

Convexity Shape Prior for Binary Segmentation

Lena Gorelick, Olga Veksler, Yuri Boykov, and Claudia Nieuwenhuis

Abstract—Convexity is a known important cue in human vision. We propose shape convexity as a new high-order regularization constraint for binary image segmentation. In the context of discrete optimization, object convexity is represented as a sum of 3-clique potentials penalizing any 1-0-1 configuration on all straight lines. We show that these non-submodular potentials can be efficiently optimized using an iterative trust region approach. At each iteration the energy is linearly approximated and globally optimized within a small trust region around the current solution. While the quadratic number of all 3-cliques is prohibitively high, we design a dynamic programming technique for evaluating and approximating these cliques in linear time. We also derive a second order approximation model that is more accurate but computationally intensive.

We discuss limitations of our local optimization and propose gradual non-submodularization scheme that alleviates some limitations. Our experiments demonstrate general usefulness of the proposed convexity shape prior on synthetic and real image segmentation examples. Unlike standard second-order length regularization, our convexity prior does not have shrinking bias, and is robust to changes in scale and parameter selection.

Index Terms—Segmentation, convexity shape prior, high-order functionals, trust region, graph cuts.

I. INTRODUCTION

Length-based regularization is commonly used for ill-posed segmentation problems, in part because efficient global optimization algorithms are well-known for both discrete and continuous formulations, e.g. [3], [23]. Nevertheless, the shrinking bias and the sensitivity to the weight of the length term in the energy are widely recognized as limitations of this form of regularization. These problems motivate active research on optimization of higher-order regularization energies, e.g. curvature [24], [6], [22], [20] and cooperative prior [12], [13], which can alleviate the shrinking bias and other issues.

We propose a new higher-order regularization model: convexity shape constraint, see Fig.1. Convexity was identified as an important cue in human vision [18], [19]. Many natural images have convex or nearly convex objects. Convex objects are also common in medical images. Yet, to the best of our knowledge, we are the first to introduce a convexity shape prior into discrete segmentation energy.

We develop an energy-based formulation for convexity prior in discrete optimization framework and propose an efficient optimization algorithm for the corresponding non-submodular high-order energy term. For a given segment $S \subset \Omega$, the overall segmentation energy $E(S)$ can combine our convexity prior $E_{convexity}(S)$ with user-defined hard-constraints, linear

appearance models[4], boundary length [3], color separation [26], or any others standard submodular terms $E_{sub}(S)$

$$E(S) = E_{convexity}(S) + E_{sub}(S). \quad (1)$$

Convexity of segment S is expressed as a penalty for all ordered triplet configurations 1-0-1 along any straight line, see Fig.2. Similar straight 3-cliques also appear in curvature modeling [20], but they also need 0-1-0 configurations to penalize negative curvature. Moreover, they use only local triplets to evaluate curvature. In contrast, convexity is **not** a local property of the segment boundary. Therefore, we have to penalize 1-0-1 configurations on straight intervals of any length. Consequently, our convexity energy model has a much larger number of cliques. We propose an efficient dynamic programming technique to evaluate and approximate these cliques in the context of trust region optimization [10].

Related Work: Many related shape priors were introduced in the past. Common length-based regularizer [3] penalizes segment perimeter favoring smooth solutions that are closer to circles and, therefore, more convex. However, as shown in our experiments, this prior needs to be carefully balanced with the appearance term as it has a strong shrinking bias. Connectivity regularizer [28], [21] does not have shrinking bias but might suffer from connecting thin structure artifacts.

Another related regularizer is the *star shape prior* [27], [11], which imposes convexity constraints only along the lines passing through a reference point given by the user: these lines are allowed to enter and exit the object only once. In contrast to our convexity prior, the star shape prior allows for non-convex objects, e.g. a star.

There are also part-based shape priors [29], [17], [7]. A shape is partitioned into several parts and each part imposes certain constraints on the direction of the boundary with the background. This approach can model some simple convex shapes, e.g. a rectangle, but it can not represent a general convexity prior.

The most related work to ours is [25], which models the object as an n -sided convex polygon. It is a part-based approach that uses one foreground and n background labels. For an accurate segmentation of an arbitrary convex object, e.g. a circle, a finer discretization (i.e. more background parts) is required, significantly increasing runtime. The larger the object, the worse is the problem. In contrast, we can obtain an arbitrary convex object for any choice of orientation discretization. Moreover, [25] relies on continuous optimization and is not efficient without GPU. Additional related work on optimization is discussed in Sec. IV-C.

Contributions: We introduce a new discrete convexity shape regularizer. When enforced as a hard constraint, it is a parameter-free convexity shape prior. In practice, we enforce convexity as a soft constraint using a finite penalty

L. Gorelick, O. Veksler and Y. Boykov are with the Department of Computer Science, University of Western Ontario, London, ON, N6A 5B7, Canada. e-mail: olga@csd.uwo.ca.

