Local Submodularization
for Binary Pairwise Energies

Lena Gorelick, Yuri Boykov, Olga Veksler, Ismail Ben Ayed, Andrew Delong

Abstract—Many computer vision problems require optimization of binary non-submodular energies. We propose a general optimization framework based on *local submodular approximations* (LSA). Unlike standard LP relaxation methods that linearize the whole energy globally, our approach iteratively approximates the energy locally. On the other hand, unlike standard local optimization methods (e.g., gradient descent or projection techniques) we use non-linear submodular approximations and optimize them without leaving the domain of integer solutions. We discuss two specific LSA algorithms based on *trust region* and *auxiliary function* principles, LSA-TR and LSA-AUX. The proposed methods obtain state-of-the-art results on a wide range of applications such as binary deconvolution, curvature regularization, inpainting, segmentation with repulsion and two types of shape priors. Finally, we discuss a move-making extension to the LSA-TR approach. While our paper is focused on pairwise energies, our ideas extend to higher-order problems. The code is available online.

Index Terms—Discrete optimization, graph cuts, trust region, auxiliary functions, local submodularization.

--- ◆ ---

1 INTRODUCTION

W E address a general class of binary pairwise non-submodular energies, which are widely used in applications like segmentation, stereo, inpainting, deconvolution, and many others. Without loss of generality, the corresponding binary energies can be transformed into the form[1]

$$E(S) = S^T U + S^T M S, \qquad S \in \{0,1\}^\Omega \qquad (1)$$

where $S = (s_p \in \{0,1\} \,|\, p \in \Omega)$ is a vector of binary indicator variables defined on pixels $p \in \Omega$, vector $U = (u_p \in \mathcal{R} \,|\, p \in \Omega)$ represents unary potentials, and symmetric matrix $M = (m_{pq} \in \mathcal{R} \,|\, p, q \in \Omega)$ represents pairwise potentials. Note that in many practical applications matrix M is sparse since elements $m_{pq} = 0$ for all non-interacting pairs of pixels. We seek solutions to the following integer quadratic optimization problem

$$\min_{S \in \{0,1\}^\Omega} E(S). \qquad (2)$$

When energy (1) is *submodular*, i.e., $m_{pq} \leq 0 \;\; \forall (p,q)$, globally optimal solution for (2) can be found in a low-order polynomial time using graph cuts [1]. The general non-submodular case of problem (2) is NP hard.

1.1 Standard linearization methods

Integer quadratic programming is a well-known challenging class of optimization problems with extensive literature in the combinatorial optimization community, e.g., see [1], [2], [3]. It often

(a) global linearization   (b) local linearization

Fig. 1. Standard linearization approaches for (1)-(2). Black dots are integer points and ∗ corresponds to the global optimum of (2). Colors in (b) show iso-levels of the quadratic energy (1). This energy can be linearized by introducing additional variables and linear constraints, see a schematic polytope in (a) and [6]. Vector ∇E is the gradient of the global linearization of (1) in (a) and the gradient of the local linear approximation of (1) at point S_0 in (b).

appears in computer vision where it can be addressed with many methods including spectral and semi-definite programming relaxations, e.g., see [4], [5].

Methods for solving (2) based on LP relaxations, e.g., QPBO [7] and TRWS [8], are considered among the most powerful in computer vision [9]. They approach integer quadratic problem (2) by *global linearization* of the objective function at a cost of introducing a large number of additional variables and linear constraints. These methods attempt to optimize the relaxed LP or its dual. However, the integer solution can differ from the relaxed solution circled in Fig.1(a). This is a well-known *integrality gap* problem. Most heuristics for extracting an integer solution from the relaxed solution have no *a priori* quality guarantees.

Our work is more closely related to *local linearization* techniques for approximating (2), e.g., parallel ICM, IPFP [10], and other similar methods [11]. Parallel ICM iteratively linearizes energy $E(S)$ around current solution S_0 using Taylor expansion and makes a step by computing an integer minimizer S_{int} of the corresponding linear approximation, see Fig.1(b). However,

- L. Gorelick, Y. Boykov, O. Veksler are with the Department of Computer Science, University of Western Ontario, London, Canada. Andrew Delong is with the Department of Electrical Engineering, University of Toronto, Toronto, Canada.
  E-mail: lenagorelick@gmail.com
- Ismail Ben Ayed is with Departement de génie de la production automatisée, École de Technologie Supérieure, Montréal, Quebec, Canada
- Andrew Delong is with the Department of Electrical Engineering, University of Toronto, Toronto, Canada.

1. Note that such transformations are up to a constant, see Sec. 3.1.

similarly to Newton's methods, this approach often gets stuck in bad local minima by making too large steps regardless of the quality of the approximation. IPFP attempts to escape such minima by reducing the step size. It explores the continuous line between integer minimizer S_{int} and current solution S_0 and finds optimal relaxed solution S_{rlx} with respect to the original quadratic energy. Similarly to the global linearization methods, see Fig.1(a), such continuous solutions give no quality guarantees with respect to the original integer problem (2).

1.2 Overview of submodularization

Linearization is a popular approximation approach to integer quadratic problem (1)-(2), but it often requires relaxation leading to the integrality gap problem. We propose a different approximation approach that we call *submodularization*. The main idea is to use submodular approximations of energy (1). We propose several approximation schemes that keep submodular terms in (1) and linearize non-submodular potentials in different ways leading to different optimization algorithms. Standard *truncation* of non-submodular pairwise terms[2] and some existing techniques for high-order energies [12], [13], [14], [15] can be seen as *submodularization* examples, as discussed later. Common properties of submodularization methods is that they compute globally optimal integer solution of the approximation and do not leave the domain of discrete solutions avoiding integrality gaps. Sumbodularization can be seen as a generalization of local linearization methods since it uses more accurate higher-order approximations.

One way to linearize non-submodular terms in (1) is to compute their Taylor expansion around current solution S_0. Taylor's approach is similar to IPFP [10], but they linearize all terms including submodular ones. In contrast to IPFP, our overall approximation of $E(S)$ at S_0 is not linear; it belongs to a more general class of submodular functions. Such non-linear approximations are more accurate while still permitting efficient optimization in the integer domain.

We also propose a different mechanism for controlling the step size. Instead of exploring relaxed solutions on continuous interval $[S_0, S_{int}]$ in Fig.1, (b), we obtain discrete candidate solution S by minimizing local submodular approximation over $\{0,1\}^{\Omega}$ under additional distance constraint $||S - S_0|| < d$. Thus, our approach avoids integrality gap issues. For example, even linear approximation model in Fig.1, (b) can produce solution S^* if Hamming distance constraint $||S - S_0|| \leq 1$ is imposed. This local submodularization approach to (1)-(2) fits a general *trust region framework* [4], [12], [16], [17]. We call it LSA-TR.

Another way to linearize the non-submodular terms in (1) is based on the general *auxiliary function* framework [13], [15], [18][3]. Instead of Taylor expansion, non-submodular terms in $E(S)$ are approximated by linear upper bounds specific to current solution S_0. Combining them with submodular terms in $E(S)$ gives a submodular upper-bound approximation, a.k.a. an *auxiliary function*, for $E(S)$ that can be globally minimized within the integer domain. This approach does not require to control the step size as the global minimizer of an auxiliary function is guaranteed to decrease the original energy $E(S)$. We refer to this type of local submodular approximation approach as LSA-AUX.

Recently both *trust region* [4], [16], [17] and *auxiliary function* [18] frameworks proved to work well for optimization of energies with high-order regional terms [12], [15]. They derive specific linear [12] or upper bound [15] approximations for non-linear cardinality potentials, KL and other distances between segment and target appearance models. To the best of our knowledge, we are the first to develop trust region and auxiliary function methods for integer quadratic optimization problems (1)-(2).

In the context of multilabel energy minimization, there is a series of works [19], [20], [21] that overestimate the intractable energy with a tractable modified version within a move making framework. Interestingly, instead of using linear (modular) upper bounds as in [13], [15], they change pairwise or higher-order terms to achieve a submodular upper bound. Their approach is iterative due to the move-making strategy for multi-label energy optimization, and would converge in a single step if reduced to our binary energy. In contrast, our approach is designed for binary energies and is iterative by definition.

In the context of binary high-order energies, more related to our to work are the auxiliary functions proposed in [13], [15]. In [15], Jensen inequality was used to derive linear upper bounds for several important classes of high-order terms that gave practically useful results. Their approach is not directly applicable to our energy, as it is not clear which continuous function to use in the Jensen inequality for our discrete pairwise energy.

The work in [13] is most related to ours. They divide the energy into submodular and supermodular parts and replace the latter with a certain permutation-based linear upper-bound. The corresponding auxiliary function allows polynomial-time solvers. However, experiments in [14] (Sec. 3.2) demonstrated limited accuracy of the permutation-based bounds [13] on high-order segmentation problems. Our LSA-AUX method is first to apply auxiliary function approach to arbitrary (non-submodular) pairwise energies. We discuss possible linear upper bounds for pairwise terms and study several specific cases. One of them corresponds to the permutation bounds [13] and is denoted by LSA-AUX-P. Recently [22] propose a generalization of [13] for higher order binary energies. In the pairwise case, their approach is equivalent to LSA-AUX-P.

In [23] they relax the upper-bound condition and replace it with a family of pseudo-bounds, which can better approximate the original energy. According to their evaluation, LSA-TR performs better than their approach in most cases.

Our contributions can be summarized as follows:
- A general *submodularization* framework for solving integer quadratic optimization problems (1)-(2) based on *local submodular approximations* (LSA). Unlike global linearization methods, LSA constructs an approximation model without additional variables. Unlike local linearization methods, LSA uses a more accurate approximation.
- In contrast to the majority of standard approximation methods, LSA works strictly within the domain of discrete solutions and requires no rounding.
- We develop move making extension to the LSA approach, which can perform better on difficult energies.
- We propose a novel *Generalized Compact* shape prior that requires optimization of binary non-submodular energy.
- State-of-the-art results on a wide range of applications. Our LSA algorithms outperform QPBO, LBP, IPFP, TRWS, its latest variant SRMP, and other standard techniques for (1)-(2).

---

2. Truncation is known to give low quality results, e.g. Fig.4, Tab.1.

3. *Auxiliary functions* are also called *surrogate functions* or *upper bounds*. The corresponding approximate optimization technique is also known as the *majorize-minimize* principle [18].

