Superdifferential Cuts for Binary Energies

Tatsunori Taniai
University of Tokyo, Japan
taniai@iis.u-tokyo.ac.jp

Yasuyuki Matsushita
Osaka University, Japan
yasumat@ist.osaka-u.ac.jp

Takeshi Naemura
University of Tokyo, Japan
naemura@nae-lab.org

Abstract

We propose an efficient and general purpose energy optimization method for binary variable energies used in various low-level vision tasks. Our method can be used for broad classes of higher-order and pairwise non-submodular functions. We first revisit a submodular-supermodular procedure (SSP) [19], which is previously studied for higher-order energy optimization. We then present our method as generalization of SSP, which is further shown to generalize several state-of-the-art techniques for higher-order and pairwise non-submodular functions [2, 9, 25]. In the experiments, we apply our method to image segmentation, deconvolution, and binarization, and show improvements over state-of-the-art methods.

1. Introduction

Many low-level vision problems such as image segmentation, binarization, denoising, and tracking are often formulated as binary energy minimization [4, 22, 1, 25, 6]. For example, in image segmentation, the use of Markov random field formulations [8] and graph cuts [17, 4] has been becoming one of primary approaches [3, 23, 21, 25, 9, 2, 10, 11, 26, 20, 1, 22]. In this approach, the energy function is typically formulated as

$$E(S) = R(S) + Q(S), \qquad (1)$$

where $R(S)$ describes appearance consistencies between resulting segments S and given information about target regions, and $Q(S)$ enforces smoothness on segment-boundaries. The form of $R(S)$ is often restricted to simple linear (*i.e.*, pixelwise unary) forms [3, 23, 21] because graph cuts allow globally optimal inference only for unary and submodular pairwise forms of energies [17]. However, recent studies [25, 2, 10, 11, 26, 20, 1, 22] have shown that the use of higher-order information (or non-linear terms) can yield outstanding improvements over conventional pixelwise consistency measures.

In general, higher-order terms involve difficult optimization problems. Recent promising approaches try reduc-
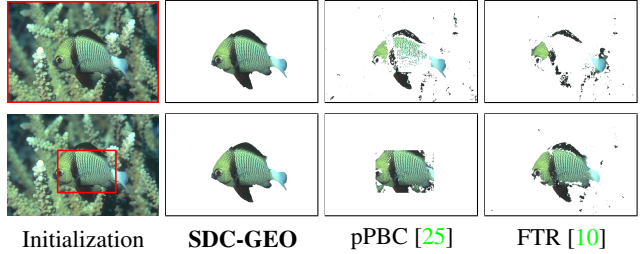

Figure 1. Matching foreground color distribution using the proposed **SDC-GEO**, parametric pseudo bound cuts (pPBC) [25], and fast trust region (FTR) [10] with two types of initialization. pPBC can only successively reduce the initial segment, while our method allows arbitrary directions of optimization and is thus robust to initialization. (L_2 distance for 64^3 bins of RGB histograms are used)

ing energies by iteratively minimizing either first-order approximations (gradient descent approach) [10, 11] or upper-bounds (bound optimization approach) [25, 2, 26, 20, 1] of non-linear functions using graph cuts. The bound optimization approach has some advantages over the gradient descent approach [2]: It requires no parameters (*e.g.*, step-size) and never worsens the solutions during iterations. But we must in turn derive appropriate bounds for individual functions. A notable work is *auxiliary cuts* (AC) [2] by Ayed *et al*., where they derive general bounds for broad classes of non-linear functionals for segmentation. However, the bounds derived in [2] are formulated to successively reduce target regions; thus the resulting segments are restricted within initial segments. Such a property actually limits the applications and accuracy of the method.

In order to derive more accurate and useful bounds, we revisit a submodular-supermodular procedure (SSP) [19], a general bound optimization scheme for supermodular functions. We then propose a bound optimization method as generalization of SSP. Unlike SSP, our method can be used even for non-supermodular functions; and unlike AC, it allows bi-directional optimization (see Fig. 1 for an illustration in segmentation) and can produce more accurate approximation bounds. We further show that our method can be seen as generalization of AC and some state-of-the-art method [9] for pairwise non-submodular functions.

This paper makes the following contributions:

- we propose an optimization method for broad classes of higher-order and pairwise non-submodular functions that allows arbitrary directions of convergence and outperforms the state-of-the-art [9, 25, 10, 2].
- our method generalizes previous optimization methods including from early [19] to state-of-the-art [9, 25, 2, 1] methods.

1.1. Scope of the Problems

Our objective is to seek the binary variable S such that it minimizes $E(S)$ of Eq. (1). In this paper, we focus on three types of energy functions. Before defining those functions, we define the following function as the basis of all types.

Pairwise Submodular Functions.

Let $s_i \in \{0, 1\}$ be a binary variable defined for pixels $i \in \Omega$ in the image domain Ω, and we define $S = \{i | s_i = 1\}$ as a *segment* in the domain Ω. If $R(S)$ in Eq. (1) is a linear product of a function $h_i = h(i) : \Omega \to \mathbb{R}$

$$R(S) = \langle h, S \rangle = \sum_{i \in \Omega} h_i s_i \qquad (2)$$

and $Q(S)$ is the sum of pairwise functions

$$Q(S) = \sum_{(i,j) \in \Omega} m_{ij} s_i s_j, \qquad (3)$$

and if all the quadratic terms are non-positive ($m_{ij} \leq 0$), then $E(S)$ is submodular and can be globally minimized via graph cuts [4, 17] in polynomial time.

Type-1: Higer-Order Supermodular Energies.

We consider the energies $E(S)$ with pairwise submodular $Q(S)$ and the following form of $R(S)$:

$$R_{type1}(S) = \sum_z R_z(S) = \sum_z f_z(\langle g_z, S \rangle) \qquad (4)$$

where $f_z(x)$ is convex and $g_z(i) : \Omega \to \mathbb{R}^+$ is a positive function. Hence, $R_z(S)$ is supermodular, *i.e.*, if it satisfies the following inequality for any $X, Y \subseteq \Omega$:

$$R_z(X) + R_z(Y) \leq R_z(X \cap Y) + R_z(X \cup Y). \qquad (5)$$

Note that $R'_z(S) = -R_z(S)$ is submodular, if $R_z(S)$ is supermodular. While any submodular functions can be minimized in polynomial time [24], the minimization of supermodular functions is NP-hard.