C. Nieuwenhuis is with International Computer Science Institute, UC Berkeley, California, USA.

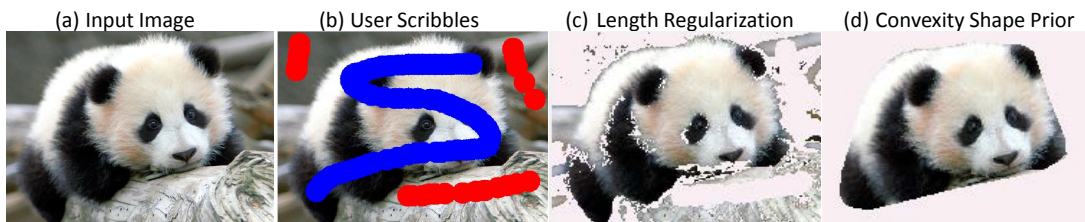


Fig. 1. Segmentation with convexity shape prior: (a) input image, (b) user scribbles, (c) segmentation with contrast sensitive length regularization. We optimized the weight of length with respect to ground truth. (d) segmentation with convexity shape prior.

ω . However, our experiments show that there is almost no variation in segmentation results for different values of the convexity weight ω , once the value is high enough. Our method is also robust to changes in scale, see Sec. IV.

We develop an optimization algorithm based on trust region framework and show how to use dynamic programming to significantly improve efficiency. In the context of trust region, we derive and compare both linear and quadratic approximations of our energy. While quadratic approximation is more accurate, it is computationally more expensive. Nonetheless, it could potentially be useful for approximating other high-order energies.

We discuss alternative convexity models such as *central clique model* and *local convexity model* and alternative optimization schemes such as *gradual non-submodularization*. We also experimentally validate the advantage of our convexity vs. the length regularizer for segmentation.

Finally, we collected a new dataset of natural images with ground truth convex segmentations, available for download.

The paper is organized as follows. Sec. II formulates convexity energy and Sec. II-A explains its efficient evaluation. Sec. III introduces trust region optimization framework. We derive linear approximation of our energy in Sec. III-A and show how to compute it efficiently using dynamic programming in Sec. III-B. In Sec. III-C we discuss a more accurate quadratic approximation of our energy. Sec. IV-A demonstrates the usefulness of convexity shape prior and compares it to the standard length regularization. In Sec. IV-B we discuss alternative *central* and *local* convexity models and in Sec. IV-C we discuss alternative optimization schemes. Finally, in Sec. IV-D we point out some limitations of our local optimization. We evaluate two possible solutions: gradual non-submodularization in Sec. IV-D2 and quadratic approximation in Sec. IV-D3 to alleviate some of the limitations.

II. ENERGY

Denote by Ω the set of all image pixels and let $S \subset \Omega$ be a segment. Let \mathbf{x} be a vector of binary indicator variables $x_s \in \{0, 1\}$, $p \in \Omega$ such that $S = \{p \mid x_s = 1\}$. Due to one-to-one correspondence, we will use either \mathbf{x} or S interchangeably.

In this paper we focus on convexity shape prior and propose a novel formulation to incorporate this prior into segmentation energy. In continuous case, segment S is convex if and only if for any $p, r \in S$ there is no q on a line between them s.t. $q \notin S$. In discrete case, we approximate convexity constraints as follows.

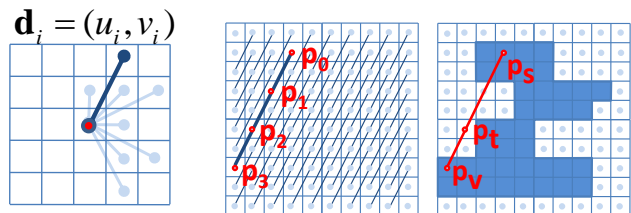


Fig. 2. Left: Example of discretized orientations given by a 5×5 stencil. One orientation d_i is highlighted. Middle: Set L_i of all discrete lines on image grid that are parallel to d_i . Right: Example of a triple clique (p_s, p_t, p_v) that violates convexity constraint.

Let $i \in \{1, \dots, m\}$ enumerate discrete line orientations, see Fig. 2 (left). Let $p \in \Omega$ be a pixel, and let l_p^i be a discrete line passing through p in orientation d_i . That is,

$$l_p^i = \{p_t \mid p_t = p + t \cdot d_i, t \in \mathcal{Z}, p_t \in \Omega\}. \quad (2)$$

We define $L_i = \{l_p^i \mid p \in \Omega\}$ as the set of all *discrete lines* l_p^i of given orientation d_i . Fig. 2 (middle) illustrates set L_i for one particular orientation d_i and highlights one l_p^i . To avoid double indexing throughout the paper we use x_t instead of x_{p_t} to denote the binary variable for pixel p_t on a line¹.