2 DESCRIPTION OF LSA ALGORITHMS

In this section we discuss our framework in detail. Section 2.1 derives local submodular approximations and describes how to incorporate them in the trust region framework. Section 2.2 briefly reviews auxiliary function framework and shows how to derive local auxiliary bounds.

2.1 LSA-TR

Trust region methods are a class of iterative optimization algorithms. In each iteration, an approximate model of the optimization problem is constructed near the current solution S_0. The approximation is assumed to be accurate only within a small region around the current solution called "trust region". The approximate model is then globally optimized within the trust region to obtain a candidate solution. This step is called *trust region sub-problem*. The size of the trust region is adjusted in each iteration based on the quality of the current approximation. For a review of TR framework see [17].

Below we provide details of our trust region approach to the binary pairwise energy optimization (see pseudo-code in Algorithm 1). The goal is to minimize $E(S)$ in (1). This energy can be decomposed into submodular and supermodular parts $E(S) = E^{sub}(S) + E^{sup}(S)$ such that

$$\begin{aligned} E^{sub}(S) &= S^T U + S^T M^- S \\ E^{sup}(S) &= S^T M^+ S \end{aligned}$$

where matrix M^- with negative elements $m_{pq}^- \leq 0$ represents the set of submodular pairwise potentials and matrix M^+ with positive elements $m_{pq}^+ \geq 0$ represents supermodular potentials. Given the current solution S_t energy $E(S)$ can be approximated by submodular function

$$E_t(S) = E^{sub}(S) + S^T U_t + const \qquad (3)$$

where $U_t = 2M^+ S_t$. The last two terms in (3) are the first-order Taylor expansion of supermodular part $E^{sup}(S)$.

While the use of Taylor expansion may seem strange in the context of functions of integer variables, Fig. 2, (a,b) illustrates its geometric motivation. Consider individual pairwise supermodular potentials $f(x, y)$ in

$$E^{sup}(S) = \sum_{pq} m_{pq}^+ \cdot s_p s_q = \sum_{pq} f_{pq}(s_p, s_q).$$

Coincidentally, Taylor expansion of each relaxed supermodular potential $f(x, y) = \alpha \cdot xy$ produces a linear approximation (planes in b) that agrees with f at three out of four possible discrete configurations (points A,B,C,D).

The standard trust region sub-problem is to minimize approximation E_t within the region defined by step size d_t

$$S^* = \underset{||S-S_t|| < d_t}{argmin} E_t(S). \qquad (4)$$

Hamming, L_2, and other useful metrics $||S - S_t||$ can be represented by a sum of unary potentials [24]. However, optimization problem (4) is NP-hard even for unary metrics[4]. One can solve Lagrangian dual of (4) by iterative sequence of graph cuts as in [25], but the corresponding duality gap may be large and the optimum for (4) is not guaranteed.

---

4. By a reduction to the *balanced cut* problem.

Instead of (4) we use a simpler formulation of the trust region subproblem proposed in [12]. It is based on unconstrained optimization of submodular Lagrangian

$$L_t(S) = E_t(S) + \lambda_t \cdot ||S - S_t|| \qquad (5)$$

where parameter λ_t controls the trust region size indirectly. Each iteration of LSA-TR solves (5) for some fixed λ_t and adaptively changes λ_t for the next iteration (Alg.1 line 27), as motivated by empirical inverse proportionality relation between λ_t and d_t discussed in [12].

Once a candidate solution S^* is obtained, the quality of the approximation is measured using the ratio between the actual and predicted reduction in energy. Based on this ratio, the solution is updated in line 24 and the step size (or λ) is adjusted in line 27. It is common to set the parameter τ_1 in line 24 to zero, meaning that any candidate solution that decreases the actual energy gets accepted. The parameter τ_2 in line 27 is usually set to 0.25 [17]. Reduction ratio above this value corresponds to good approximation model allowing increase in the trust region size.

Algorithm 1: GENERAL TRUST REGION APPROACH

1 **Initialize** $t = 0$, $S_t = S_{\text{init}}$, $\lambda_t = \lambda_{\text{init}}$, convergedFlag $= 0$
2 **While** !convergedFlag
3   //**Approximate** $E(S)$ **around** S_t
4     $E_t(S) = E^{sub}(S) + S^T U_t$ as defined in (3)
5   //**Solve Trust Region Sub-Problem**
6     $S^* \longleftarrow argmin_{S \in \{0,1\}^\Omega} L_t(S)$ // as defined in (5)
7   //**Evaluate Reduction in Energy**
8     $P = E_t(S_t) - E_t(S^*)$ //predicted reduction in energy
9     $R = E(S_t) - E(S^*)$ //actual reduction in energy
10   **If** $P = 0$ // meaning $S^* = S^t$ and $\lambda > \lambda_{\text{max}}$
11   //**Try smallest discrete step possible**
12         $\lambda_t \longleftarrow \lambda_{\text{max}}$
13   //**Solve Trust Region Sub-Problem**
14         $S^* \longleftarrow argmin_{S \in \{0,1\}^\Omega} L_t(S)$ // as defined in (5)
15   //**Evaluate Reduction in Energy**
16         $P = E_t(S_t) - E_t(S^*)$ //predicted reduction in energy
17         $R = E(S_t) - E(S^*)$ //actual reduction in energy
18   //**Update current solution**
19         $S_{t+1} \longleftarrow \begin{cases} S^* & \text{if } R/P > \tau_1 \\ S_t & \text{otherwise} \end{cases}$
20   //**Check Convergence**
21   convergedFlag $\longleftarrow (R \leq 0)$
22   **Else**
23   //**Update current solution**
24         $S_{t+1} \longleftarrow \begin{cases} S^* & \text{if } R/P > \tau_1 \\ S_t & \text{otherwise} \end{cases}$
25   **End**
26   //**Adjust the trust region**
27     $\lambda_{t+1} \longleftarrow \begin{cases} \lambda_t/\alpha & \text{if } R/P > \tau_2 \\ \lambda_t \cdot \alpha & \text{otherwise} \end{cases}$
28 **End**

In each iteration of the trust region, either the energy decreases or the trust region size is reduced. When the trust region is so small that it does not contain a single discrete solution, namely $S^* = S^t$ (Line 10), one more attempt is made using λ_{max}, where $\lambda_{\text{max}} = \sup \{\lambda | S^* \neq S_t\}$ (see [12]). If there is no reduction in energy with smallest discrete step λ_{max} (Line 21), we are at a local minimum [26] and we stop.

2.2 LSA-AUX

Bound optimization techniques are a class of iterative optimization algorithms constructing and optimizing upper bounds, a.k.a.

(a) Supermodular potential $\alpha \cdot xy$ (b) "Taylor" based local linearization (c) Upper bound linearization
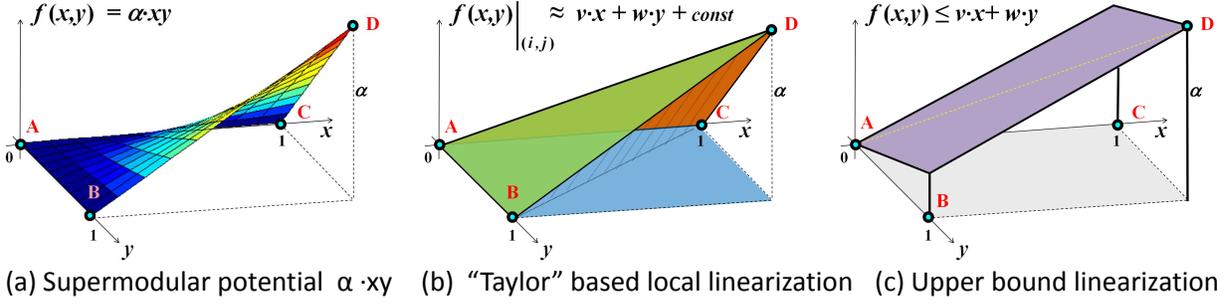
Fig. 2. Local linearization of supermodular pairwise potential $f(x,y) = \alpha \cdot xy$ for $\alpha > 0$. This potential defines four costs $f(0,0) = f(0,1) = f(1,0) = 0$ and $f(1,1) = \alpha$ at four distinct configurations of binary variables $x, y \in \{0, 1\}$. These costs can be plotted as four 3D points A, B, C, D in (a-c). We need to approximate supermodular potential f with a linear function $v \cdot x + w \cdot y + const$ (plane or unary potentials). **LSA-TR:** one way to derive a local linear approximation is to take Taylor expansion of $f(x,y) = \alpha \cdot xy$ over relaxed variables $x, y \in [0, 1]$, see the continuous plot in (a). At first, this idea may sound strange since there are infinitely many other continuous functions that agree with A, B, C, D but have completely different derivatives, e.g., $g(x,y) = \alpha \cdot x^2 \sqrt{y}$. However, Taylor expansions of **bilinear** function $f(x,y) = \alpha \cdot xy$ can be motivated geometrically. As shown in (b), Taylor-based local linear approximation of f at any fixed integer configuration (i,j) (e.g., . blue plane at A, green at B, orange at C, and striped at D) coincides with discrete pairwise potential f not only at point (i,j) but also with two other closest integer configurations. Overall, each of those planes passes exactly through three out of four points A, B, C, D. **LSA-AUX:** another approach to justify a local linear approximation for non-submodular pairwise potential f could be based on upper bounds passing through a current configuration. For example, the green or orange planes in (b) are the tightest linear upper bounds at configurations $(0,1)$ and $(1,0)$, correspondingly. When current configuration is either $(0,0)$ or $(1,1)$ then one can choose either orange or green plane in (b), or anything in-between, e.g., the purple plane passing though A and D in (c).

auxiliary functions, for energy E. It is assumed that those bounds are easier to optimize than the original energy E. Given a current solution S_t, the function $A_t(S)$ is an auxiliary function of E if it satisfies the following conditions:

$$E(S) \leq A_t(S) \tag{6a}$$
$$E(S_t) = A_t(S_t) \tag{6b}$$

To approximate minimization of E, one can iteratively minimize a sequence of auxiliary functions:

$$S_{t+1} = \arg \min_S A_t(S), \quad t = 1, 2, \ldots \tag{7}$$

Using (6a), (6b), and (7), it is straightforward to prove that the solutions in (7) correspond to a sequence of decreasing energy values $E(S_t)$. Namely,

$$E(S_{t+1}) \leq A_t(S_{t+1}) \leq A_t(S_t) = E(S_t).$$

The main challenge in bound optimization approach is designing an appropriate auxiliary function satisfying conditions (6a) and (6b). However, in case of integer quadratic optimization problem (1)-(2), it is fairly straightforward to design an upper bound for non-submodular energy $E(S) = E^{sub}(S) + E^{sup}(S)$. As in Sec.2.1, we do not need to approximate the submodular part E^{sub} and we can easily find a linear upper bound for E^{sup} as follows.