Higher-order supermodular functions have been used, for example, as a L_p-distance histogram constraint $R(S) = \sum_{z \in Z} \left| h_z - n_z^S \right|^p$ for co-segmentation [22, 28,

18] and tracking [13], where $z \in Z$ is a bin of color or feature histograms, h_z is the given target histogram, and n_z^S is the number of pixels in S that fall into the bin z. A volumetric constraint $R(S) = |V_0 - |S||^p$ has been also used for medical image analysis [11, 10].

Type-2: Fractional Higer-Order Energies.

We further deal with non-supermodular functions:

$$R_{type2}(S) = \sum_z f_z(\langle g_z, S \rangle / \langle w_z, S \rangle) \qquad (6)$$

such that $R_z(S) = f_z(\langle g_z, S \rangle / \langle w_z, S_0 \rangle)$ becomes a Type-1 term for fixed S_0. Examples of such functions include the KL-divergence $R(S) = -\sum_z p_z \log(n_z^S / |S| + \epsilon) + const.$ and the Bhattacharyya coefficient $R(S) = -\sum_z \sqrt{p_z n_z^S / |S|}$, both are used for image segmentation [1, 2, 25]. Here, $\sum_z p_z = 1$ is the target distribution.

Type-3: Pairwise Non-submodular Energies.

We also consider pairwise non-submodular energies, *i.e.*, $Q(S)$ of Eq. (3) containing non-submodular or supermodular terms (*i.e.* $m_{ij} > 0$). QPBO [16] is often used for such functions, but it leaves many variables *unlabeled* when the amount of non-submodular terms is significant [9].

2. Submodular-Supermodular Procedure

Before presenting our method, we review SSP [19], an optimization method for general supermodular functions, and later propose our method as its generalization.

SSP is classified as a bound optimization approach, where a tight upper bound function $\hat{E}(S|S^t)$ given an auxiliary variable S^t is derived for $E(S)$, *i.e.*,

$$E(S) \leq \hat{E}(S|S^t) \text{ and } E(S^t) = \hat{E}(S^t|S^t). \qquad (7)$$

Then, the bound is iteratively minimized as

$$S^{t+1} = \arg\min \hat{E}(S|S^t), \ (t = 0, 1, 2, ...). \qquad (8)$$

Here, it is guaranteed that the energy does not go up, *i.e.* $E(S^t) \geq E(S^{t+1})$ holds for any t, because $E(S^t) = \hat{E}(S^t|S^t) \geq \hat{E}(S^{t+1}|S^t) \geq E(S^{t+1})$.

2.1. Permutation Bounds

SSP derives tight bounds for supermodular functions based on the *superdifferential* [7, 12], which is a similar concept to the subderivative of continuous functions. Given a supermodular functions $R(S)$ and $S^t \subseteq \Omega$, a modular (or linear) function $H(S) := \langle h, S \rangle + R(\emptyset)$ that satisfies

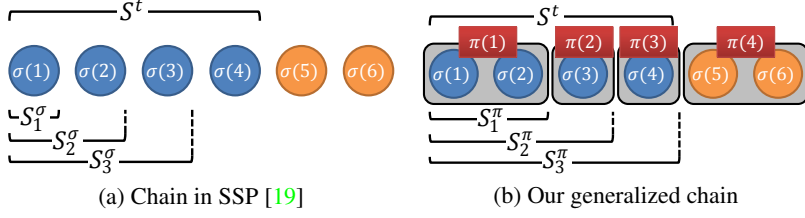$$H(S) - H(S^t) \geq R(S) - R(S^t) \qquad (9)$$

(a) Chain in SSP [19]   (b) Our generalized chain

Figure 2. Illustration of the chain and permutation.

Segments   Euclidean   Geodesic [5]

Figure 3. Examples of Euclidean and geodesic distance [5] for segments (pink).

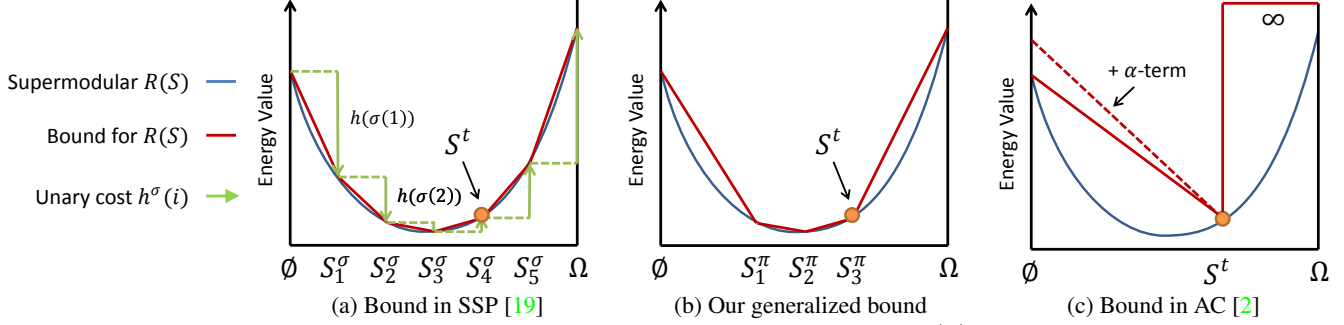(a) Bound in SSP [19]   (b) Our generalized bound   (c) Bound in AC [2]

Figure 4. Illustration of upper bounds for supermodular functions. The supermodular function $R(S)$ and its bounds are visualized by blue and red lines, respectively. The green arrows in (a) show the unary costs $h(\sigma(j))$ of a supergradient $H^\sigma(S|S^t)$.

for any $S \subseteq \Omega$, is called a *supergradient* of R at S^t. We denote $\partial R(S^t)$ the set of all the supergradients of R at S^t, which is called the superdifferential. Notice that if $H(S^t) = R(S^t)$ holds, then $H(S)$ gives a tight upper bound to $R(S)$. Such extreme points of $\partial R(S^t)$ may be obtained using the following theorem.