One way to represent discrete convexity constraint is based on potential $\phi : \{0, 1\}^3 \rightarrow \mathcal{R}$ defined for all triplets of *ordered* pixels (p_s, p_t, p_v) , $s < t < v$ along any discrete line $l \in \bigcup L_i$

$$\phi(x_s, x_t, x_v) = \begin{cases} \infty & \text{if } (x_s, x_t, x_v) = (1, 0, 1) \\ 0 & \text{otherwise.} \end{cases}$$

In practice we use some finite penalty ω redefining potential ϕ algebraically as

$$\phi(x_s, x_t, x_v) = \omega \cdot x_s(1 - x_t)x_v. \quad (3)$$

The convexity energy $E_{convexity}(\mathbf{x})$ integrates this triple clique potential over all orientations, all lines and all triplets:

$$E_{convexity}(\mathbf{x}) = \sum_{l \in \bigcup L_i} \sum_{\substack{(p_s, p_t, p_v) \in l \\ s < t < v}} \phi(x_s, x_t, x_v). \quad (4)$$

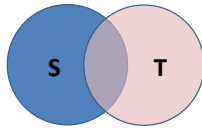
As discussed below, 3^{rd} -order energy (4) is hard to optimize for two reasons: it is non-submodular and it has a prohibitively large number of cliques.

It is easy to verify that this energy is non-submodular [8]. It is enough to show that there exist segments $S, T \subset \Omega$ s.t.

$$E(S) + E(T) < E(S \cap T) + E(S \cup T) \quad (5)$$

¹Note, pixel $p \in \Omega$ has unique index t_l on each line l passing through it.

Consider the example below. Since both S and T are convex, the left hand side in (5) is zero, while the right hand side is infinite since the union of S and T is not convex. Therefore, our energy cannot be optimized with standard methods for sub-modular functions.



At the first glance, it seems prohibitively expensive to even evaluate our energy on reasonably sized images. For example, for a 200×300 image, with just 8 orientations, there are roughly 32 billion triple cliques. In Sec. II-A, III-B we show how to evaluate and approximate the $E_{convexity}$ in time linear w.r.t. image size using dynamic programming. Then, in Sec. III we show how to optimize our energy using trust region techniques [30], [10]. Alternative convexity models and optimization schemes are discussed in Sec. IV-B and IV-C.

A. Energy Evaluation via Dynamic Programming

This section explains how to evaluate our convexity term $E_{convexity}(\mathbf{x})$ efficiently. We show how to compute the inner summation in (4) for one given line l . The idea is to use dynamic programming to efficiently count the number of triplets $(1, 0, 1)$ on a line violating convexity constraints.

Let \mathbf{x}_l denote a vector of binary indicator variables on line l . We rewrite

$$\begin{aligned} E_{convexity}(\mathbf{x}_l) &= \sum_{\substack{(p_s, p_t, p_v) \in l \\ s < t < v}} \phi(x_s, x_t, x_v) \\ &= \omega \cdot \sum_{s < t < v} x_s \cdot (1 - x_t) \cdot x_v. \end{aligned}$$

Consider pixels $p_s, p_t, p_v \in l$. We say pixel p_s precedes pixel p_t on line l if $s < t$. Similarly, pixel p_v succeeds pixel p_t if $v > t$. Let $C^-(t)$ be the number of pixels p_s preceding pixel p_t such that $x_s = 1$, and $C^+(t)$ be the number of pixels p_v succeeding pixel p_t such that $x_v = 1$:

$$C^-(t) = \sum_{s < t} x_s, \quad C^+(t) = \sum_{v > t} x_v. \quad (6)$$

To count the number of all violating configuration $(1, 0, 1)$ for ordered triplets on line l we first consider one fixed pixel $p_t \in l$ with zero label $x_t = 0$. Each preceding pixel with label one and each succeeding pixel with label one form configuration $(1, 0, 1)$. Thus, the total combinatorial number of ordered triplets (p_s, p_t, p_v) , $s < t < v$, with configuration $(1, 0, 1)$ is given by $C^+(t) \cdot C^-(t)$, see Fig. 3. Summing over all zero label pixels on line l gives

$$E_{convexity}(\mathbf{x}_l) = \omega \cdot \sum_t C^+(t) \cdot C^-(t) \cdot (1 - x_t).$$

Note that $C^-(t) = C^-(t-1) + x_{t-1}$ and $C^+(t) = C^+(t+1) + x_{t+1}$. Hence each of $C^+(t)$ and $C^-(t)$ can be computed for all pixels on a line in one pass using running sums. For a particular orientation d_i , each pixel appears in one line only. Therefore, the total number of operations needed to compute $E_{convexity}(\mathbf{x})$ is $O(mN)$, where $N = |\Omega|$ is the number of pixels in the image and m is the number of distinct orientations.

x_l	1	1	0	1	0	1	1	0
C^-	0	1	2	2	3	3	4	5
C^+	4	3	3	2	2	1	0	0
			6		6			0

Fig. 3. Evaluation of $E_{convexity}$. The top row shows current configuration \mathbf{x}_l of pixels on line l . The second and the third rows show the number of pixels p_s with $x_s = 1$ before and after each pixel p_t , that is, functions $C^-(t)$ and $C^+(t)$. The last row shows the number of violated constraints for each p_t with $x_t = 0$, resulting in total of 12 violations on the line.

