Since the cost function is lower bounded (because the Bhattacharyya measure is upper bounded by one), the algorithm converges.

Optimization in step 1 with a graph cut: Now notice that the auxiliary function $\mathcal{A}(\mathcal{L}, \mathcal{L}^a, \alpha)$ in *step 1* of the optimization procedure is the sum of *unary* and *pairwise* (submodular) penalties. In combinatorial optimization, a global optimum of such sum can be computed efficiently in low-order polynomial time with a graph cut by solving an equivalent max-flow problem [10]. Furthermore, the condition that the solution should verify $\mathbf{R}_1^{\mathcal{L}} \subset \mathbf{R}_1^{\mathcal{L}^a}$ can be imposed easily by adding a hard constraint [16]. We used the well-known max-flow algorithm of Boykov and

Method	BMGC	TRGC [1]
Error	0.24%	2.33%

Table 1. Evaluation on the GrabCut database (50 images with ground truth segmentations): average error for the proposed method (BMGC) and the Target Region Graph Cut (TRGC) method in [1]. BMGC yielded a significant improvement in segmentation accuracy.

Kolmogorov [10] for the optimization in *step 1*. Here we omit the details of the max-flow algorithm and hard constraints. The details of these are well-known in the literature, and can be found in [10], [16].

Interpretation of the upper-bound optimization and link to statistical hypothesis testing: For a clear interpretation of the upper-bound optimization, let us assume that K_z is the Dirac function. In this case, for each pixel p within the current foreground region, i.e., $p \in \mathbf{R}_1^{\mathcal{L}^a}$ (at the first iteration, this corresponds to all the pixels in the image), we have

$$\begin{aligned} \mathbf{H}_p &= \mathbf{m}_p(\mathbf{0}) - (1 - \alpha)\mathbf{m}_p(\mathbf{1}) = \sqrt{\frac{\mathcal{M}(I_p)}{\mathbf{P}_{\mathcal{L}^a}(I_p)}} + \alpha \frac{\mathbf{B}(\mathcal{L}^a)}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \\ &= \sqrt{\frac{\mathcal{M}(I_p)}{\mathbf{P}_{\mathcal{L}^a}(I_p)}} - \alpha \frac{\mathcal{B}(\mathbf{P}_{\mathcal{L}^a}, \mathcal{M})}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \end{aligned} \quad (26)$$

When $\mathbf{H}_p < 0$, the graph cut excludes pixel p from the current foreground region so as to decrease the upper bound, $\mathcal{J}(\mathcal{L}, \mathcal{L}^a, \alpha)$. This has a clear meaning, and amounts to a statistical hypothesis testing by an image likelihood ratio test. It evaluates the hypotheses that the image at pixel p is drawn from model \mathcal{M} or from the image distribution within the current foreground region. If the likelihood ratio is lower than the following critical value

$$\frac{\mathcal{M}(I_p)}{\mathbf{P}_{\mathcal{L}^a}(I_p)} < \left(\alpha \frac{\mathcal{B}(\mathbf{P}_{\mathcal{L}^a}, \mathcal{M})}{\mathbf{A}(\mathbf{R}_1^{\mathcal{L}^a})} \right)^2, \quad (27)$$

$\mathbf{H}_p < 0$, which results in excluding pixel p from the foreground region. This makes sense because it results in decreasing the image distribution within the current foreground region at value I_p , which means a better match with the model at that value.

3. Experiments

Quantitative evaluation and comparison with TRGC:

To verify the optimality and accuracy of the proposed method, referred to as BMGC (Bhattacharyya Measure Graph Cut), we carried out a *quantitative* evaluation on the Microsoft GrabCut segmentation database [1] (50 images with ground truth segmentations into two regions: a

foreground and a background). Similar experiments on the same data were presented in [1] to evaluate the Trust Region Graph Cut (TRGC) method, which uses graph cut iterations to optimize the L_1 difference between histograms. Given the model distribution of the target region (foreground) learned from the ground truth, each image in the database is segmented and the average error, i.e., percentage of misclassified pixels in comparison to the ground truth, was evaluated to measure the accuracy of the proposed method.

Parameters for all the data (50 images): We fixed the parameters as follows. $\lambda = 10^{-5}$, $\alpha_0 = 0.85$, and $\rho = 1.1$. The total number of iterations is fixed equal to 4. We used a trivial initial auxiliary labeling: $\mathcal{L}^{(0)}(p) = 1 \ \forall p \in \mathcal{P}$, i.e., the initial foreground region correspond to the image domain. Therefore, the proposed method does not require an initialization. The photometric variable is color specified in RGB coordinates. A 3-dimensional histogram based on $192 \times 192 \times 192$ bins was used as a density estimate.