Similarly to Sec.2.1, consider supermodular pairwise potentials $f(x,y) = \alpha \cdot xy$ for individual pairs of neighboring pixels according to

$$E^{sup}(S) = \sum_{pq} m_{pq}^+ \cdot s_p s_q = \sum_{pq} f_{pq}(s_p, s_q) \tag{8}$$

where each f_{pq} is defined by scalar $\alpha = m_{pq}^+ > 0$. As shown in Fig. 2, (b,c), each pairwise potential f can be bound above by linear function $u(x,y)$

$$f(x,y) \leq u(x,y) := v \cdot x + w \cdot y$$

for some positive scalars v and w. Assuming current solution $(x,y) = (x^t, y^t)$, the table below specifies linear upper bounds (planes) for four possible discrete configurations

(x^t, y^t)	upper bound $u(x,y)$	plane in Fig.2(b,c)
(0,0)	$\frac{\alpha}{2}x + \frac{\alpha}{2}y$	purple
(0,1)	αx	green
(1,0)	αy	orange
(1,1)	$\frac{\alpha}{2}x + \frac{\alpha}{2}y$	purple

We denote the approach that uses bounds in the table above as LSA-AUX. As clear from Fig.2, (b,c), there are many other possible linear upper bounds for pairwise terms f. Interestingly, the "permutation" approach to high-order supermodular terms in [13] reduces to linear upper bounds for $f(x,y)$ where each configuration (0,0) or (1,1) selects either orange or green plane randomly (depending on a permutation). We denote permutation based upper bounds LSA-AUX-P. Our tests showed inferior performance of such bounds for pairwise energies compared to LSA-AUX on most applications. The upper bounds using purple plane for (0,0) and (1,1), as in the table, work better in practice.

Summing upper bounds for all pairwise potentials f_{pq} in (8) using linear terms in this table gives an overall linear upper bound for supermodular part of energy (1)

$$E^{sup}(S) \leq S^T U_t \tag{9}$$

where vector $U_t = \{u_p^t | p \in \Omega\}$ consists of elements

$$u_p^t = \sum_q \frac{m_{pq}^+}{2}(1 + s_q^t - s_p^t)$$

and $S_t = \{s_p^t | p \in \Omega\}$ is the current solution configuration for all pixels. Defining our auxiliary function as

$$A_t(S) := S^T U_t + E^{sub}(S) \tag{10}$$

and using inequality (9) we satisfy condition (6a)

$$E(S) = E^{sup}(S) + E^{sub}(S) \leq A_t(S).$$

Since $S_t^T U_t = E^{sup}(S_t)$ then our auxiliary function (10) also satisfies condition (6b)

$$E(S_t) = E^{sup}(S_t) + E^{sub}(S_t) = A_t(S_t).$$
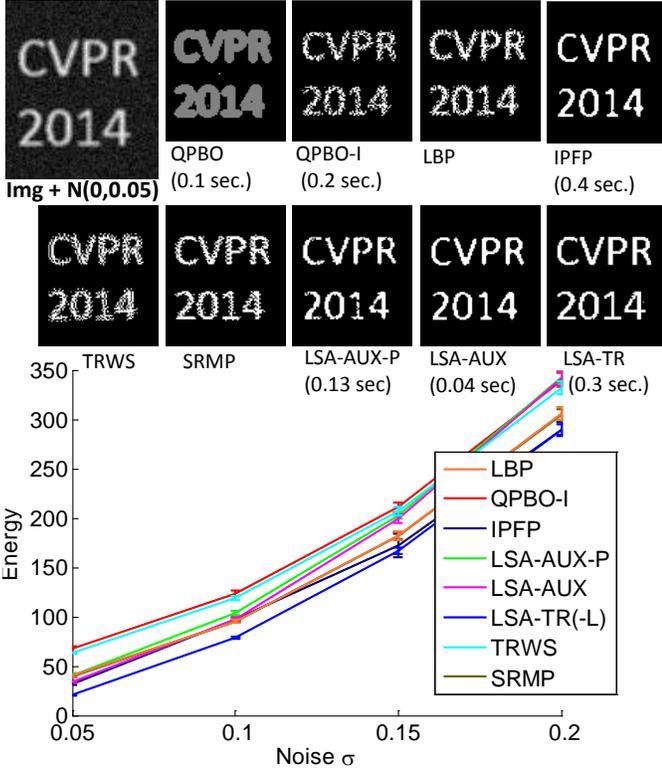
Fig. 3. Binary deconvolution of an image created with a uniform 3×3 filter and additive Gaussian noise ($\sigma \in \{0.05, 0.1, 0.15, 0.2\}$). No length regularization was used. We report mean energy (+/-2std.) and time as a function of noise level σ. TRWS, SRMP and LBP are run for 5000 iterations.

Function $A_t(S)$ is submodular. Thus, we can globally optimize it in each iteration guaranteeing an energy decrease.

3 APPLICATIONS

Below we apply our method in several applications such as binary deconvolution, segmentation with repulsion, curvature regularization, inpainting and two different shape priors, one of which is a novel contribution by itself. We report results for both LSA-TR and LSA-AUX frameworks and compare to existing state of the art methods such as QPBO [7], LBP [27], IPFP [10], TRWS and SRMP [8] in terms of energy and running time[5]. For the sake of completeness, and to demonstrate the advantage of non-linear submodular approximations over linear approximations, we also compare to a version of LSA-TR where both submodular and supermodular terms are linearized, denoted by LSA-TR-L.

In the following experiments, all local approximation methods, e.g., IPFP, LSA-AUX, LSA-AUX-P, LSA-TR, LSA-TR-L are initialized with the entire domain assigned to the foreground. All global linearization methods, e.g., TRWS, SRMP and LBP, are run for 50, 100, 1000 and 5000 iterations. For QPBO results, unlabeled pixels are shown in gray color. Running time is shown in log-scale for clarity. Our preliminary experiments showed inferior performance of the Hamming distance compared to Euclidean. See example in Sec. 3.5. Therefore, we used L_2 distance for all the experiments below.

5. We used *http://pub.ist.ac.at/~vnk/software.html* code for SRMP and *www.robots.ox.ac.uk/~ojw* code for QPBO, TRWS, and LBP. The corresponding version of LPB is sequential without damping.
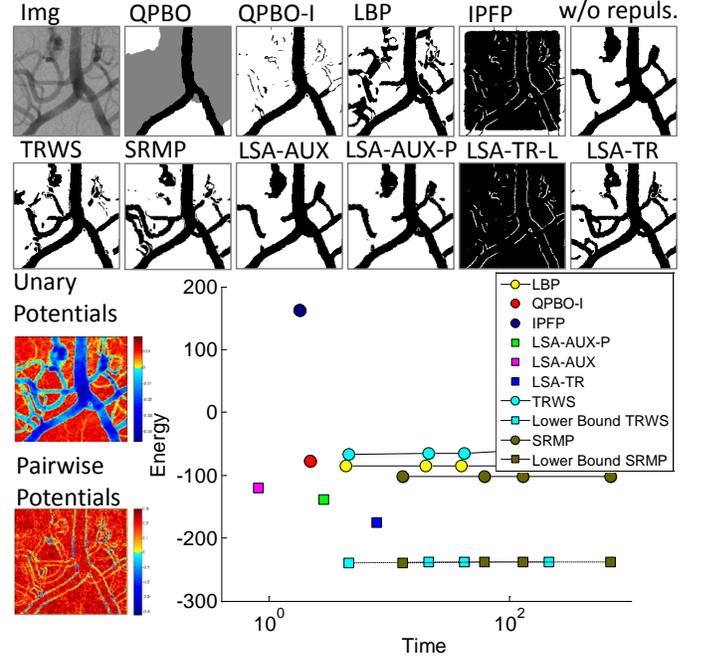
Fig. 4. Segmentation with repulsion and attraction. We used μ_{fg}=0.4, μ_{bg}=0.6, σ=0.2 for appearance, λ_{reg}=100 and c=0.06. Repulsion potentials are shown in blue and attraction - in red.

3.1 Energy Transformation

For some applications, instead of defining the energy as in (1), it is more convenient to use the following form:

$$E(S) = \sum_{p \in \Omega} D_p(s_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(s_p, s_q), \qquad (11)$$

where D_p is the unary term, V_{pq} is the pairwise term and $\mathcal{N}$ is a set of ordered neighboring pairs of variables. We now explain how to transform the energy in (11) to the equivalent form in (1).

Transformation of the unary terms D_p results in a linear term (i.e., vector) $J = (j_p | p \in \Omega)$, where $j_p = D_p(1) - D_p(0)$.

Let the pairwise terms $V_{pq}(s_p, s_q)$ be as follows:

s_p	s_q	V_{pq}
0	0	a_{pq}
0	1	b_{pq}
1	0	c_{pq}
1	1	d_{pq}

Transformation of the pairwise terms V_{pq} results in two linear terms H, K one quadratic term M and a constant. Term H accumulates for each variable p all V_{pq} in which p is the first argument. That is,

$$H = (h_p | p \in \Omega), \text{where } h_p = \sum_{(p,q) \in \mathcal{N}} (c_{pq} - a_{pq}).$$

Term K does the same for the second argument of V_{pq}. That is,

$$K = (k_q | q \in \Omega), \text{where } k_q = \sum_{(p,q) \in \mathcal{N}} (b_{pq} - a_{pq}).$$

We define quadratic term M in (1) as $m_{pq} = a_{pq} - b_{pq} - c_{pq} + d_{pq}$.