III. OPTIMIZATION

This section describes our optimization algorithm for segmentation energy (1) with the convexity shape prior. In terms of indicator variables \mathbf{x} this energy is

$$E(\mathbf{x}) = E_{convexity}(\mathbf{x}) + E_{sub}(\mathbf{x}) \quad (7)$$

where E_{sub} is any submodular term² that can be optimized with graph cuts, e.g. boundary length [3], color separation [26], etc. As mentioned earlier, our term $E_{convexity}$ is non-submodular and therefore hard to optimize. An additional difficulty is the large number of triple cliques in $E_{convexity}(\mathbf{x})$.

For optimization, we use iterative trust region (TR) framework [30], which has been shown promising for various non-submodular energies [10], [20], [9]. In each iteration, we construct an approximate tractable model \tilde{E}^k of the energy E in (7) near current solution \mathbf{x}^k . The model is only accurate within a small region around \mathbf{x}^k called “trust region”. The approximate \tilde{E}^k is then 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. See [30] for a review of trust region.

Algorithm 1 summarizes our approach. Line 4 computes unary approximate energy E_{approx}^k for the non-submodular $E_{convexity}$ around \mathbf{x}^k . Line 5 combines E_{approx}^k with the submodular E_{sub} . The resulting \tilde{E}^k is submodular and coincides with the exact energy E on \mathbf{x}^k . The TR sub-problem requires minimization of \tilde{E}^k within a small region $\|\mathbf{x} - \mathbf{x}^k\| \leq d_k$ around \mathbf{x}^k . Unfortunately, minimizing \tilde{E}^k under distance constraints is NP-hard [9]. Instead, we use a simpler formulation of the TR sub-problem proposed in [9], [10] based on unconstrained optimization of submodular Lagrangian

$$L^k(\mathbf{x}) = \tilde{E}^k(\mathbf{x}) + \lambda_k \|\mathbf{x} - \mathbf{x}^k\|. \quad (8)$$

Here parameter λ_k controls the trust region size indirectly instead of distance d_k . We use L_2 distance expressed with unary terms, see [10], [9]. Therefore $L^k(\mathbf{x})$ is submodular. Line 7 solves (8) for some fixed λ_k using one graph-cut.

The candidate solution \mathbf{x}^* is accepted whenever the original energy decreases (line 25). The Lagrange multiplier λ_k is

²The submodularity of the last term in (7) is assumed for clarity. The proposed trust region approach can approximate non-submodular terms jointly with $E_{convexity}$.

Algorithm 1: TRUST REGION CONVEXITY

```

1  $\mathbf{x}^0 \leftarrow \mathbf{x}_{\text{init}}, \lambda_0 \leftarrow \lambda_{\text{init}}, \text{convergedFlag} \leftarrow 0$ 
2 Repeat Until convergedFlag
3   //Approximate  $E_{\text{convexity}}(\mathbf{x})$  around  $\mathbf{x}^k$ 
4   Compute  $E_{\text{approx}}^k(\mathbf{x})$  (see Sec. III-A)
5    $\tilde{E}_k(\mathbf{x}) = E_{\text{approx}}^k(\mathbf{x}) + E_{\text{sub}}(\mathbf{x})$  // keep the submodular part
6   //Trust region sub-problem
7    $\mathbf{x}^* \leftarrow \text{argmin}_{\mathbf{x}} L^k(\mathbf{x})$  (8)
8   Evaluate  $E_{\text{convexity}}(\mathbf{x}^k), E_{\text{convexity}}(\mathbf{x}^*)$  (see Sec. II-A)
9   Evaluate  $E_{\text{approx}}^k(\mathbf{x}), E_{\text{approx}}^k(\mathbf{x}^*)$  (see Sec. III-B)
10   $R = E(\mathbf{x}^k) - E(\mathbf{x}^*)$  // actual reduction in energy
11   $P = \tilde{E}^k(\mathbf{x}^k) - \tilde{E}^k(\mathbf{x}^*)$  // predicted reduction in energy
12  If  $P = 0$  // meaning  $\mathbf{x}^* = \mathbf{x}^k$  and  $\lambda > \lambda_{\text{max}}$ 
13     $\lambda_k \leftarrow \lambda_{\text{max}}$ 
14    //Trust region sub-problem:
15     $\mathbf{x}^* \leftarrow \text{argmin}_{\mathbf{x}} L^k(\mathbf{x})$  (8)
16    Evaluate  $E_{\text{convexity}}(\mathbf{x}^k), E_{\text{convexity}}(\mathbf{x}^*)$  (see Sec. II-A)
17    Evaluate  $E_{\text{approx}}^k(\mathbf{x}^k), E_{\text{approx}}^k(\mathbf{x}^*)$  (see Sec. III-B)
18     $R = E(\mathbf{x}^k) - E(\mathbf{x}^*)$  // actual reduction in energy
19     $P = \tilde{E}^k(\mathbf{x}^k) - \tilde{E}^k(\mathbf{x}^*)$  // predicted reduction in energy
20    //Update current solution
21     $\mathbf{x}^{k+1} \leftarrow \begin{cases} \mathbf{x}^* & \text{if } R > 0 \\ \mathbf{x}^k & \text{otherwise} \end{cases}$ 
22    convergedFlag  $\leftarrow (R \leq 0)$ 
23  Else // meaning  $\mathbf{x}^* \neq \mathbf{x}^k$  and  $\lambda \leq \lambda_{\text{max}}$ 
24    //Update current solution
25     $\mathbf{x}^{k+1} \leftarrow \begin{cases} \mathbf{x}^* & \text{if } R > 0 \\ \mathbf{x}^k & \text{otherwise} \end{cases}$ 
26    //Adjust the trust region
27     $\lambda_{k+1} \leftarrow \begin{cases} \lambda_k / \alpha & \text{if } R/P > \tau_2 \\ \lambda_k \cdot \alpha & \text{otherwise} \end{cases}$ 
28 we use  $\alpha = 10, \tau_2 = 0.25$ 