Table 1 reports the results: BMGC yielded an average error equal to 0.24%, whereas the authors in [1] report an average error equal to 2.33% on the same data set. In comparison to TRGC, the proposed method leads to a significant improvement in segmentation accuracy. Furthermore, different from TRGC, the proposed method does not use a submodular-supermodular procedure [15] for initialization.

Segmentation examples: A representative sample of the tests we run on the GrabCut segmentation database is depicted in Figure 1. The image, its segmentation with BMGC, and the corresponding ground truth are shown, respectively, in the first, second, and third columns. The error, obtained optimal Bhattacharyya measure corresponding to the optimal labeling $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M})$, initial Bhattacharyya measure $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M})$, run time, and image size are given for each example. The proposed method yielded very accurate segmentations, although in some examples a significant overlap (similarity) exists between the foreground and background distributions. Furthermore, it requires very few iterations to converge (4 iterations). The example in Figure 2 illustrates the fast convergence of the proposed algorithm.

Optimality: To measure the optimality of the algorithm, we evaluated over the GrabCut database the statistics (mean and std) of the optimal Bhattacharyya measures obtained with BMGC (refer to table 2 for details). We obtained $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M}) = 0.9984 \pm 0.0016$ (\mathcal{B} expressed as *mean* \pm *std*). The obtained optimal energies are very close to 1, which is the maximum possible value. This demonstrates that BMGC yields nearly global optima, although it requires only few iterations.

Comparisons with active contour methods: Using the Grabcut database, we performed comparisons with gradient flow active contour optimization, commonly used to tackle global distribution measures [2, 3, 4, 5, 6, 7]. The results and computation time/load, reported in table 3, demonstrate

Obtained measure $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M})$ (mean \pm std)	Initial measure $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M})$ (mean \pm std)	Maximum measure
0.9984 ± 0.0016	0.4820 ± 0.1186	1

Table 2. Evaluation of the optimality of the algorithm on the GrabCut database (50 images): statistics (mean and std) of the optimal Bhattacharyya measures obtained with BMGC ($\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M})$) and the initial measure ($\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M})$). *The optimal measures are very close to 1, which is the maximum possible value. This demonstrates that BMGC yields nearly global optima with few iterations.*

Method	BMGC	Active contour optimization [2, 3, 4]
Number of KDEs per image (mean)	4	1737
Run time per image (mean)	3.51 secs	631 secs
Error	0.24%	2.49%
Obtained Bhattacharyya measure (mean \pm std)	0.9984 ± 0.0016	0.9791 ± 0.01

Table 3. Comparisons with active contour optimization [2, 3, 4] on the GrabCut database: number of Kernel Density Estimations per image (mean), run time per image (mean), error (percentage of misclassified pixels), and obtained Bhattacharyya measure (the higher the measure, the more optimal the solution). BMGC leads to significant improvements in regard to segmentation accuracy, optimality, and computational efficiency. *It relaxes the need of a large number of updates of the kernel densities and the corresponding measures.*

that BMGC leads to significant improvements in regard to computational efficiency, segmentation accuracy, and optimality. BMGC has an important computational advantage over active contour methods. It leads to a significant decrease in computational load because it does not require a large number of updates of computationally onerous kernel densities. With BMGC, the solution is reached after 4 Kernel Density Estimations (KDEs), whereas the average number of KDEs for active contour optimization is 1737 (mean). BMGC took an average run time equal to 3.51 secs to compute nearly global optima corresponding approximately to the maximum possible value of the Bhattacharyya measure, whereas active contour optimization took 631 seconds, converged to a less optimal Bhattacharyya measure, and yielded a higher error (refer to table 3 for details).

Object tracking examples: The example in Figure 3 depicts the tracking of an object with an arbitrary shape in the tennis table sequence. Given the model distribution learned from a manual segmentation of the first frame, the target object is recovered with BMGC in subsequent frames. In frame 30, the target object undergoes significant variations in shape/size in comparison to the first frame. The proposed method handles implicitly these variations because no assumptions were made as to the size, shape, or position of the target region.