Theorem 1 (Theorem 6.11 in [7]): *For any $S^t \subseteq \Omega$, a modular function $H(S) = \langle h, S \rangle + R(\emptyset)$ is an extreme point of $\partial R(S^t)$, if and only if there exists a maximal chain*

$$C : \emptyset = S_0 \subset S_1 \subset \cdots \subset S_n = \Omega, \qquad (10)$$

with $S_j = S^t$ for some j, such that

$$H(S_j) - H(S_{j-1}) = R(S_j) - R(S_{j-1}), \; (j = 1, ..., |\Omega|). \qquad (11)$$

Based on this theorem, SSP [19] derives a supergradient $H^\sigma(S|S^t)$ of $R(S)$ at S^t by the following greedy algorithm. Let $\sigma(j) : \{1, 2, ..., |\Omega|\} \to \Omega$ be a permutation of Ω that assigns the elements in S^t to the first $|S^t|$ positions, *i.e.*, $\sigma(j) \in S^t$ if and only if $j \leq |S^t|$. A maximal chain C^σ is then defined as $S_0^\sigma = \emptyset$ and $S_j^\sigma = \{\sigma(1), \sigma(2), ..., \sigma(j)\}$, so $S_{|S^t|}^\sigma = S^t$. See Fig. 2 (a) for an illustration. Using this chain C^σ, a supergradient $H^\sigma(S|S^t)$ is obtained as

$$H^\sigma(S|S^t) = \langle h^\sigma, S \rangle + R(\emptyset), \qquad (12)$$

where each unary cost $h^\sigma(i)$ is given by

$$h^\sigma(\sigma(j)) = R(S_j^\sigma) - R(S_{j-1}^\sigma), \; (j = 1, 2, .., |\Omega|). \qquad (13)$$

Figure 4 (a) illustrates how $h^\sigma(\sigma(j))$ are computed. Since $R(S)$ is supermodular, variables in earlier positions of σ are assigned lower unary costs, *i.e.*, more prone to be labeled as 1 via cost-minimization. Therefore, if we knew a priori how likely each variable is 1, the ideal permutation would arrange variables in order of decreasing likelihood so as to maximize the likelihood via cost minimization. Here, the bounds $H^\sigma(S)$ approximate $R(S)$ tightly at along the chain of solutions $\{S_j^\sigma\}$. However, because $H^\sigma(S)$ can largely deviate from $R(S)$ at other than $\{S_j^\sigma\}$, SSP is problematic when likelihood or permutation is given inaccurate.

3. Proposed Method

In the following sections, we first present our key idea by extending SSP [19]. We then show how to apply it to and optimize the three types of functions in Sec. 3.2, and describe implementation details in Sec. 3.3.

3.1. Grouped Permutation Bounds

In this section, we derive general bounds for Type-1 terms $R(S) = f(\langle g, S \rangle)$ by extending the SSP's permutation scheme. First, we introduce a *grouped permutation* π, which is made by grouping SSP's ordered-elements $\sigma = \{\sigma(1), \sigma(2), \cdots, \sigma(|\Omega|)\}$ into M ($M \leq |\Omega|$) groups: $\pi(1), \pi(2), \cdots, \pi(M) \subseteq \Omega$. Each group $\pi(j)$ contains some consecutive elements of σ: $\pi(j) = \{\sigma(j'), \sigma(j' + 1), \cdots, \sigma(j' + m)\}$, and groups are mutually disjoint: $\pi(j) \cap \pi(j') = \emptyset$ if $j \neq j'$. Using this grouped permutation π we define a chain C^π: $S_0^\pi = \emptyset$ and $S_j^\pi = \pi(1) \cup \pi(2) \cup ... \cup \pi(j)$ as illustrated in Fig. 2 (b). Here,

we make sure that any group does not across $\sigma(|S^t|)$ and $\sigma(|S^t|+1)$, so there exists $S_j^\pi = S^t$ for some j. Then, our bound for $R(S)$ is defined similarly to that of SSP in Eq. (12) as

$$H^\pi(S|S^t) = \langle h^\pi, S \rangle + R(\emptyset), \qquad (14)$$

where unary costs $h^\pi(i) : \Omega \to \mathbb{R}$ are defined for $i \in \pi(j)$ and $j = 1, 2, ..., M$ as

$$h^\pi(i) = g(i) \left[R(S_j^\pi) - R(S_{j-1}^\pi) \right] / \langle g, \pi(j) \rangle. \qquad (15)$$

The essence of this formulation becomes clearer if we assume $g = 1$ so that $H^\pi(S|S^t)$ becomes piecewise-mean-approximations of SSP's bound $H^\sigma(S|S^t)$, which is visualized in Fig. 4 (b) using the example permutation π shown in Fig. 2 (b). Note that our bound $H^\pi(S|S^t)$ becomes equivalent to the supergradients $H^\sigma(S|S^t)$ of SSP, if $\pi(j) = \{\sigma(j)\}$.

Proposition 1: *The function $H^\pi(S|S^t)$ satisfies the conditions of Eq. (7) and is thus a tight upper bound for $R(S)$.*

Proof. See our supplementary. □

The spirit behind this grouping or piecewise-mean-approximation scheme is to make fine/coarse approximation bounds when permutations σ are accurate/inaccurate. Specifically, when the permutation of $\sigma(j)$ and $\sigma(j+1)$ is unreliable, we put them into the same group in order to treat them equally and leave a decision (*i.e.*, which is more likely to be labeled as 1) to other interactions, *e.g.*, pairwise smoothness terms. As we will show in Sec. 3.3, our method makes coarse-to-fine approximation bounds by iterations.

3.2. Optimization Procedure

We optimize $E(S)$ by iteratively minimizing its approximation function $\hat{E}(S|S^{t-1})$ derived by our grouped permutation bounds. Here, the minimization of $\hat{E}(S)$ is achieved using a max-flow/min-cut algorithm [4].

Type-1: We derive a bound $H_z^\pi(S|S^t)$ for each $R_z(S)$ of $R_{type1}(S)$, and set $\hat{E}(S|S^t) = \sum_z H_z^\pi(S|S^t) + Q(S)$. Here, it is guaranteed that minimization of $\hat{E}(S)$ does not increase the energy $E(S)$. Therefore, its optimization procedure is a simple iteration algorithm shown in Algorithm 1 (without lines 5 and 6).