Letting $U = J + H + K$ and M as defined above, it is easy to show that the energy in (11) can be written in the form of (1) up to a constant $C = \sum_p D_p(0) + \sum_{(p,q) \in \mathcal{N}} a_{pq}$.
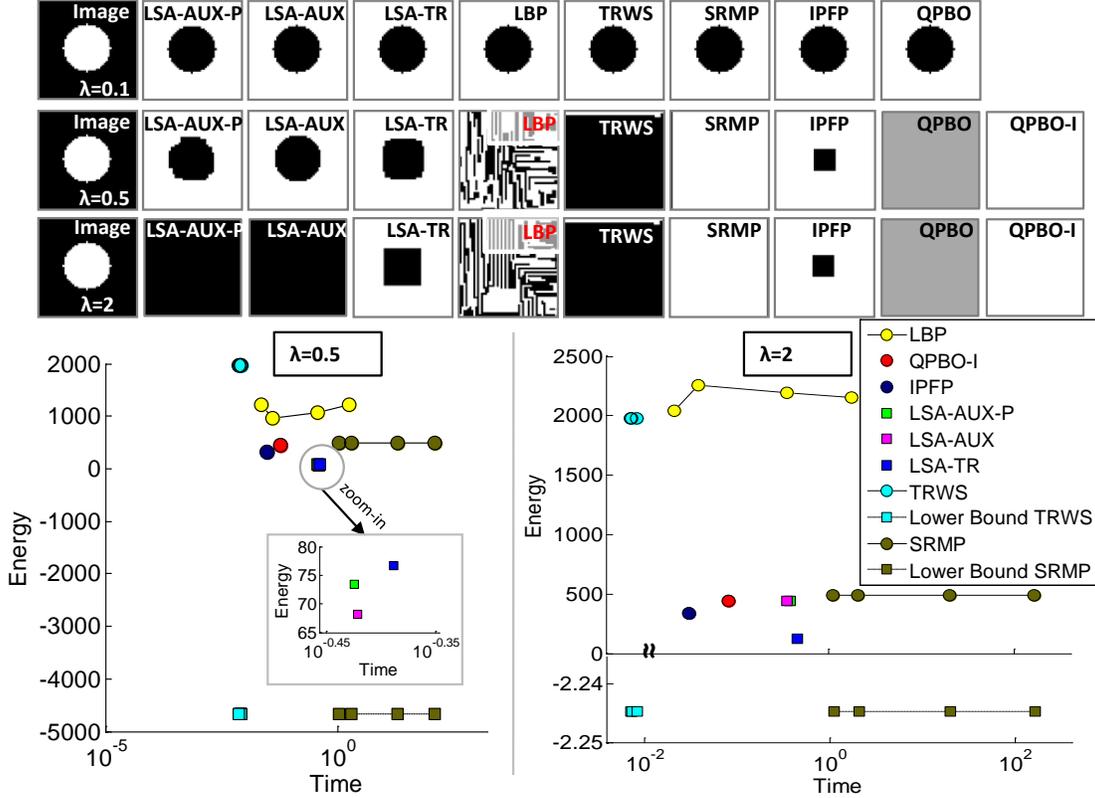
Fig. 5. Curvature regularizer [28] is more difficult to optimize when regularizer weight is high. We show segmentation results for $\lambda_{curv} = 0.1$ (top row), $\lambda_{curv} = 0.5$ (middle row), $\lambda_{curv} = 2$ (bottom row) as well as energy plots. We used $\mu_{fg} = 1$, $\mu_{bg} = 0$, $\lambda_{app} = 1$.

3.2 Binary Deconvolution

Figure 3, (top-left) shows a binary image after convolution with a uniform 3×3 combined with Gaussian noise ($\sigma = 0.05$). The goal of binary deconvolution is to recover the original binary image. The energy is defined as

$$E(S) = \sum_{p \in \Omega} (I_p - \frac{1}{9} \sum_{q \in \mathcal{N}_p} s_q)^2 \qquad (12)$$

Here $\mathcal{N}_p$ denotes the 3×3 neighborhood window around pixel p and all pairwise interactions are supermodular. We did not use length regularization, since it would make the energy easier to optimize. Figure 3 compares the performance of LSA-TR and LSA-AUX to standard optimization methods such as QPBO, LBP, IPFP, TRWS and SRMP. In this case LSA-TR-L and LSA-TR are identical since energy (12) has no submodular pairwise terms. The bottom of Fig. 3 shows the mean energy as a function of noise level σ. For each experiment the results are averaged over ten instances of random noise. The mean time is reported for the experiments with $\sigma = 0.05$.

3.3 Segmentation with Repulsion

In this section we consider segmentation with attraction and repulsion pairwise potentials. Adding repulsion is similar to correlation clustering [29] and multiway cut [30], where data points either attract or repulse each other. Using negative repulsion in segmentation can avoid the bias of submodular length regularizer to *short-cutting*, whereby elongated structures are shortened to avoid high length penalty. Figure 4 (top-left) shows an example

of an angiogram image with elongated structures. We use 16-neighborhood system and the pairwise potentials are defined as follows:

$$\omega(p, q) = \frac{-\Delta(p, q) + c}{\text{dist(p,q)}}.$$

Here dist(p,q) denotes the distance between image pixels p and q and $\Delta(p, q)$ is the difference in their respective intensities (see pairwise potentials in Fig. 4, bottom-left). The constant c is used to make neighboring pixels with similar intensities attract and repulse otherwise. Being supermodular, repulsions potentials make the segmentation energy more difficult to optimize, but are capable to extract thin elongated structures. To demonstrate the usefulness of "repulsion" potentials we also show segmentation results with graph-cut a la Boykov-Jolly [31] where negative pairwise potentials were removed/truncated (top-right).

3.4 Curvature

Below we apply our optimization method to curvature regularization. We focus on the curvature model proposed in [28]. The model is defined in terms of 4-neighborhood system and accounts for 90 degrees angles. In combination with appearance terms, the model yields discrete binary energy that has both submodular and non-submodular pairwise potentials. Originally, the authors of [28] proposed using QPBO for optimization of the curvature regularizer. We show that our method significantly outperforms QPBO and other state-of-the-art optimization techniques, especially with large regularizer weights.

First, we deliberately choose a toy example (white circle on a black background, see Fig. 5), where we know what an optimal

solution should look like. With the 4-neighborhood system, as the weight of the curvature regularizer increases, the solution should minimize the number of 90 degrees angles (corners) while maximizing the appearance terms. Therefore, when the weight of curvature regularizer is high, the solution should look more like a square than a circle. Consider the segmentation results in Fig. 5. With low curvature weight, $\lambda_{curv} = 0.1$, all compared methods perform equally well (see top row). In this case appearance data terms are strong compared to the non-submodular pairwise terms. However, when we increase the curvature weight and set $\lambda_{curv} = 0.5$ or 2 there is a significant difference between the optimization methods both in terms of the energy and the resulting solutions (see Fig. 5 middle and bottom).

Next, we selected an angiogram image example from [28] and evaluate the performance[6] of the optimization methods with two values of regularizer weight $\lambda_{curv} = 19$ and $\lambda_{curv} = 21$ (see Fig. 6). Although the weight λ did not change significantly, the quality of the segmentation deteriorated for all global linearization methods, namely QPBO, TRWS, LBP. The proposed methods LSA-TR and LSA-AUX seem to be robust with respect to the weight of the supermodular part of the energy.

3.5 Chinese Characters Inpainting

Below we consider the task of in-painting in binary images of Chinese characters, *dtf-chinesechar* [9]. We used a set of pre-trained unary and pairwise potentials provided by the authors with the dataset. While each pixel variable has only two possible labels, the topology of the resulting graph and the non-submodularity of its pairwise potentials makes this problem challenging. Figure 7 shows two examples of inpainting. Table 1 reports the performance of our LSA-TR and LSA-AUX methods on this problem and compares to other standard optimization methods reported in [9], as well as, to *truncation* of non-submodular terms. LSA-TR is ranked second, but runs four orders of magnitudes faster.

3.6 Segmentation of Multi-Region Objects

Many objects can be described by a combination of spatially coherent and visually distinct regions. Such objects can often be segmented using multi-label segmentation framework, where a separate appearance-boundary model is maintained for each label. Recently a multi-label segmentation model has been proposed in [32] that uses a separate binary graph layer for each label and allows encoding many useful geometric interactions between different parts of an object. For example inclusion of an object part within another part while enforcing a minimal margin around the interior part is modeled using submodular pairwise interactions between corresponding nodes in different layers. Exclusion constraints are in general supermodular.

In this section we focus on one particular example of multi-part model designed for segmentation of liver on an MRI image, see Fig. 9, (a-left). The image contains one foreground object (liver) with four distinct mutually exclusive interior parts (tumors). Below we formally define the energy for our model using the form in (11). To convert this energy to the form in (1) see details in Sec. 3.1.

Given an image with N pixels, we construct a graph with five layers of binary variables. The layers correspond to liver

6. For QPBO, we only run QPBO-I and do not use other post-processing heuristics as suggested in [28], since the number of unlabeled pixel might be significant when the regularization is strong.

Alg. Name	Mean Runtime	Mean Energy	#best /100	Rank
MCBC	2053.89 sec	-49550.1	85	1
BPS (LBP)*	72.85 sec	-49537.08	18	3
ILP	3580.93 sec	-49536.59	8	6
QPBO	0.16 sec	-49501.95	0	8
SA	NaN sec	-49533.08	13	4
TRWS	0.13 sec	-49496.84	2	7
TRWS-LF	2106.94 sec	-49519.44	11	5
Truncation	0.06 sec	-16089.2	0	9
LSA-AUX	0.30 sec	-49515.95	0	9
LSA-AUX-P	0.16 sec	-49516.63	0	9
LSA-TR(Euc.)	0.21 sec	-49547.61	35	2
LSA-TR(Ham.)	0.23 sec	-49536.76	1	8

TABLE 1
Chinese characters in-painting database [9]. We tested three methods (at the bottom) and compared with other techniques (above) reported in [9]. * - To the best of our knowledge, BPS in [9] is the basic sequential version of loopy belief-propagation without damping that we simply call LBP in this paper.

(Fg), and four tumors (A, B, C, D). Each layer has N nodes and each node has a corresponding binary variable. We use a standard Potts regularization on each layer to account for boundary length between object parts. In addition we employ pairwise inclusion and exclusion constraints between the layers to enforce correct geometric interactions between the different parts of the object, see Fig. 8, (b). Finally we derive unary terms for the binary variables so that they correspond to the correct multilabel appearance energy term.

Each graph node p has three coordinates (r_p, c_p, l_p) and a corresponding binary variable s_p. The first two coordinates denote the row and column of the corresponding pixel in the image (top-left corner as origin) and the last coordinate l_p denotes the layer of the node, $l_p \in \{\text{Fg, A, B, C, D}\}$.