```

adaptively changed (line 27), based on the quality of the current approximation as motivated by empirical inverse proportionality relation between λ_k and d_k (see [10]). 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 $\mathbf{x}^* = \mathbf{x}^k$ (Line 12), one more attempt is made using λ_{max} , where $\lambda_{\text{max}} = \sup \{\lambda | \mathbf{x}^* \neq \mathbf{x}^k\}$ (see [10]). If there is no reduction in energy with smallest discrete step λ_{max} (Line 21), we are at a local minimum [5] and we stop (Line 22).

A. Linear Approximation of $E_{\text{convexity}}$

Below we derive linear approximation $E_{\text{approx}}^k(\mathbf{x})$ for the energy term $E_{\text{convexity}}(\mathbf{x})$ in (4) around current solution \mathbf{x}^k

$$E_{\text{approx}}^k(\mathbf{x}) = \sum_{l \in \bigcup L_i} \sum_{\substack{(p_s, p_t, p_v) \in l \\ s < t < v}} \phi^k(x_s, x_t, x_v) \quad (9)$$

where $\phi^k(x_s, x_t, x_v)$ is a linear approximation of the corresponding $\phi(x_s, x_t, x_v)$ in (3) around \mathbf{x}^k , as explained below.

Property III-A.1. For any potential $\phi(\mathbf{x}) : \{0, 1\}^n \rightarrow \mathcal{R}$ of n binary variables $\mathbf{x} = (x_1, \dots, x_n)$ and any subset $A \subset \{0, 1\}^n$ of $n+1$ distinct binary configurations of \mathbf{x} , there is a linear function $L_A(\mathbf{x}) = a_0 + a_1 x_1 + a_2 x_2 + \dots + a_n x_n$ such that $\phi(\mathbf{x}) = L_A(\mathbf{x})$ for any $\mathbf{x} \in A$.

The proof of Prop. III-A.1 is by construction. To find linear approximation $L_A(\mathbf{x})$ of any potential $\phi(\mathbf{x})$, we need to compute $n+1$ unknown coefficients of $L_A(\mathbf{x})$. This can be done by solving a system of $n+1$ equations $\phi(\mathbf{x}) = L_A(\mathbf{x})$ for $n+1$ binary configurations in some chosen A . Note that there are 2^n possible configurations of n binary variables. Therefore, for our triple clique potential in (3), set A contains four out of eight distinct configurations of labels (x_s, x_t, x_v) . Since in our energy we have a large number of overlapping triple cliques, we apply property III-A to simultaneously approximate the energy on all triple cliques and sum up the resulting approximations.

In practice, for each triple clique, we use an approach that avoids solving systems of equations and implicitly selects a specific set A . Note, any discrete potential ϕ can be written as a combination of *multilinear* functions of variables (x_1, \dots, x_n) , see (3). In this case, it is easy to verify that Taylor expansion ϕ^k of the potential ϕ around configuration \mathbf{x}^k is a linear function satisfying Prop. III-A.1. That is, $\phi^k(\mathbf{x})$ agrees with $\phi(\mathbf{x})$ on configuration \mathbf{x}^k and n other ‘‘neighboring’’ configurations obtained by flipping one of the variables in $\mathbf{x}^k = (x_1^k, \dots, x_n^k)$.