Type-2: Similarly to [2, 25], we approximate $R_{type2}(S)$ by partially fixing S at S^t as

$$\tilde{R}_{type2}(S|S^t) = \sum_z f_z \left(\langle g_z, S \rangle / \langle w_z, S^t \rangle \right). \qquad (16)$$

Here, $\tilde{R}_{type2}(S|S^t)$ is a Type-1 function. Therefore, we can approximate $\tilde{R}_{type2}(S|S^t)$ using our bounds

---

Algorithm 1 OPTIMIZATION FOR TYPE-1,3 [TYPE-2]

1: Initialize S^0
2: **for** $t = 0, 1, 2, \cdots$ **do**
3:    Create a permutation π // for all Types
4:    $S^{t+1} \leftarrow \arg\min \hat{E}(S|S^t)$ // for Type-1,3
5:    $[\{S^\lambda\} \leftarrow \arg\min \hat{E}(S|S^t) + \lambda(|S^t - |S|)$ for $^\forall \lambda]$
6:    $[S^{t+1} \leftarrow \arg\min E(S)$ for $S \in \{S^\lambda, S^t\}]$ // for Type-2
7: **end for**

$H^\pi(S|S^t)$. As long as $S \subseteq S^t$ is forced, this linear function H^π makes a tight bound for R_{type2} [2]. However, we rather do not restrict S for allowing bi-directional optimization. In this case, the minimization of $\hat{E}(S|S^t) = H^\pi(S|S^t) + Q(S)$ may increase the actual energy $E(S)$. For this we use the pseudo bound optimization scheme of [25], where we make a family of relaxed bounds $\hat{E}_\lambda(S) = \hat{E}(S|S^t) + \lambda(|S^t| - |S|)$, and we exhaustively search $S^\lambda = \arg\min \hat{E}_\lambda(S)$ for all $\lambda \in (-\infty, \infty)$ by using parametric maxflow [15]. Then we choose the best S^λ that minimizes $E(S)$. Therefore, the optimization procedure in Algorithm 1 takes lines 5 and 6 instead of 4 for Type-2.

Type-3: We make a bound $\hat{E}(S)$ by approximating each of non-submodular terms $R(S) = m_{ij} s_i s_j$ $(m_{ij} > 0)$ with a linear function $h_i s_i + h_j s_j + const$. Its form depends on both current values (s_i^t, s_j^t) and the permutation π of $\{i, j\}$, and given as $h(i') = \left[R(S_{j'}^\pi) - R(S_{j'-1}^\pi) \right] / |\pi(j')|$ similarly to Eq. (15). We summarize the conversion in Table 1.

3.3. Implementation Details

We make permutations σ and π according to the signed distance from the boundary of S^t. In [22], the Euclidean distance was used, but we use geodesic distance [5] for more effectively creating the permutations. Figure 3 shows examples of both distances. We denote the geodesic distance for segments S by $D(i|S, I) : \Omega \to \mathbb{R}$, and $D(i|S) \leq 0$ for $i \in S$. See Eq. (7) of [5] and also our supplementary for its definition. As discussed in [5], $D(i|S, I)$ is efficiently computed in $O(|\Omega|)$ using an approximate algorithm [27].

Using the geodesic distance, we construct the bound $H^\pi(S|S^t)$ in each iteration as follows. Firstly, we compute $D(i|S^t)$ for the current segments S^t. Secondly, we make a permutation σ such that $D(\sigma(j)) \leq D(\sigma(j+1))$. Finally, we make a grouped permutation π from σ. We process $\sigma(j)$ from $\sigma(2)$ to $\sigma(|\Omega|)$, and put $\sigma(j)$ into the same group with $\sigma(j-1)$ if $D(\sigma(j)) - D(\sigma(j-1)) \leq \tau$, while making sure the group does not across $\sigma(|S^t|)$ and $\sigma(|S^t|+1)$. Basically, the size of the threshold τ reflects how much the permutation σ by $D(i)$ is reliable. We empirically use a grouping threshold given by $\tau = (\mu + as)/(t+1)^\kappa$ $(t = 0, 1, 2, ...)$, where μ and s are the mean and standard deviation of distance differences $|D(\sigma(j)) - D(\sigma(j-1))|$. We use this monotonically decreasing thresholds, because as iterations

Table 1. The proposed linear conversions of non-submodular terms $m_{ij}s_is_j$ with comparisons to the conversions of SSP [19] and LSA [9]. Below, the subscript of m_{ij} is omitted. Notice that our linear conversion can bee seen as generalization of both SSP and LSA-AUX.

(s_i^t, s_j^t)	Ours	SSP [19]	LSA-AUX [9]	LSA-TR [9]
$(0,0)$	$\frac{m}{2}s_i + \frac{m}{2}s_j$ if $\pi(1)=\{i,j\}$ ms_j if $\pi(1)=\{i\}$ ms_i if $\pi(1)=\{j\}$	ms_j if $\sigma(1)=i$ ms_i if $\sigma(1)=j$	$\frac{m}{2}s_i + \frac{m}{2}s_j$	0
$(0,1)$	ms_i	ms_i	ms_i	ms_i
$(1,0)$	ms_j	ms_j	ms_j	ms_j
$(1,1)$	$\frac{m}{2}s_i + \frac{m}{2}s_j$ if $\pi(1)=\{i,j\}$ ms_j if $\pi(1)=\{i\}$ ms_i if $\pi(1)=\{j\}$	ms_j if $\sigma(1)=i$ ms_i if $\sigma(1)=j$	$\frac{m}{2}s_i + \frac{m}{2}s_j$	$ms_i + ms_j - m$

proceed the segments S^t are expected to be more accurate and so permutations σ by $D(i|S^t)$ becomes accordingly more reasonable. Also, when $S^0 = \Omega$ or $S^0 = \emptyset$, so $D(i|S^t)$ cannot be defined, we set $\pi(1) = \Omega$. This makes the full linear approximations of $R(S)$ drawn from $R(\emptyset)$ to $R(\Omega)$, which give reasonable initial approximations.

4. Relationship with Prior Art

In this section, we discuss relationships between our method and other previous methods [2, 25, 9].

4.1. Auxiliary Cuts (AC) [2] and pPBC [25]

The authors of [2] derived a bound for $R_{type1}(S)$ by using the Jensen's inequality and assuming $S \subseteq S^t$. Its essential form is given as

$$A_\alpha(S|S^t) = B_{shr}(S|S^t) + \alpha R(S^t)\left(1 - \frac{\langle 1, S \rangle}{\langle 1, S^t \rangle}\right), \quad (17)$$

where $\alpha \geq 0$ is a parameter and $B_{shr}(S|S^t)$ is expressed as

$$B_{shr}(S|S^t) = R(\emptyset) + \frac{R(S^t) - R(\emptyset)}{\langle g, S^t \rangle}\langle g, S \rangle. \quad (18)$$

From Eq. (18), $B_{shr}(S|S^t)$ can be seen as the linear approximations of $R(S)$ drawn from $R(\emptyset)$ to $R(S^t)$. See Fig. 4 (c), where the solid red line visualizes this linear approximation bound, the infinite bound reflects the restriction $S \subseteq S^t$, and the dotted line depicts the effect of the α-term in Eq. (17). The use of this bound results in successively *shrinking* segments: $S^0 \supseteq S^1 \supseteq S^2$. pPBC [25] extends AC by exhaustively searching the best $\alpha \in (-\infty, \infty)$ in each iteration using parametric maxflow [15].