For length regularization, we use 8-neighborhood system within each layer and the pairwise potentials are defined as follows. Let p, q be neighboring nodes in some layer $l \in \{\text{A, B, C, D}\}$, then

$$V_{p,q}^1(s_p, s_q) = \lambda_{\text{Potts}} \frac{-\Delta(p,q)}{\text{dist(p,q)}} \cdot [s_p \neq s_q].$$

Here $\text{dist(p,q)} = \sqrt{(r_p - r_q)^2 + (c_p - c_q)^2}$ denotes the distance between the corresponding image pixels in the image domain, $\Delta(p,q)$ is the distance between in their respective colors in the RGB color space and λ_{Potts} is the weight.

Next, we explain how to implement inclusion and exclusion constraints, see Fig. 8, (b). Let p and q be two nodes corresponding to the same pixel such that node p is in the liver (Fg) layer and node q is in a tumor layer. That is $(r_q = r_p) \wedge (c_q = c_p)$ and $(l_p = \text{Fg}) \wedge (l_q \in \{\text{A, B, C, D}\})$. Inclusion pairwise potential $V_{p,q}^2$ forces any interior tumor part to be geometrically inside the foreground object by penalizing configuration $(0,1)$ for the corresponding nodes p, q. That is

$$V_{p,q}^2(s_p, s_q) = \lambda_{sub} \cdot \begin{cases} \infty & \text{if } (s_p, s_q) = (0,1) \\ 0 & \text{otherwise.} \end{cases}$$
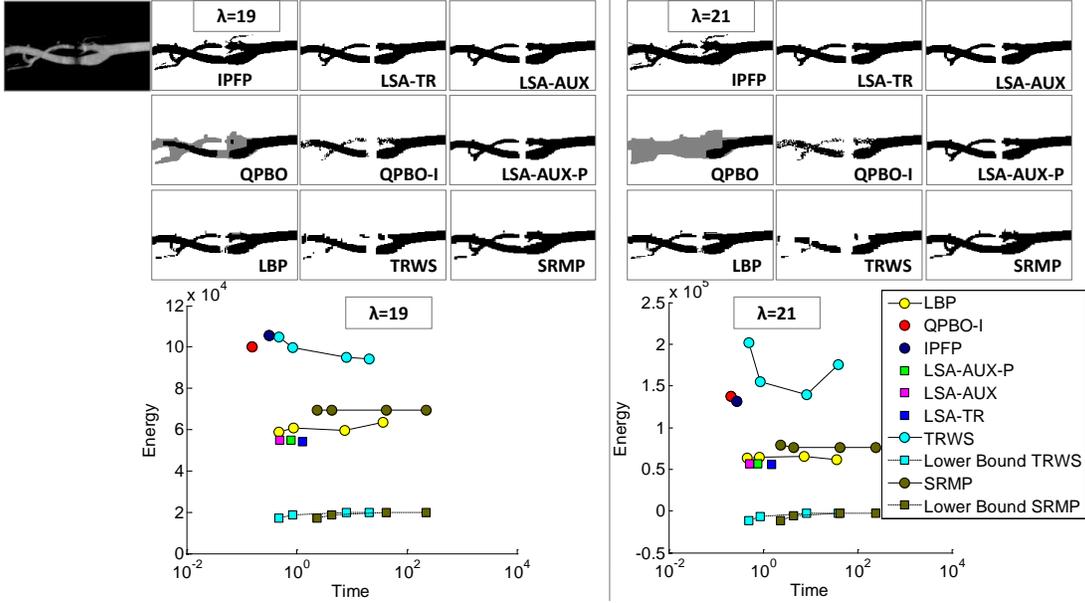
Fig. 6. Curvature regularizer [28]: we show segmentation results and energy plots for λ_{curv}=19 (left), λ_{curv}=21 (right).

Fig. 7. Examples of Chinese characters inpainting.

The tumor parts are mutually exclusive, see Fig. 8, (b). Let p and q be two nodes corresponding to the same image pixel but in different tumor layers. That is $(r_q = r_p) \wedge (c_q = c_p)$ and $l_p \neq l_q$ where $l_p, l_q \in \{A, B, C, D\}$. Then the supermodular exclusion pairwise potential $V_{p,q}^3$ penalizes illegal configuration $(1,1)$. Each pixel can only belong to one tumor. That is,

$$V_{p,q}^3(s_p, s_q) = \lambda_{sup} \cdot \begin{cases} \infty & \text{if } (s_p, s_q) = (1,1) \\ 0 & \text{otherwise.} \end{cases}$$

Since for each image pixel (r, c) we have five binary variables (one in each layer), there are 2^5 possible configurations of labels for each quintuple. However, our inclusion and exclusion constraints render most of the configurations illegal, i.e., having infinite cost. Figure 8, (c) summarizes all legal configurations for each quintuple of variables, their interpretation in terms of image segmentation and the respective multilabel appearance cost $D_{r,c}(l)$. Below, we define the unary terms D_p in (11) for our binary graph so that the binary energy corresponds to the multilabel energy in terms of appearance cost. Let $p = (r, c, l)$ be a node in our graph and let $D_{r,c}(l)$ be the multilabel appearance term at image pixel (r, c) for label l. Then,

$$D_p(s_p) = \begin{cases} D_{r_p,c_p}(\text{Fg}) & \text{if } l_p = \text{Fg} \wedge s_p = 1 \\ D_{r_p,c_p}(\text{Bg}) & \text{if } l_p = \text{Fg} \wedge s_p = 0 \\ D_{r_p,c_p}(l_p) - D_{r_p,c_p}(\text{Fg}) & \text{if } l_p \in \{A, B, C, D\} \\ & \wedge s_p = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 8. Multi-part object model for liver segmentation: (a) schematic representation of the liver containing four distinct and mutually excluding tumors. (b) each part of the object is represented with a separate binary layer in the graph. Each image pixel has a corresponding node in all five layers, resulting in a quintuple (FG, A, B, C, D). Interactions between corresponding nodes of different layers are shown with black solid lines for inclusion and blue dashed lines for exclusion. (c) summarizes six legal configurations for each pixel's quintuple and the associated multi-label cost. All other configurations have an infinite cost due to inclusion or exclusion violations.

If each pixel's quintuples is labeled with legal configuration, the unary appearance term on our graph is equal to the multilabel appearance term for image pixels.

Below we apply our multi-part object prior model in the task of multi-label segmentation of liver with tumors on an MRI image. Figure 9, (a) shows an input image containing one foreground object (liver) with four distinct interior parts (tumors). User scribbles are used to obtain appearance models for the liver and the tumors and as hard constraints. The liver is scribbled with the blue brush stroke and the tumors are scribbled with the green, cyan, yellow and magenta. Background is scribbled with the red color. While in theory our model has infinity constraints, in practice we need to select a finite weight for our submodular and supermodular

pairwise potentials. Here, we used $\lambda_{sub} = \lambda_{sup} = 100$ for the inclusion and exclusion terms respectively and $\lambda_{\text{Potts}} = 25$. For appearance we used histograms with 16 bins per color channel.

Figure 9, (b) shows segmentation results and compares different methods. For each compared method we show the final image segmentation, color coded as in the legend. We chose not to color the background pixels red for clarity, but rather leave them light gray. Dark gray pixels in QPBO denote pixels that were unlabeled at least in one of the five layers.

Figure 9, (c-right) compares the methods in terms of energy and the running time (shown in log-scale). The graph in (c-left) zooms in on the most interesting part of the plot. All the compared methods arrived at poor or very poor solutions that have violations of inclusion and exclusion constraints. This is due to the large number of the supermodular terms. LSA-TR achieves the lowest energy and the best segmentation.

3.7 Generalized Compact Shape Prior

In this section we propose a novel shape prior that is formulated as a multilabel energy and is subsequently reduced to a binary non-submodular pairwise energy using reduction similar to that in Sec. 3.6. Our new model generalizes *compact* shape prior proposed in [33]. Compact shape prior is useful in industrial part detection and medical image segmentation applications.

The compact shape prior in [33] assumes that an object can be partitioned into four quadrants around a given object center, provided by the user. Within each quadrant an object contour is either a monotonically decreasing or increasing function in the allowed direction for each quadrant. Figure 10, (a-top), shows an example of an object (along with user provided center) that can be segmented using the model in [33]. Allowed orientations for each quadrant are shown with blue arrows. We propose a more general model. It does not require user interaction, nor it assumes an object center, allowing for a larger class of object shapes.

Instead of dividing the whole object into four quadrants, our new model explicitly divides the background into four regions as in Fig. 10, (a-bottom), corresponding to four labels: top-left (TL), top-right (TR), bottom-left (BL), bottom-right (BR). There is an additional label for the foreground object (Fg). Each background label allows discontinuities only in certain orientation as is illustrated with the blue arrows. For example, the red region can have discontinuity only in the up-right orientation. Our model includes the model proposed in [33] as a special case when the transitions between different background labels are horizontally and vertically aligned as in (a-bottom). However, our model is more general because the discontinuities between the background regions do not need to align. For example, the object in (b-top) can be segmented using our model (b-bottom), but not the model in [33]. Below we formally define the energy for our model using the form in (11). To convert this energy to the form in (1) see details in Sec. 3.1.

Given an image with N pixels, we construct a graph with four binary layers: TL, TR, BL, BR. Each layer has N nodes and each node has a corresponding binary variable. Each layer is responsible for the respective region of the background and allows discontinuities only in a certain direction. In addition, there are also exclusion constraints between the layers to enforce a coherent foreground object, see Fig. 10, (c).

Each graph node p has three coordinates (r_p, c_p, l_p) and a corresponding binary variable s_p. The first two coordinates denote the row and column of the corresponding pixel in the image (top-left corner as origin) and the last coordinate denotes the layer of the node, $l \in \{TL, TR, BL, BR\}$.

There are two types of pairwise potentials in our model. The first type of potentials is defined between nodes within the same layer. It maintains the allowed orientation of the corresponding region boundary. For example, top-left layer TL allows switching from label 0 to 1 in the right and upward directions. Formally,

$$V_{pq}^{\text{TL}}(s_p, s_q) =$$
$$= \begin{cases} \infty & \text{if } (s_p, s_q) = (1,0) \wedge (r_q = r_p) \wedge (c_q = c_p + 1) \\ \infty & \text{if } (s_p, s_q) = (1,0) \wedge (r_q = r_p + 1) \wedge (c_q = c_p) \\ 0 & \text{otherwise.} \end{cases}$$

Similar intra-layer pairwise potentials are defined on the other three layers.