References

- [1] C. Rother, V. Kolmogorov, T. Minka, and A. Blake, Cosegmentation of Image Pairs by Histogram Matching—Incorporating a Global Constraint into MRFs, *CVPR*, 2006.
- [2] I. Ben Ayed, S. Li, and I. Ross, A Statistical Overlap Prior for Variational Image Segmentation, *Int. J. of Computer Vision*, 85(1): 115-132, 2009.
- [3] T. Zhang and D. Freedman, Improving performance of distribution tracking through background mismatch, *IEEE Trans. on Pattern Anal. and Machine Intell.*, 27(2):282-287, 2005.
- [4] D. Freedman and T. Zhang, Active contours for tracking distributions, *IEEE Trans. on Image Processing*, 13(4):518-526, 2004.
- [5] T. F. Chan, S. Esedoglu, and K. Ni, Histogram Based Segmentation Using Wasserstein Distances, *SSVM 2007*, pp. 697-708.
- [6] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson, Image Segmentation Using Active Contours: Calculus of Variations or Shape Gradients?, *SIAM Applied Mathematics*, 63(6): 2128-2154, 2003.
- [7] O. V. Michailovich, Y. Rathi, and A. Tannenbaum, Image Segmentation Using Active Contours Driven by the Bhattacharyya Gradient Flow, *IEEE Trans. on Image Processing*, 16(11):2787-2801, 2007.
- [8] D. Comaniciu, V. Ramesh, and P. Meer, Kernel-based object tracking, *IEEE Trans. on Pattern Anal. and Machine Intell.*, 25(5):564-577, 2003.
- [9] V. Kolmogorov, Y. Boykov, and C. Rother, Applications of parametric maxflow in computer vision, *ICCV 2007*.
- [10] Y. Boykov and V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, *IEEE Trans. on Pattern Anal. and Machine Intell.*, 26(9):1124-1137, 2004.
- [11] Y. Boykov, O. Veksler, and R. Zabih, Efficient Approximate Energy Minimization via Graph Cuts, *IEEE Trans. on Pattern Anal. and Machine Intell.*, 20(12):1222-1239, 2001.

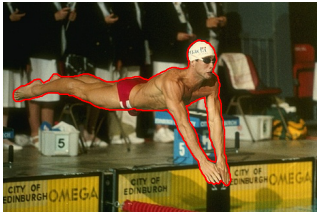





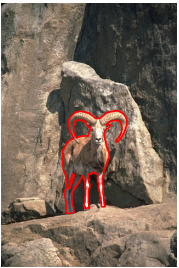


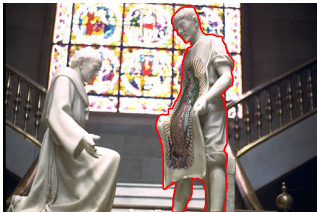





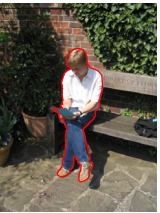


Image	Segmentation with BMGC	Ground truth	Measures
			$error = 0.04\%$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M}) = 0.9989$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M}) = 0.3779$ run time: 2.59 secs size: 321×481
			$error = 0.13\%$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M}) = 0.9973$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M}) = 0.4741$ run time: 3.84 secs size: 600×450
			$error = 0.05\%$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M}) = 0.9991$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M}) = 0.4287$ run time: 2.91 secs size: 481×321
			$error = 0.19\%$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M}) = 0.9991$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M}) = 0.5885$ run time: 2.80 secs size: 321×481
			$error = 0.16\%$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M}) = 0.9968$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M}) = 0.5543$ run time: 2.81 secs size: 321×481
			$error = 0.09\%$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{opt}}, \mathcal{M}) = 0.9988$ $\mathcal{B}(\mathbf{P}_{\mathcal{L}^{(0)}}, \mathcal{M}) = 0.3704$ run time: 3.72 secs size: 600×450

Figure 1. A sample of the segmentations obtained with the proposed method. First column: the image and the segmentation boundary obtained with BMGC, depicted with the red curve. Second column: foreground region obtained with BMGC. Third column: ground truth. The models are learned from the ground truth as in [1]. For all the images, we used a trivial initial labeling: $\mathcal{L}^{(0)}(p) = 1 \forall p \in \mathcal{P}$. $\lambda = 10^{-5}$, $\alpha_0 = 0.85$, and $\rho = 1.1$. The total number of iterations is fixed equal to 4. The photometric variable is color specified in RGB coordinates. A 3-dimensional histogram based on $192 \times 192 \times 192$ bins was used as a density estimate. BMGC yielded accurate results, although in some examples a significant overlap (similarity) exists between the foreground and background distributions.