To point out a relationship to our method, we use another bound $B_{exp}(S|S^t)$ for $R(S)$, which is similar to $B_{shr}(S|S^t)$ and can be derived by [2]'s derivations as

$$B_{exp}(S|S^t) = R(S^t) + \frac{R(\Omega) - R(S^t)}{\langle g, \Omega \setminus S^t \rangle}\langle g, S \setminus S^t \rangle. \quad (19)$$

Here, S^t is restricted to $S^t \subseteq S$, so the iterative minimization of $B_{exp}(S|S^t)$ successively *expands* the segments S.

This bound can be seen as the linear approximation of $R(S)$ drawn from $R(S^t)$ to $R(\Omega)$.

We now show that these bounds $B_{shr}(S|S^t)$ and $B_{exp}(S|S^t)$ can be derived using our grouped permutation bounds. When a grouped permutation is given as $\pi(1) = S^t$ and $\pi(2) = \bar{S}^t$ with $\bar{S}^t := \Omega \setminus S^t$, our bound $H^\pi(S|S^t)$ becomes the following form:

$$H_{full}(S|S^t) = B_{shr}(S \cap S^t|S^t) + B_{exp}(S \cap \bar{S}^t|S^t). \quad (20)$$

Notice that it no longer requires the restrictions for S (*i.e.*, $S \subseteq S^t$ or $S^t \subseteq S$) used in [2, 25], which are turned out to be unnecessary by our derivation. Also notice that H_{full} does not depend on the permutation π. Although H_{full} does not accurately approximate its original function $R(S)$, it is still useful when permutations are inaccurate (*e.g.* at first iterations). Our method is designed to behave between AC [2] and SSP [19] and to produce coarse-to-fine approximation bounds as iterations proceed, by using the monotonically decreasing grouping-threshold τ defined in Sec. 3.3.

4.2. Local Submodular Approximations (LSA) [9]

Very recently, an optimization method for pairwise non-submodular energies called LSA [9] has been proposed and shown to outperform other state-of-the-art methods such as TRW-S [14] and QPBO [16]. This method approximates non-submodular pairwise terms by linear terms. In [9], two types of approximation conversions are proposed. LSA-TR applies a Taylor-based approximation and uses the gradient-descent framework of FTR [10]. LSA-AUX uses a bound-based approximation and uses the bound optimization framework. The linear conversions of LSA-TR and LSA-AUX are summarized in Table 1.

As you can see from the table, our linear conversion includes the conversion of LSA-AUX. In fact, our conversion with a full-grouping π produces the same bounds with LSA-AUX. Furthermore, as mentioned in [9], there are other types of bounds that can be made by the permutation scheme of SSP [19]. Our method generalizes both conversions of SSP and LSA-AUX, and adaptively chooses either conversion for each term.

5. Experiments

For evaluation, we use four variations of our method: **SDC-GEO** is the proposed method described in Sec. 3. **SDC-DIST** uses the standard Euclidean distance for making permutations σ, but the other settings are the same with SDC-GEO. **SSP-DIST** is SSP [19] that follows the implementations by Rother *et al*. [22]. It is basically the same with SDC-DIST but uses no mean approximations for bound constructions. We also use **SSP-GEO**, which is the same with SSP-DIST but uses the geodesic distance. Note that when $S^0 = \Omega$ at the first iterations, we make permutations σ randomly for SSP-DIST and SSP-GEO, based on 10×10-pixels of patches as described in [22].

We also compare with three state-of-the-art methods for higher-order energies: **AC** [2] uses the bound A_α of Eq. (17). **pPBC** [25] uses the same bound but exhaustively chooses the best α using parametric maxflow [15]. **FTR** [10] is the state-of-the-art of gradient-descent methods. For pairwise energies, we compare with **LSA-AUX,-TR** [9] and **pPBC-T,-B,-L** [25] as state-of-the-art.

All the methods are implemented by C++, and run on a system with a 3.5GHz Core i7 CPU and 16GB RAM.

5.1. Segmentation via Distribution Matching

Similarly to [25, 2, 22], we evaluate the performances of the methods using the GrabCut dataset [23]. For pure evaluations of optimization performances, we learn the target histograms from the ground truth[1]. We use a standard 16-neighbor pairwise smoothness term similar to [23]: $Q(S) = \lambda \sum_{ij} \max(w_{ij}, \epsilon)|s_i - s_j|/|p_i - p_j|$ where p_i is pixel coordinates and $w_{ij} = \exp(-\beta|I_i - I_j|^2)$. Here, β is automatically estimated as $\beta = 2E[|I_i - I_j|^2]$ the expectation over all neighbor pairs. We use RGB-histograms.

Type-1: L_2 and L_1 Distance of Histograms

For Type-1 terms, we use the L_2 and L_1 distances between histograms. For $\{\lambda, \epsilon\}$, we use $\{1.0, 0.5\}$ and $\{0.5, 0.5\}$ for the L_2 and L_1-distances, respectively. S^t is trivially initialized as $S^0 \leftarrow \Omega$. For both SDC-GEO and SDC-DIST, we use a grouping threshold $\tau = (\mu + 8s)/(t + 1)^{2.5}$.

Tables 2 and 3 summarize the performance comparisons, showing average misclassified pixel rates, energy values of $E(S)$ and $R(S)$, running times, and individual-image comparisons with SDC-GEO. Among the seven methods, our proposed method SDC-GEO outperforms the others for all error and energy scores. In some cases, SDC-GEO completely outperforms pPBC, AC, FTR, and SSP-DIST for all individual images. Note that the error rates of SSP-DIST are better than the rates originally reported in [22] because

---

[1]If the target histograms are inaccurate, the minimum solutions of $E(S)$ are deviated from the ground truth [26], and the error rate criteria thus does not reflect the actual performances of the optimization methods.