In our case, omitting the constant terms, Taylor expansion of (3) around \mathbf{x}^k yields³:

$$\begin{aligned} \phi^k(x_s, x_t, x_v) &= (1 - x_t^k) \cdot x_v^k \cdot x_s & (10) \\ &- x_v^k \cdot x_s^k \cdot x_t \\ &+ (1 - x_t^k) \cdot x_s^k \cdot x_v. \end{aligned}$$

The components in (10) have an intuitive interpretation. Consider the first component $(1 - x_t^k) \cdot x_v^k \cdot x_s$. Recall that pixels p_s, p_t, p_v are on a line and p_t is between p_s and p_v . If the current configuration \mathbf{x}^k is such that $x_t^k = 0$, and $x_v^k = 1$, then assigning label 1 to pixel p_s violates convexity, assuming p_t and p_v keep their labels unchanged from \mathbf{x}^k . The unary term $(1 - x_t^k) \cdot x_v^k \cdot x_s$ penalizes this violation: assignment $x_s = 1$ carries a penalty, whereas $x_s = 0$ is not penalized. The other two components in (10) have similar intuitive interpretations.

Summing approximations of all triple potentials in (9) results in a linear function $E_{\text{approx}}^k(\mathbf{x})$ that coincides with the original $E_{\text{convexity}}(\mathbf{x})$ on current solution \mathbf{x}^k .

Approximation in (10) gives three unary terms for each triple clique. Consider pixel $p_s \in l$. It can be either the leftmost, middle, or rightmost member of a clique on that line. We need to sum the terms from all triple cliques on line l involving pixel p_s . First with p_s being on the left, then in the middle and finally on the right of the clique. All these terms contribute to the unary potential $u_s^l(x_s)$ for a single pixel p_s^4 :

$$\begin{aligned} u_s^l(x_s) &= \sum_{\substack{(p_t, p_v) \in l \\ s < t < v}} (1 - x_t^k) \cdot x_v^k \cdot x_s & (11) \\ &- \sum_{\substack{(p_t, p_v) \in l \\ t < s < v}} x_t^k \cdot x_v^k \cdot x_s \\ &+ \sum_{\substack{(p_t, p_v) \in l \\ t < v < s}} (1 - x_v^k) \cdot x_t^k \cdot x_s. \end{aligned}$$

³Here we assume $\omega = 1$. If $\omega \neq 1$, all the derived formulas should be multiplied by ω .

⁴For brevity we use $u_s^l(x_s)$ instead of $u_{p_s}^l(x_s)$.

The full Taylor based unary term for pixel $p \in \Omega$ sums the above expression over all lines passing through p .

Fig. 5 illustrates the resulting unary terms arising from such approximation. They encourage any holes or concavities in the foreground segment to be filled in, and any protrusions/disconnected components to be erased. Efficient computation of (11) is discussed in Section III-B.

There is a relation between the Taylor unary terms in (11) and parallel ICM algorithm, first noted in [16]. However, our trust region framework has many differences from parallel ICM [16]. See [9] for the experimental comparison.

B. Computation of E_{approx}^k via Dynamic Programming

Naive computation of the summations in (11) is too costly. We now explain how to compute the unary terms in (11) efficiently. Similarly to Sec. II-A, the speedup is achieved with running sums on each line.

Let s, t, v enumerate the pixels on line l . In (11), the first sum counts the number of pixel pairs (p_t, p_v) such that $s < t < v$ and $x_t^k = 0, x_v^k = 1$. The second sum counts the number of pixels pairs (p_t, p_v) such that $t < s < v$ and $x_t^k = x_v^k = 1$. The last sum counts the number of pixels pairs (p_t, p_v) such that $t < v < s$ and $x_t^k = 1, x_v^k = 0$.

Let C^- and C^+ be as in (6). Recall that each of them can be computed in one pass over the line. Then the second sum in (11) is simply $C^-(s) \cdot C^+(s)$. For the other two sums, we need additional running sums.

Denote by $A^-(s)$ as the number of pixel pairs (p_t, p_v) preceding pixel p_s such that $x_t^k = 1, x_v^k = 0$ and pixel p_t precedes pixel p_v ,

$$A^-(s) = \sum_{\substack{(p_t, p_v) \in l \\ t < v < s}} (1 - x_v^k) \cdot x_t^k.$$

Similarly, we define $A^+(s)$ the number of pixel pairs (p_t, p_v) succeeding pixel p_s such that $x_t^k = 0, x_v^k = 1$ and pixel p_t precedes pixel p_v ,

$$A^+(s) = \sum_{\substack{(p_t, p_v) \in l \\ s < t < v}} (1 - x_t^k) \cdot x_v^k.$$

Given C^- , we compute A^- in one pass over line l using $A^-(0) = A^-(1) = 0$ and recurrence $A^-(s) = A^-(s-1) + (1 - x_{s-1}^k) \cdot C^-(s-1)$.