The other type of pairwise potentials is defined between corresponding nodes of different layers. They are responsible for exclusion constraint between the different background labels, see Fig. 10, (c). For example the red region (TL) in Fig. 10, (a-bottom) cannot overlap any of the other background regions (TR, BL, BR). Such pairwise potentials are super-modular.

Let p and q be two nodes corresponding to the same image pixel but in different graph layers. That is $(r_q = r_p) \wedge (c_q = c_p)$ and $l_p \neq l_q$ where $l_p, l_q \in \{TR, TL, BR, BL\}$. Then the supermodular exclusion pairwise potential $V_{p,q}^{\text{ex}}$ penalizes illegal configuration $(0,0)$. That is

$$V_{pq}^{\text{ex}}(s_p, s_q) = \begin{cases} \infty & \text{if } (s_p, s_q) = (0,0) \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

To interpret the optimal solution on our graph in terms of binary image segmentation, for each pixel we consider a quadruple of corresponding binary variables on layers TR, TL, BR and BL. We assign image pixel to foreground object (fg) if all its corresponding graph nodes have label one, and to the background (bg) otherwise, see table in Fig. 10, (d). As in [33], our model can incorporate any unary term in (11) defined on image pixels, e.g., appearance terms. We now define the corresponding unary terms on the nodes of our four layers binary graph.

Let $D_{r,c}(\text{fg})$ and $D_{r,c}(\text{bg})$ be the costs of assigning image pixel (r, c) to the foreground (fg) and background (bg) respectively. For each image pixel (r, c) we have a set of four corresponding graph nodes $\{p = (r_p, c_p, l_p) | (r_p = r) \wedge (c_p = c)\}$. All these nodes have the same unary term:

$$D_p(s_p) = \begin{cases} D_{r_p, c_p}(\text{fg}) & \text{if } s_p = 1 \\ D_{r_p, c_p}(\text{bg}) & \text{if } s_p = 0. \end{cases}$$

With the infinity constraints in our model, each image pixel (r, c) can have only two possible label configurations for the corresponding four graph nodes. It will either have three foreground and one background labels, in which case the image pixel is assigned to the background with a cost of $3 \cdot D_{r,c}(\text{fg}) + D_{r,c}(\text{bg})$. Or, all four nodes will have foreground labels, in which case the image pixel is assigned to the foreground with the cost of $4 \cdot D_{r,c}(\text{fg})$. In both cases, each image pixel will pay the additional constant cost of $3 \cdot D_{r,c}(\text{fg})$. This constant does not affect optimization.

Finally, we switch the meaning of zeros and ones for layers TR and BL. Labels 0 and 1 mean background and foreground in layers TR and BL and switch their meaning in layers TL and BR. While the switch is not necessary, it reduces the total
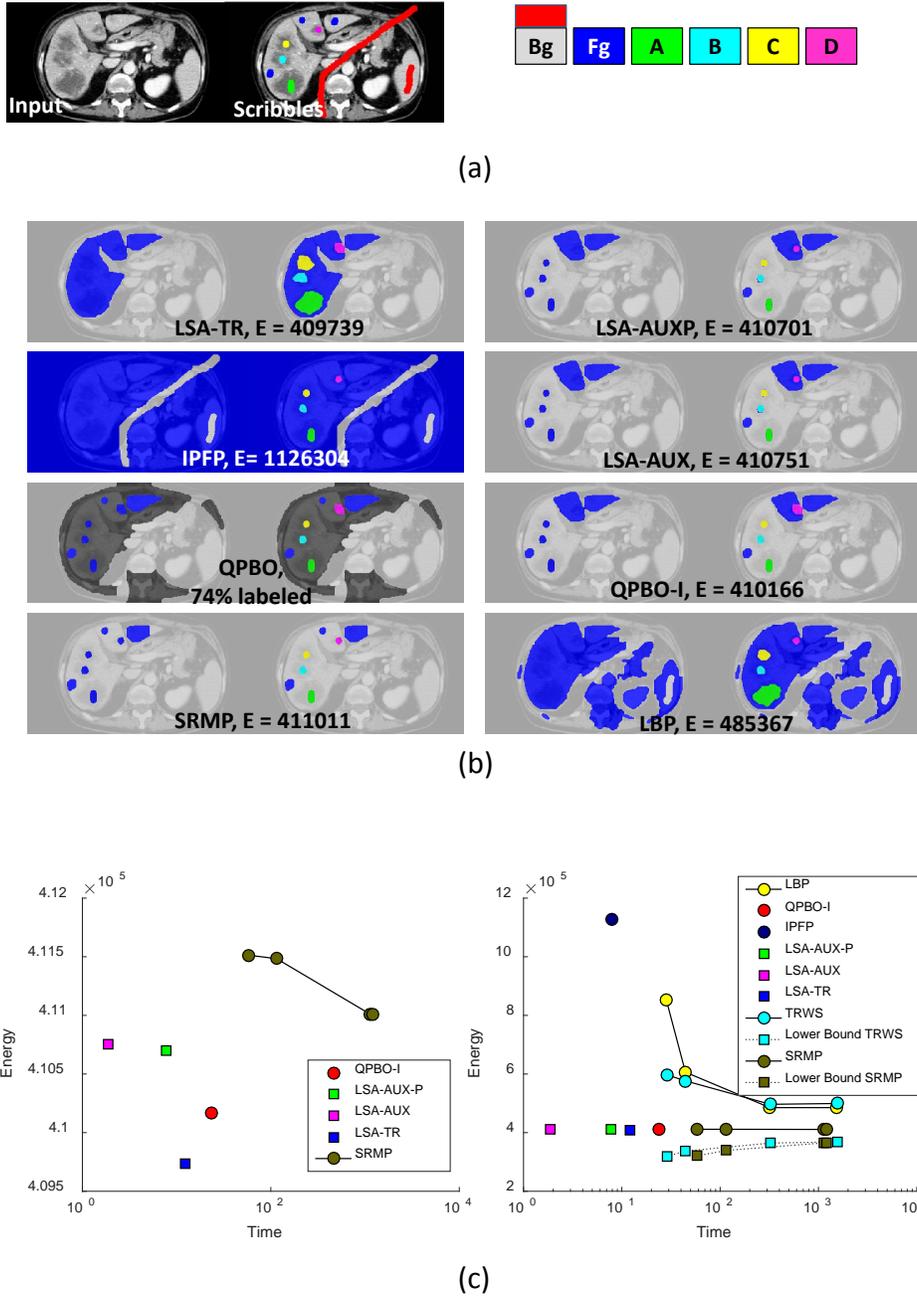
Fig. 9. Multi-part object model for liver segmentation - results and comparison. Top-left: input image containing one foreground object (liver) with four distinct interior parts (tumors). Top-right: appearance models for the liver and the tumors are obtained from user scribbles. The liver is scribbled with blue brush stroke and the tumors are scribbled with green, cyan, yellow and magenta. Background is scribbled with red color. For each method we show the resulting segmentation, color coded as in legend shown in top-right. (We chose not to color the background pixels red for clarity). Dark gray pixels in QPBO show pixels that were unlabeled at least in one of the layers.

number of supermodular terms V^{ex} in (13) to the one third of the original number. Note, that there is prior work on switching the meaning of binary variables to obtain better optimization, e.g., [1], [34], however there is no known algorithm for finding the optimal switching of labels in energies that are not permuted-submodular.

Our model has strong regularizing properties as it does not allow complex segmentation boundary. At the same time, due to zero costs in our intra-layer potentials, our model does not have a shrinking bias as opposed to the popular length based regularization models. This is similar to the lack of shrinking bias

in convexity shape prior [35]. The trade-off is that our model does not encourage alignment of the boundary with the image edges.

Below we apply our compact shape prior model in the task of binary image segmentation. Figure 11, (top-left) shows an example of an input image with a hot-air balloon. Below we show user scribbles and the resulting appearance terms. Blue colors denote preference for the background and cyan-red colors - preference for the foreground. Again, we replace infinity constraints of our submodular and supermodular pairwise potentials with a finite weight Here, we used $\lambda_{sub} = 250$ and $\lambda_{sup} = 500$ for the

Four Binary Labels per Pixel in Our Graph				Img Pixel Label	Total Unary Cost for Pixel (r,c): $D_{(r,c)}(.)$
TL	BL	TR	BR		
1	1	1	1	fg	$D_{r,c}(fg) + 3D_{(r,c)}(fg)$
0	1	1	1	bg (TL)	$D_{r,c}(bg) + 3D_{(r,c)}(fg)$
1	0	1	1	bg (BL)	$D_{r,c}(bg) + 3D_{(r,c)}(fg)$
1	1	0	1	bg (TR)	$D_{r,c}(bg) + 3D_{(r,c)}(fg)$
1	1	1	0	bg (BR)	$D_{r,c}(bg) + 3D_{(r,c)}(fg)$
Any other configuration					∞
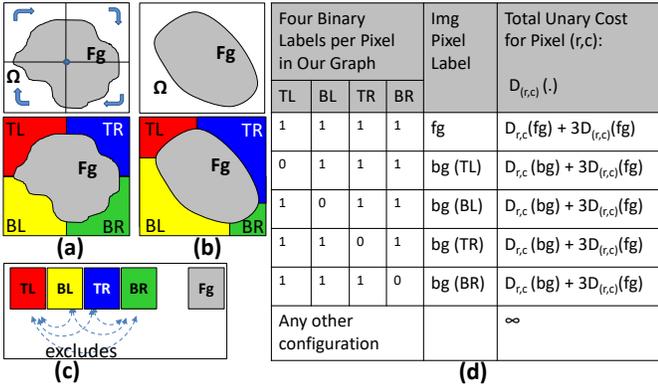
Fig. 10. Compact Shape Prior Illustration: (a-top) the model in [33], (a-bottom) our multilabel model, (b-top) - an input silhouette that can be modeled with our model but not with the model in [33] (see text for details), (b-bottom) demonstrates how we split the image into five regions in our new model, (c) schematic representation of the geometric exclusion constraints between the layers of our graph for our model. (d) unary terms for each layer used in our graph

submodular and supermodular terms respectively. To better illustrate the effect of using compact shape prior, in this experiment we did not utilize hard constraints on user scribbles. The optimization relies completely on the given appearance model and the compact shape prior. For each compared method, we show the final image segmentation along with the corresponding labeling on each of the four layers: TL, TR, BR, BL (clock-wise from top-left).

Figure 11, (bottom) compares the methods in terms of energy and the running time (shown in log-scale). Most of the methods arrived at poor or very poor solutions that have violations of monotonicity and coherence of the segment boundary. This is due to the high weight and large number of the supermodular terms. LSA-TR is the only method that could optimize such energy. It achieved the lowest energy and the most satisfying result.