Table 4. Evaluations on the GrabCut dataset [23] using the Bhattacharyya distance and KL divergence. We use 64^3 bins and the bounding box initialization.

Method	Error (%)		$E(S)$		Time (sec)	
	Bhat.	KL-div.	Bhat.	KL-div.	Bhat.	KL-div.
SDC	**0.373**	**0.515**	**-14906**	**7025**	19.8	12.9
pPBC [25]	0.498	<u>0.818</u>	-14870	7057	<u>1.8</u>	<u>1.5</u>
AC [2]	18.29	16.07	-11878	7490	**0.4**	**0.4**
FTR [10]	<u>0.435</u>	1.076	<u>-14894</u>	<u>7049</u>	2.9	6.4

the definition of pixel feature histograms is different[2]. Comparing the results of SDC-DIST and SSP-DIST with L_2 and 64^3 bins, SDC-DIST finds more accurate segmentations in spite of the higher energies. This is because in SSP-DIST the appearance consistencies are forced regardless of how permutations σ and corresponding bounds are inaccurate, resulting in highly non-smooth, visibly bad local minimas. In Fig. 5, we show example results of L_2 and L_1 with 64 bins. Figures 6 and 7 show the plots of the accuracy transitions using the L_2 and L_1-distances w.r.t. the number of bins. As shown, SDC-GEO is robust to the difference of the number of bins. In order to show robustness to initialization, we compare SDC-GEO, pPBC, and FTR using two types of initialization shown in Fig. 1. Unlike pPBC (and AC) that can only reduce the target regions, our method is robust to initialization and finds very accurate solutions even for such difficult camouflage images.

Type-2: Bhattacharyya Distance and KL Divergence

We use Bhattacharyya distance and KL divergence as Type-2. As described in Sec. 3.2, we use the pseudo bound optimization scheme of [25] using parametric maxflow [15] for our method SDC-GEO. We show the performance comparisons with pPBC, AC, and FTR in Table 4, where our method outperforms the others in error and energy values. It is worth noting that although SSP [19] is originally oriented for supermodular functions, our extended method successfully optimizes non-supermodular functions (Type-2).

5.2. Type-3: Image Deconvolution

We make a blurred image $\tilde{I}$ by a mean filter and additive Gaussian noises: $\tilde{I}_i = \frac{1}{9}\sum_{j \in W_i} I_j + \mathcal{N}(0, \sigma^2)$, where W_i is a 3×3 window centered at i. We recover the original image I by minimizing $E(S) = \sum_{i \in \Omega}(\tilde{I}_i - \frac{1}{9}\sum_{j \in W_i} s_j)^2$. In Fig. 8, we show example results of our method, pPBC [25], and LSA [9] for two noise levels $\sigma = 0.15, 0.30$. In Fig. 9, we show the plots of energies, squared errors ($\sum_i |s_i - I_i|^2$), and running times. LSA-TR and pPBC-T,-L reach the lower energies but inaccurate results. Our method and LSA-AUX perform best in terms of both accuracy and efficiency.

---

[2][22] uses a normalized 2D color vector and texton as a pixel feature, since the method is intended for co-segmentation and image retrieval.

Table 2. Evaluations on the GrabCut dataset [23] using L_2-distance. We show average error rates, $E(S)$, $R(S)$, times over 50 images. The last column shows the number of images for which the proposed method (**SDC-GEO**) outperforms each method. We use 192^3 and 64^3 bins.

Method (**L_2-distance**)	Error (%)		$E(S)$		$R(S)$		Time (sec)		SDC-GEO vs	
	192^3	64^3	192^3	64^3	192^3	64^3	192^3	64^3	192^3	64^3
ref. Ground Truth	0	0	3569	3569	0	0	-	-	-	-
SDC-GEO	**0.095**	**0.288**	**3511**	**3809**	<u>179</u>	**235**	6.2	11.7	-	-
SDC-DIST	0.134	<u>1.123</u>	4121	18922	577	9946	2.3	<u>2.7</u>	32	46
SSP-GEO	<u>0.116</u>	1.132	<u>3594</u>	<u>6477</u>	**166**	<u>377</u>	4.9	15.1	24	45
SSP-DIST [22]	0.312	2.575	4415	13969	295	877	<u>2.0</u>	4.2	32	50
pPBC [25]	1.062	2.677	19381	190332	13187	176315	36.7	24.6	49	50
AC [2]	1.214	3.542	19888	195729	13646	177947	**1.0**	**1.0**	50	50
FTR [10]	1.859	3.167	21003	153212	17669	146394	36.8	121.2	50	48

Table 3. Evaluations on the GrabCut dataset [23] using L_1-distance. See Table 2 for the descriptions.

Method (**L_1-distance**)	Error (%)		$E(S)$		$R(S)$		Time (sec)		SDC-GEO vs	
	192^3	64^3	192^3	64^3	192^3	64^3	192^3	64^3	192^3	64^3
ref. Ground Truth	0	0	1785	1785	0	0	-	-	-	-
SDC-GEO	**0.033**	**0.205**	**1804**	**1882**	54	**120**	5.4	9.4	-	-
SDC-DIST	<u>0.041</u>	<u>0.242</u>	<u>1813</u>	<u>1967</u>	70	<u>229</u>	1.9	3.8	38	42
SSP-GEO	0.043	0.943	<u>1813</u>	2766	**31**	298	3.7	9.7	28	46
SSP-DIST [22]	0.075	1.341	1898	3666	<u>50</u>	411	<u>1.6</u>	<u>2.4</u>	34	48
pPBC [25]	0.154	0.583	2013	2632	309	997	17.4	18.9	48	48
AC [2]	0.339	1.281	2502	4336	762	2476	**0.6**	**0.6**	50	50
FTR [10]	0.147	0.366	1908	2105	277	495	46.1	100.7	46	41