A^+ is computed analogously to A^- , given C^+ . Then the first sum in (11) is $A^+(t)$ and the third sum is $A^-(t)$, and $u_s^l(x_s) = x_s \cdot [A^+(s) - C^-(s) \cdot C^+(s) + A^-(s)]$.

Computing A^- and A^+ is linear in the number of pixels on a line. For orientation d_i , each pixel appears on one line only. Therefore we can compute A^- and A^+ for all lines in $O(mN)$ time, where m is the number of orientations and $N = |\Omega|$. Then the unary term for each pixel is computed in $O(m)$ time. Thus the total time to compute Taylor based unary terms for all pixels is $O(mN)$.

C. Quadratic Approximation of $E_{convexity}$

Linear approximation derived in section III-A works well for most synthetic and natural images. However, we were able

to design synthetic examples (see Fig. 14 and Fig. 15) where linear approximation yields poor results for some initializations. In this section we derive a quadratic approximation that gives better results but at a larger computational cost.

As before, we derive approximation $E_{approx}^k(\mathbf{x})$ for $E_{convexity}(\mathbf{x})$ in (4) around current solution \mathbf{x}^k . It is sufficient to derive a quadratic approximation $\phi^k(x_s, x_t, x_v)$ for a clique $\phi(x_s, x_t, x_v)$ and then substitute into (9).

Similarly to linear approximation in Sec. III-A, we first try second-order Taylor expansion of (3) around \mathbf{x}^k . Omitting constants, we get:

$$\begin{aligned} \phi^k(x_s, x_t, x_v) &= x_s \cdot x_v & (12) \\ &- x_s^k \cdot x_t \cdot x_v - x_s \cdot x_t^k \cdot x_v - x_s \cdot x_t \cdot x_v^k \\ &+ x_s \cdot x_t^k \cdot x_v^k + x_s^k \cdot x_t \cdot x_v^k + x_s^k \cdot x_t^k \cdot x_v. \end{aligned}$$

The second and third lines in (12) contain submodular quadratic and unary terms. The first term in (12) is supermodular and therefore cannot be optimized directly using standard techniques. We tried to replace the non-submodular term with two different approximations: (i) $x_s \cdot x_v \approx x_s^k \cdot x_v + x_s \cdot x_v^k$, i.e. Taylor-based unary approximation; (ii) $x_s \cdot x_v \approx 0$, i.e. zero approximation. Both produce poor results (omitted) on the difficult example in Fig. 15.

We now derive an alternative quadratic approximation. Recall from Sec. III-A that linear approximation in (11) can be derived from parallel ICM [16]. First we give details of this derivation and later extend it to quadratic case.

In ICM, given current labeling \mathbf{x}^k , we fix the labels of all pixels except one, and optimize over the remaining variable. For example, fixing all labels except at pixel p , the clique potential in (3) reduces to a unary term $x_s \cdot (1 - x_t^k) \cdot x_v^k$. In parallel ICM this is done simultaneously for all pixels, thus explaining the name parallel. For a given triple clique (p_s, p_t, p_v) , summing up the reductions for pixels p, q and r , and omitting the constant term $x_s^k \cdot x_v^k$, gives exactly the same linear approximation as in Taylor based approximation derived in (10). This is a standard *single* pixel ICM, which we distinguish from a more general version described below.

The propose a new quadratic approximation based on parallel *pairwise* (two pixel) ICM. Similarly to single pixel ICM, given current labeling \mathbf{x}^k , we now fix the labels of all pixels except a pair of interacting pixels. For example, fixing the labels of all pixels except (x_s, x_t) , we get the following quadratic reduction of the clique potential in (3):

$$x_s \cdot x_v^k - x_s \cdot x_t \cdot x_v^k.$$

Repeating the same procedure for pixel pairs (q, r) and (p, r) and summing up all the reductions results in:

$$\begin{aligned} \phi^k(x_s, x_t, x_v) &= x_s \cdot x_v^k - x_s \cdot x_t \cdot x_v^k \\ &+ x_s \cdot x_v - x_s \cdot x_t^k \cdot x_v & (13) \\ &+ x_s^k \cdot x_v - x_s^k \cdot x_t \cdot x_v \end{aligned}$$

Note that the approximation in (13) contains a non-submodular term $x_s \cdot x_v$ and, therefore, cannot be efficiently optimized with standard techniques. As previously, we can replace the supermodular term with its linear approximation. We tried two different approximations: $x_s \cdot x_v \approx x_s^k \cdot x_v + x_s \cdot x_v^k$ and