3.8 Trust Region with Expansion Moves

We now suggest a move making extension for the LSA algorithms based on expansion moves [36]. While the extension is general, here we focus on LSA-TR due to its superior performance compared to LSA-AUX. We call this extension LSA-TR-EXP.

In move making optimization [36] one seeks a solution that is optimal only within a restricted search space around the current solution. Expansion moves restrict the search space in such a way that approximation of supermodular terms is more accurate. This is because many configurations for which our linear approximation is not exact are ruled out.

Given binary label $a \in \{0, 1\}$, an *a-expansion* move allows each binary variable s_p to either stay unchanged or switch to label a. Thus we have 0- and 1-expansion moves.

As described in Fig. 2 in each iteration of LSA-TR, each pairwise supermodular term $\alpha \cdot xy$ is linearized so that the approximation coincides with the original energy term on current configuration of (x, y) and two out of three remaining configurations. The green line in Fig. 12 specifies approximation for each of four possible current configurations (x, y). This approximation is used to evaluate all possible new configurations. There are four possible approximations are four possible new configurations, yielding in total 16 cases, see table in Fig. 12, gray section. LSA-TR computes approximation that is exact in twelve out of

sixteen cases. In contrast, during 0-expansion (pink section) or 1-expansion (blue section), only nine out of sixteen cases are valid moves. The advantage of the smaller search space is that the same approximation is now accurate in eight out of nine possible cases.

Similarly to other move making algorithms, LSA-TR-EXP starts with an initial solution and applies a sequence of 0- and 1-expansion until convergence. Each a-expansion is optimized with standard LSA-TR alg. 1. For simplicity, in a-expansion, we use hard constraint to prevent variables currently labeled with a from changing their label. More efficient implementation would exclude variables that are currently labeled with a from optimization.

Below we focus on the squared curvature regularization model proposed in [37]. The model is defined in terms of $n \times n$ neighborhood system, where larger n corresponds to higher angular resolution. In combination with appearance terms, the model yields discrete binary energy that has both submodular and non-submodular pairwise potentials. The weight of curvature term relative to appearance term is controlled by parameter λ_{curv}. In [37] they show that LSA-TR outperforms all other currently available optimization methods for non-submodular binary energies. Therefore, in this section we only compare LSA-TR with the proposed move making LSA-TR-EXP.

For this application we selected a synthetic image example where foreground object has an osculating contour. We vary the weight λ_{curv} and compare the performance of LSA-TR and LSA-TR-EXP. Figure 13 shows the input image in top-left and the comparison graph in top right. When the weight of supermodular curvature terms increases, LSA-TR-EXP (red line) consistently outperforms LSA-TR (blue line) when starting from the same initial solution. LSA-TR-EXP-improve (green line) attempts and often succeeds to improve the final solution of LSA-TR. The energy of the initial solution for each λ_{curv} is shown in black.

Figure. 13, bottom shows the final results of the three different methods for three different values of λ_{curv}. The red outline within each image denotes the final solution. Blue, red and green frames correspond to results of LSA-TR, LSA-TR-EXP and LSA-TR-EXP-improve respectively.

4 OPTIMIZATION LIMITATIONS

The proposed LSA-TR and LSA-AUX methods belong to a more general class of local iterative optimization and therefore can only guarantee a local minimum at convergence, see Sec. 2.1 and 2.2. Figure 14 demonstrates some sensitivity with respect to initialization. The trivial initialization with all pixels in the foreground, denoted by "init 1" and delineated by the red contour, leads to a poor local minimum. Using the appearance based maximum likelihood label per pixel as initialization, denoted by "init 2", results in a much lower optimum. From empirical observations, we obtain better results starting with appearance based maximum likelihood labels when possible.

5 CONCLUSIONS AND FUTURE WORK

We proposed two specific LSA algorithms based on *trust region* and *auxiliary function* principles. Our methods obtain state-of-the-art results on a wide range of applications that require optimization of binary non-submodular energies. Our methods outperform many standard techniques such as LBP, QPBO, and TRWS. In addition, we proposed a move-making extension to the LSA-TR approach. In the future, we plan to explore other variants of move making algorithms in combination with LSA.
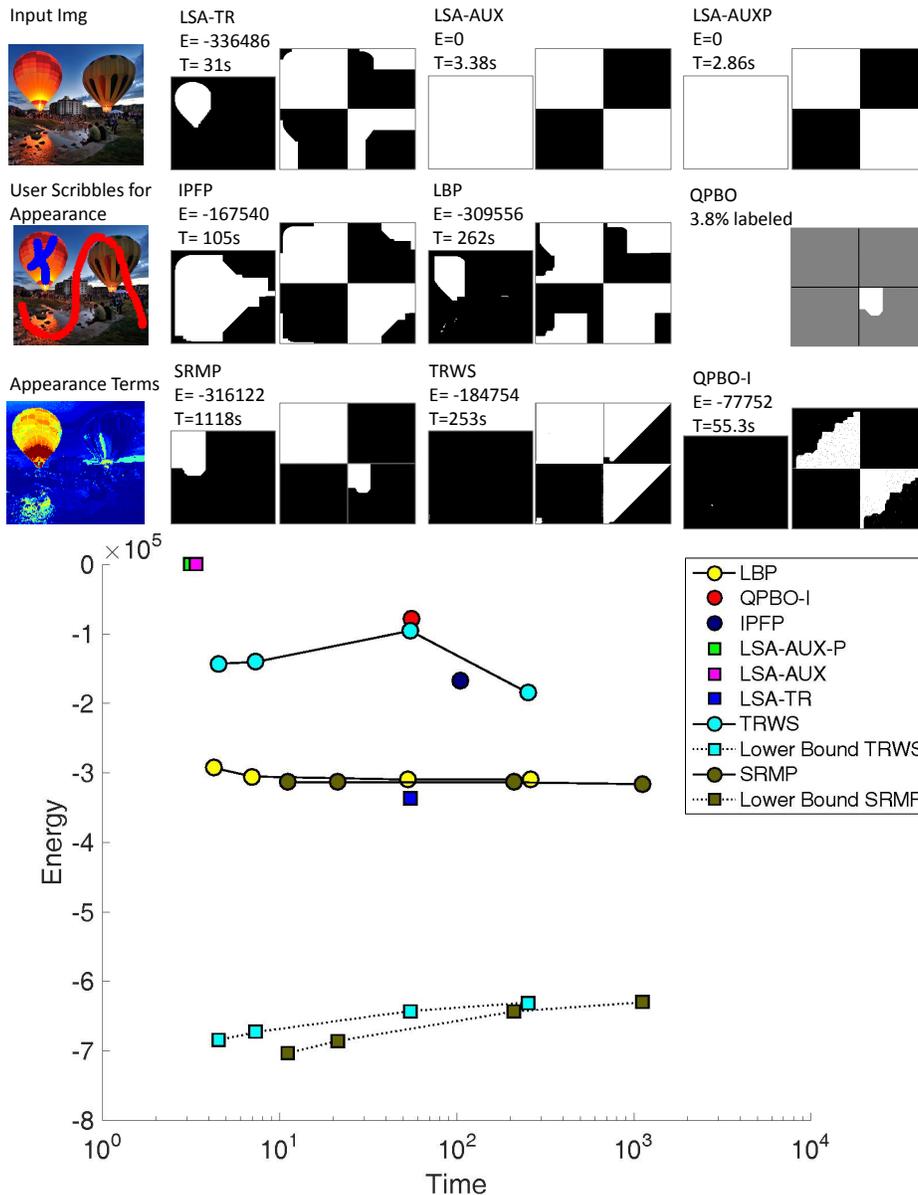
Fig. 11. Compact Shape Prior results and comparison with other methods. The left column shows input image, user scribbles and resulting appearance terms. Red colors show preference to foreground and blue colors show preference to background. The remaining columns show for each method the final image segmentation along with the corresponding labeling on each of the four layers in the graph. Gray color in QPBO results denotes unlabeled pixels. In the bottom we compare the methods in terms of energy and the running time (shown in log-scale).

We also plan to research additional applications that can benefit from efficient optimization of binary non-submodular pairwise energies. For instance, our experiments show that our approach can improve non-submodular α-expansion and fusion moves for multilabel energies.

Moreover, while our paper focuses on pairwise interactions, our approach naturally extends to high-order potentials that appear in computer vision problems. We already successfully applied LSA to optimization of convexity shape prior [35]. We further plan to explore other high-order energies such as visibility and silhouette consistency in multi-view reconstruction, connectivity shape prior and absolute curvature regularization.

6 ACKNOWLEDGEMENTS

REFERENCES

[1] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, p. 2002, 2001. 1, 10
[2] R. Lazimy, "Mixed integer quadratic programming," *Mathematical Programming*, vol. 22, pp. 332–349, 1982. 1
[3] M. Goemans and D. Wiliamson, "Improved approximation algorithms for maximum cut and satisfiability problem using semidefinite problem," *ACM*, vol. 42, no. 6, pp. 1115–1145, 1995. 1
[4] C. Olsson, A. Eriksson, and F. Kahl, "Improved spectral relaxation methods for binary quadratic optimization problems," *Comp. Vis. & Image Underst.*, vol. 112, no. 1, pp. 3–13, 2008. 1, 2

	Curr Config → New Config	(0,0)	(0,1)	(1,0)	(1,1)
LSA-TR	(0,0)	✓	✓	✓	✗
	(0,1)	✓	✓	✗	✓
	(1,0)	✓	✗	✓	✓
	(1,1)	✗	✓	✓	✓
LSA-TR-EXP 0-expansion	(0,0)	✓	✓	✓	✗
	(0,1)	⊘	✓	⊘	✓
	(1,0)	⊘	⊘	✓	✓
	(1,1)	⊘	⊘	⊘	✓
LSA-TR-EXP 1-expansion	(0,0)	✓	⊘	⊘	⊘
	(0,1)	✓	✓	⊘	⊘
	(1,0)	✓	⊘	✓	⊘
	(1,1)	✗	✓	✓	✓
	Approximation for current config	0	αx	αy	$\alpha x + \alpha y - \alpha$

✓ - exact approx.   ✗ - not accurate approx.   ⊘ - invalid move

Fig. 12. Approximation of supermodular term $\alpha \cdot xy$ in LSA-TR and LSA-TR-EXP. Columns and rows correspond to current and new configuration of (x,y) respectively. Gray section shows that twelve out of 16 cases yield exact approximation for LSA-TR. Pink and blue sections show that eight out of nine valid cases yield exact approximation for LSA-TR-EXP due to restricted search space. Green section specifies the approximation used for each current configuration.