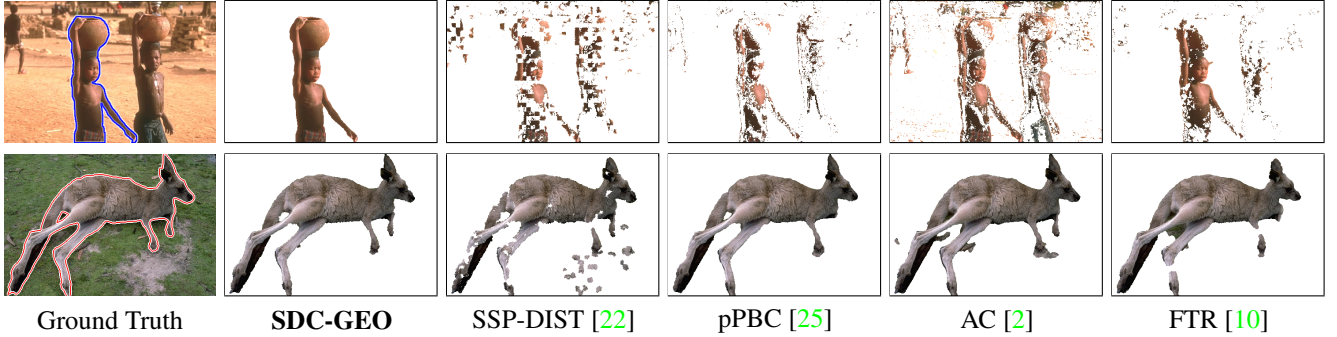| Ground Truth | **SDC-GEO** | SSP-DIST [22] | pPBC [25] | AC [2] | FTR [10] |

Figure 5. L_2 (top) and L_1 (bottom) examples. The segments are initialized as all foreground, and 64^3 bins of histograms are used.
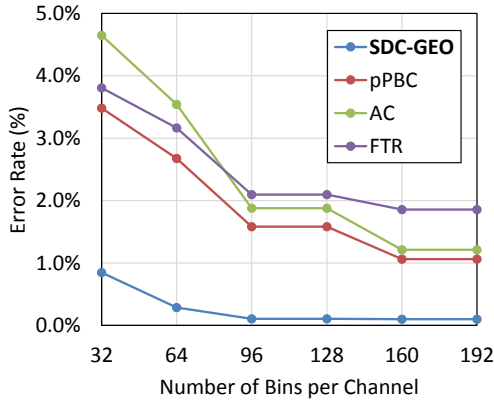

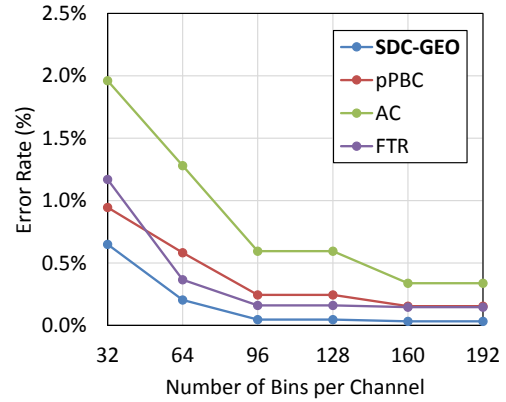Figure 6. Error rate transitions w.r.t. the number of bins (L_2)

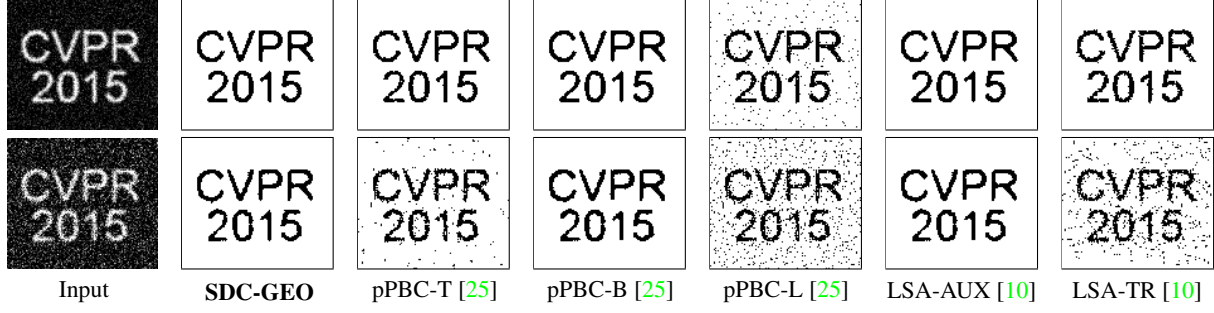Figure 7. Error rate transitions w.r.t. the number of bins (L_1)

Figure 8. Image deconvolution results for two images with noise levels of $\sigma = 0.15$ (top) and $\sigma = 0.30$ (bottom).
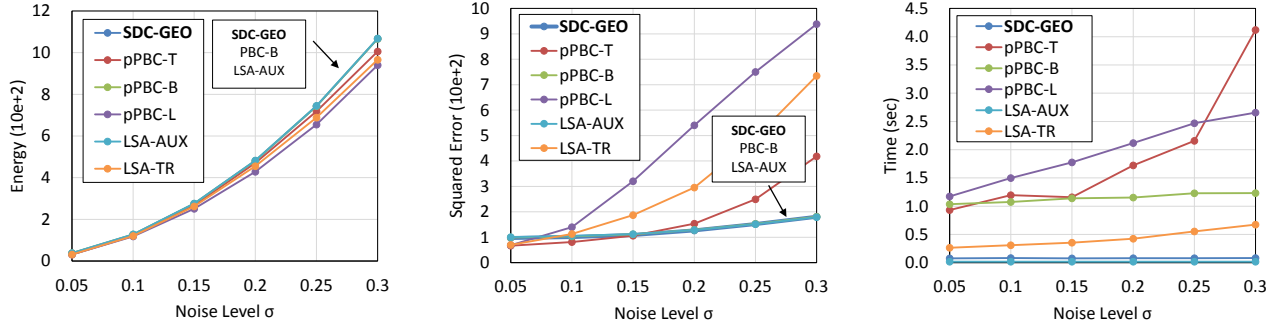

Figure 9. Performance comparisons of image deconvolution. Convergence energies, square errors, and running times w.r.t. noise levels are shown. The values are averaged over 30 random noise images at each point. We use $\tau = (\mu + s)/(t+1)^{2.5}$ for our method. Notice that our method obtains more accurate solutions in spite of their higher energies, because our course-to-fine scheme avoids bad local minimums.
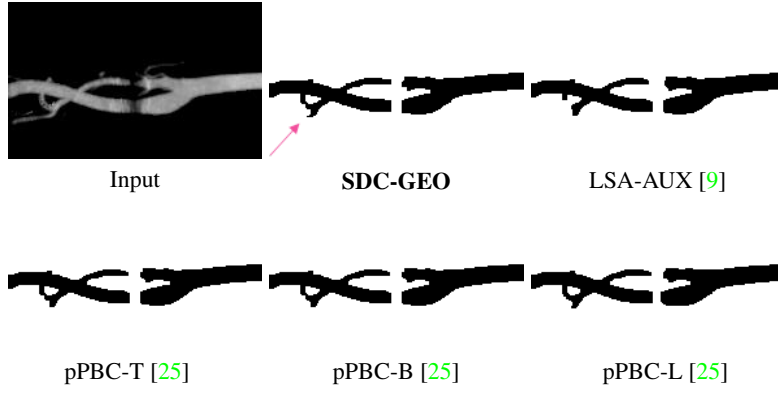