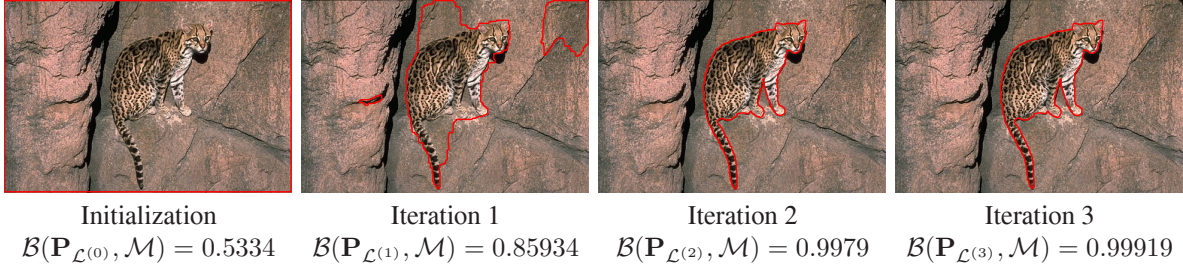


Figure 2. Illustration of the fast convergence of the algorithm: evolution of the segmentation and optimized Bhattacharyya measure within the first three iterations. In the third iteration, the algorithm yielded a Bhattacharyya measure very close to 1, which is the maximum possible value. $\alpha_0 = 0.85$ and $\rho = 1.1$.

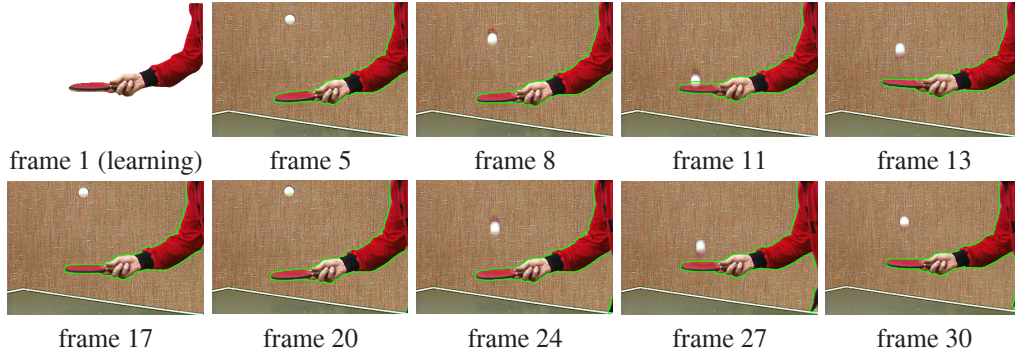


Figure 3. Tracking of an object with an arbitrary shape in the table tennis sequence. Given the model learned from a manual segmentation of frame 1, the object is recovered with BMGC in subsequent frames (green curves). We used a trivial initial labeling for each test frame: $\mathcal{L}^{(0)}(p) = 1 \forall p \in \mathcal{P}$. The photometric variable is color specified in RGB coordinates. A 3-dimensional kernel density estimate of the distribution was computed using $32 \times 32 \times 32$ bins and a kernel width $\sigma = 2/32$. $\rho = 3$, $\alpha = 0.85$, and $\lambda = 2.5 \times 10^{-5}$. Number of iterations per frame: 4. Run time/frame=1.51 secs. Size: 240x352x30 frames. In frame 30, the target object undergoes significant variations in shape/size in comparison to the first frame. The proposed method handles these variations because no assumptions were made as to the size, shape, or position of the object.

[12] V. Kolmogorov and R. Zabih, What Energy Functions can be Minimized via Graph Cuts?, *IEEE Trans. on Pattern Anal. and Machine Intell.*, 26(2):147-159, 2004.

[13] D. P. Bertsekas, Nonlinear Programming, *Athena Scientific, 2nd edition*, 1999.

[14] F. Aherne, N. Thacker, and P. Rockett, The Bhattacharyya metric as an absolute similarity measure for frequency coded data, *Kybernetika*, 32(4):1-7, 1997.

[15] M. Narasimhan and J. Bilmes, A supermodular-submodular procedure with applications to discriminative structure learning, *In UAI*, July 2005.

[16] Y. Boykov, and G. Funka-Lea, Graph Cuts and Efficient N-D Image Segmentation, *Int. J. of Computer Vision*, 70(2):109-131, 2006.

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

[18] X. Liu, O. Veksler, and J. Samarabandu, Graph Cut with Ordering Constraints on Labels and its Applications, *CVPR*, 2008.

[19] J. Malcolm, Y. Rathi, and A. Tannenbaum, Multi-Object Tracking Through Clutter Using Graph Cuts, *ICCV*, 2007.

[20] V. Lempitsky, S. Roth, and C. Rother, FusionFlow: Discrete-Continuous Optimization for Optical Flow Estimation, *CVPR*, 2008.

[21] J. Kim, V. Kolmogorov, and R. Zabih, Visual Correspondence Using Energy Minimization and Mutual Information, *ICCV*, 2003.

[22] Y. Boykov and V. Kolmogorov, Computing geodesics and minimal surfaces via graph cuts, *ICCV*, 2003.

[23] D. D. Lee and H.S. Seung, Algorithms for nonnegative matrix factorization, *Advances in Neural Information Processing Systems (NIPS) 2002*, 13: 556-562.