Fig. 13. Optimization of squared curvature model. Top-left: input image with one foreground object. Top-right: Comparison of optimization energies as a function of increasing curvature weight λ_{curv}. LSA-TR (blue line) optimizes the energy starting from the maximum likelihood solution based on appearance terms. We used normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $(\mu = 0, \sigma = 0.2)$ for the foreground and $(\mu = 1, \sigma = 0.2)$ for the background respectively. We used 7×7 neighborhood for the angular resolution. LSA-TR-EXP-improve (green line) attempts to improve the final solution of LSA-TR using expansion moves. LSA-TR-EXP (red line) performs expansion moves starting from the same initial solution as LSA-TR. The energy of the initial solution is shown in black. In the bottom of the figure we show final solution of the three methods for three different values of λ_{curv}. Blue, red and green frames correspond to results of LSA-TR, LSA-TR-EXP and LSA-TR-EXP-improve respectively.

[5] J. Keuchel, C. Schnörr, C. Schellewald, and D. Cremers, "Binary partitioning, perceptual grouping, and restoration with semidefinite programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1364–1379, 2003. 1

[6] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008. 1

[7] 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, pp. 1–8. 1, 5

[8] V. Kolmogorov and T. Schoenemann, "Generalized seq. tree-reweighted message passing," *arXiv:1205.6352*, 2012. 1, 5

[9] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnorr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, J. Lellmann, N. Komodakis, and C. Rother, "A comparative study of modern inference techniques for discrete energy minimization problem," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1328–1335. 1, 7

[10] M. Leordeanu, M. Hebert, and R. Sukthankar, "An integer projected fixed point method for graph matching and map inference." in *Neural Information Processing Systems (NIPS)*, 2009, pp. 1114–1122. 1, 2, 5

[11] W. Brendel and S. Todorovic, "Segmentation as maximum-weight independent set," in *Neural Information Processing Systems (NIPS)*, 2010, pp. 307–315. 1

[12] L. Gorelick, F. R. Schmidt, and Y. Boykov, "Fast trust region for segmentation," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, Oregon, June 2013, pp. 1714–1721. [Online]. Available: http://www.csd.uwo.ca/~yuri/Abstracts/cvpr13-ftr-abs.shtml 2, 3

[13] M. Narasimhan and J. A. Bilmes, "A submodular-supermodular procedure with applications to discriminative structure learning," in *UAI*, 2005, pp. 404–412. 2, 4

[14] C. Rother, V. Kolmogorov, T. Minka, and A. Blake, "Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs," in *Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 993 – 1000. 2

[15] I. Ben Ayed, L. Gorelick, and Y. Boykov, "Auxiliary cuts for general classes of higher order functionals," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, Oregon, June 2013, pp. 1304–1311. [Online]. Available: http://www.csd.uwo.ca/~yuri/Abstracts/cvpr13-auxcut-abs.shtml 2

[16] R. Fletcher, *Practical Meth. of Opt.* Wiley & Sons, 1987. 2

[17] Y. Yuan, "A review of trust region algorithms for optimization," in *Proceedings of the Fourth International Congress on Industrial & Applied Mathematics (ICIAM)*, 1999. 2, 3

[18] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 1–20, 2000. 2

[19] M. P. Kumar, O. Veksler, and P. H. Torr, "Improved moves for truncated convex models," *J. Mach. Learn. Res.*, vol. 12, pp. 31–67, 2011. 2

[20] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake, "Digital tapestry," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, January 2005. 2

[21] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Graph cut based inference with co-occurrence statistics," in *Proceedings of the 11th European Conference on Computer Vision: Part V*, ser. ECCV'10, 2010, pp. 239–253. 2

[22] T. Taniai, Y. Matsushita, and T. Naemura, "Superdifferential cuts for
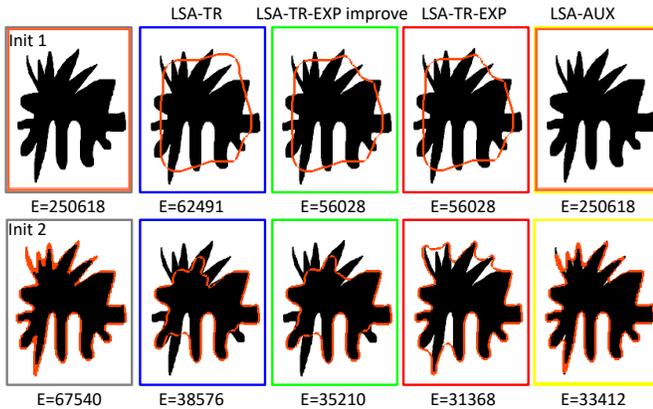
Fig. 14. Local optimization of squared curvature might yield different segmentation results for different initializations. First row - starting with all pixels assigned to foreground, second row - starting with appearance based ML labeling. Here we used $\lambda_{\text{curv}} = 1000$.

binary energies," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 2030–2038. 2

[23] M. Tang, I. B. Ayed, and Y. Boykov, "Pseudo-bound optimization for binary energies," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, 2014, pp. 691–707. 2

[24] Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong, "An integral solution to surface evolution PDEs via Geo-Cuts," in *European Conf. on Comp. Vision (ECCV)*, 2006. 3

[25] J. Ulen, P. Strandmark, and F. Kahl, "An efficient optimization framework for multi-region segmentation based on Lagrangian duality," *IEEE Transactions on Medical Imaging*, vol. 32, no. 2, pp. 178–188, 2013. 3

[26] Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong, "An Integral Solution to Surface Evolution PDEs via Geo-Cuts," *ECCV, LNCS 3953*, vol. 3, pp. 409–422, May 2006. 3

[27] J. Pearl, "Reverend bayes on inference engines: A distributed hierarchical approach," in *National Conference on Artificial Intelligence*, 1982, pp. 133–136. 5

[28] N. El-Zehiry and L. Grady, "Fast global optimization of curvature," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, no. 3257-3264, 2010. 6, 7, 8

[29] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," in *Machine Learning*, vol. 56, no. 1-3, 2004, pp. 89–113. 6

[30] J. H. Kappes, M. Speth, G. Reinelt, and C. Schnörr, "Higher-order segmentation via multicuts," *Comput. Vis. Image Underst.*, vol. 143, no. C, pp. 104–119, Feb. 2016. 6

[31] 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)*, no. 105-112, 2001. 6

[32] A. Delong and Y. Boykov, "Globally optimal segmentation of multi-region objects," in *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*, 2009, pp. 285–292. 7

[33] P. Das, O. Veksler, V. Zavadsky, and Y. Boykov, "Semiautomatic segmentation with compact shape prior," *Image Vision Computing*, vol. 27, no. 1-2, pp. 206–219, 2009. 9, 11

[34] D. Schlesinger, "Exact solution of permuted submodular minsum problems," in *Energy Minimization Methods in Computer Vision and Pattern Recognition, 6th International Conference, EMMCVPR 2007, Ezhou, China, August 27-29, 2007, Proceedings*, 2007, pp. 28–38. 10

[35] L. Gorelick, O. Veksler, Y. Boykov, and C. Nieuwenhuis, "Convexity shape prior for segmentation," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, 2014, pp. 675–690. 10, 12

[36] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001. 11

[37] C. Nieuwenhuis, E. Töppe, L. Gorelick, O. Veksler, and Y. Boykov, "Efficient squared curvature," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 4098–4105. 11

Lena Gorelick received the BSc degree cum laude in computer science from Bar-Ilan University in 2001, the MSc degree summa cum laude in computer science and applied mathematics from the Weizmann Institute of Science in 2004 and PhD degree in computer science and applied mathematics from the Weizmann Institute of Science in 2009. From 2009 to 2014 she was a postdoctoral fellow at computer science department of the University of Western Ontario and since 2014 she is a research scientist there. Her current research interests lie in computer vision, specifically in the area of shape analysis, image segmentation and discrete energy minimization methods.

Olga Veksler received BS degree in mathematics and computer science from New York University in 1995 and a PhD degree from Cornell University in 1999. She was a postdoctoral associate at NEC Research Institute. She is currently a full professor with Computer Science Department University of Western Ontario. Her research interests are energy minimization methods, graph algorithms, stereo correspondence, motion, and segmentation. She is a receiver of the early researcher award from Ontario Ministry of Research and Innovation, NSERC-DAS award, and Helmholtz Prize (Test of Time) awarded at the International Conference on Computer Vision, 2011.

Yuri Boykov received "Diploma of Higher Education" with honors at Moscow Institute of Physics and Technology in 1992 and completed his Ph.D. at the department of Operations Research at Cornell University in 1996. He is currently a full professor at the department of Computer Science at the University of Western Ontario. His research is concentrated in the area of computer vision and biomedical image analysis. In particular, he is interested in problems of early vision, image segmentation, restoration, registration, stereo, motion, model fitting, feature-based object recognition, photo-video editing and others. He is a recipient of the Helmholtz Prize (Test of Time) awarded at International Conference on Computer Vision (ICCV), 2011 and Florence Bucke Science Award, Faculty of Science, The University of Western Ontario, 2008.

Andrew Delong received the B.Math. degree in computer science from the University of Waterloo in 2003 and the M.Sc. and Ph.D. degree in computer science from Western University in 2006 and 2011, respectively. He is a Postdoctoral Fellow at the University of Toronto. Prior to entering academia he worked in the computer graphics industry. His research interests include machine learning, computer vision, combinatorial optimization, and computational biology. Dr. Delong was awarded a NSERC Postgraduate Scholarship, an NSERC Postdoctoral Fellowship, a Heffernan Commercialization Fellowship, and a 2016 Invention of the Year Award from the University of Toronto.

Ismail Ben Ayed received the PhD degree (with the highest honor) in computer vision from the INRS-EMT, Montreal in 2007. He is currently an Associate Professor at the TS, University of Quebec. Before joining the TS, he worked for 8 years as a research scientist at GE Healthcare, London, ON. He also holds an adjunct professor appointment at Western University (since 2012). Ismails research interests include computer vision, optimization, machine learning and their potential applications in medical image analysis. He co-authored a book, over seventy peer-reviewed publications, mostly published in the top venues in these subject areas, and six US patents.