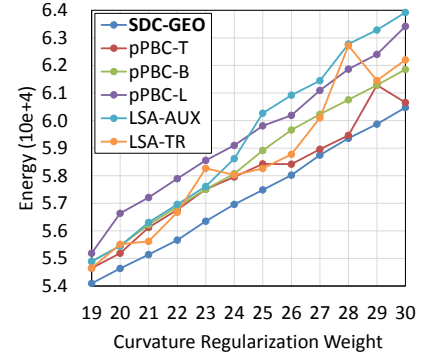Figure 10. Results of curvature optimization at the weight of 25.

Figure 11. Convergence energies w.r.t. curvature regularization weights. We use $\tau = \mu/(t+1)^{2.5}$.

5.3. Type-3: Curvature Regularization

We apply our method to a curvature regularization model of [6] in image binarization. We show the input image and results by our method, pPBC [25], and LSA [9] in Fig. 10. The plots in Fig. 11 show energies at convergence w.r.t. regularization weights. Our method always reaches the lowest energies and is most stable among all methods.

6. Conclusions

In this paper we have revisited SSP [19], an early approach to higher-order energy optimization, and pro-

posed our method as generalization of SSP. The key idea of our method is piecewise mean approximation bounds, which are designed to produce coarse-to-fine approximation bounds during iterations. We further show that our method has close connections to some state-of-the-art methods [2, 25, 9]. Although the proposed method shows promising improvements over state-of-the-art methods [25, 10, 9, 2], we would like to further push the envelope by improving the definition of geodesic distance and the thresholding scheme for making grouped permutations.

Acknowledgments

References

[1] I. B. 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 *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3288–3295, 2010. 1, 2

[2] I. B. Ayed, L. Gorelick, and Y. Boykov. Auxiliary Cuts for General Classes of Higher Order Functionals. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1304–1311, 2013. 1, 2, 3, 4, 5, 6, 7, 8

[3] Y. Boykov and M. P. Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 105–112, 2001. 1

[4] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 26:1124–1137, 2004. 1, 2, 4

[5] A. Criminisi, T. Sharp, and A. Blake. GeoS: Geodesic Image Segmentation. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 99–112, 2008. 3, 4

[6] N. El-Zehiry and L. Grady. Fast Global Optimization of Curvature. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3257–3264, 2010. 1, 8

[7] S. Fujishige. *Submodular Functions and Optimization*, volume 58. Elsevier Science, 2005. 2, 3

[8] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 6(6):721–741, 1984. 1

[9] L. Gorelick, Y. Boykov, O. Veksler, I. Ben Ayed, and A. Delong. Submodularization for Binary Pairwise Energies. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2, 5, 6, 8

[10] L. Gorelick, F. R. Schmidt, and Y. Boykov. Fast Trust Region for Segmentation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1714–1721, 2013. 1, 2, 5, 6, 7, 8

[11] L. Gorelick, F. R. Schmidt, Y. Boykov, A. Delong, and A. D. Ward. Segmentation with Non-Linear Regional Constraints via Line-Search Cuts. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 583–0597, 2012. 1, 2

[12] R. Iyer, S. Jegelka, and J. Bilmes. Fast Semidifferential-based Submodular Function Optimization. *Proc. of International Conference on Machine Learning (ICML)*, 28(3):855–863, 2013. 2

[13] H. Jiang. Linear Solution to Scale Invariant Global Figure Ground Separation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 678–685, 2012. 2

[14] V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 28(10):1568–1583, 2006. 5

[15] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of Parametric Maxflow in Computer Vision. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2007. 4, 5, 6

[16] V. Kolmogorov and C. Rother. Minimizing Nonsubmodular Functions with Graph Cuts – A Review. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 29(7):1274–1279, 2007. 2, 5

[17] V. Kolmogorov and R. Zabin. What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 26(2):147–159, 2004. 1, 2

[18] L. Mukherjee, V. Singh, and C. Dyer. Half-Integrality based Algorithms for Cosegmentation of Images. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2028–2035, 2009. 2

[19] M. Narasimhan and J. A. Bilmes. A Submodular-Supermodular Procedure with Applications to Discriminative Structure Learning. In *Proc. of Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 404–412, 2005. 1, 2, 3, 5, 6, 8

[20] V.-Q. Pham, K. Takahashi, and T. Naemura. Foreground-Background Segmentation using Iterated Distribution Matching. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2113–2120, 2011. 1

[21] B. L. Price, B. Morse, and S. Cohen. Geodesic Graph Cut for Interactive Image Segmentation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3288–3295, 2010. 1

[22] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of Image Pairs by Histogram Matching–Incorporating a Global Constraint into MRFs. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 993–100, 2006. 1, 2, 4, 6, 7

[23] C. Rother, V. Kolmogorv, and A. Blake. GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts. *Proc. of SIGGRAPH (ACM Trans. on Graph.)*, 23:309–314, 2004. 1, 6, 7

[24] A. Schrijver. A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time. *J. of Combinational Theory Ser. B*, 80(2):346–355, 2000. 2

[25] M. Tang, I. Ben Ayed, and Y. Boykov. Pseudo-Bound Optimization for Binary Energies. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 691–707, 2014. 1, 2, 4, 5, 6, 7, 8

[26] T. Taniai, V.-Q. Pham, K. Takahashi, and T. Naemura. Image Segmentation using Dual Distribution Matching. In *Proc. of British Machine Vision Conf. (BMVC)*, pages 74.1–74.11, 2012. 1, 6

[27] P. J. Toivanen. New Geodesic Distance Transforms for Grayscale Images. *Pattern Recogn. Lett.*, 17(5):437–450, 1996. 4

[28] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation Revisited: Models and Optimization. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 465–479, 2010. 2