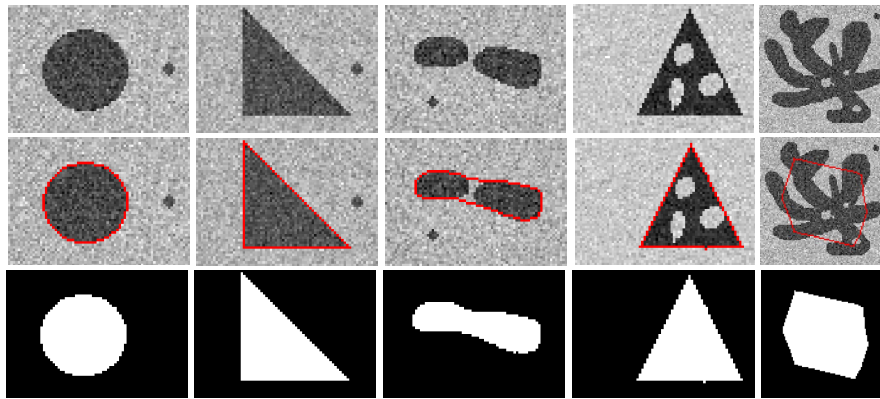


Fig. 4. First row shows synthetic images with added noise $\sigma_{noise} = 0.2$; Second and third rows show contours and masks of segmentation, $\omega = 0.1$. We used log-likelihood appearance terms, $(\mu_{fg} = 0, \sigma_{fg} = 0.1)$ and $(\mu_{bg} = 1, \sigma_{bg} = 0.1)$. The convexity prior removes noise, connects components and fills holes while preserving sharp corners.

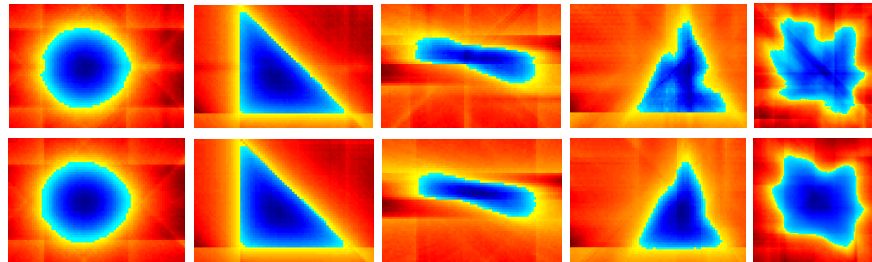


Fig. 5. First and second rows show unary approximation terms of $E_{convexity}$ as in (11) during the first and second iterations of trust region for the examples in Fig. 4. Red-yellow colors denote preference to background, and blue-cyan colors - preference to foreground. Unary terms encourage filling of holes/concavities and removal of protrusions/disconnected components.

$x_s \cdot x_v \approx 0$, and the second one works better in practice. Thus, our final quadratic approximation based on pairwise ICM is:

$$\begin{aligned} \phi^k(x_s, x_t, x_v) &= x_s \cdot x_v^k - x_s \cdot x_t \cdot x_v^k \\ &- x_s \cdot x_t^k \cdot x_v \\ &+ x_s^k \cdot x_v - x_s^k \cdot x_t \cdot x_v \end{aligned} \quad (14)$$

Observe that pairwise ICM approximation in (14) is different from the second order Taylor approximation in (12) even though the single pixel ICM approximation coincides with the first order Taylor approximation.

Pairwise ICM approximation is more accurate than linear approximation. It coincides with the original triple clique potential $\phi(\mathbf{x})$ on at least five out of eight configurations. More precisely depending on the current configuration \mathbf{x}^k the approximation coincides on either five, six or seven configurations. This is in contrast to at least four out of eight configurations for the linear approximation. Pairwise ICM approximation (14) works better than linear approximation, but is less efficient (see Sec. IV-D3). In each iteration, two new submodular pairwise cliques are created for each triple clique based on current solution. Furthermore, while dynamic programming can be used to efficiently compute unary terms, the same ideas do not apply to pairwise terms. Thus, optimization of quadratic approximation is more time consuming, compared to the unary case. Nonetheless, approximation based

on pairwise ICM may be useful for optimization of energies other than $E_{convexity}$.

IV. EXPERIMENTS

Below we apply convexity shape prior to image segmentation. We discretized orientations using 11×11 stencil yielding 40 orientations for all synthetic images. For natural images we found a 5×5 stencil yielding 8 orientations sufficient. Note, since the convexity term is NP-hard and our optimization is local, in general, we cannot guarantee convex results. However in practice the number of violations is either zero or negligible.

We collected a new dataset with ground truth convex segmentations. The code and the dataset are available from <http://vision.csd.uwo.ca/code/> and <http://vision.csd.uwo.ca/code/>.

The remainder of this section is structured as follows. First in Sec.IV-A we apply our method as described in Algorithm 1 (i.e. using linear approximation and dynamic programming) to synthetic and natural images. We call this approach ‘‘Direct’’ to distinguish it from subsequent variants. Next, in Sec.IV-B we experiment with alternative *central-clique* and *local* convexity models. In Sec. IV-C we compare our ‘‘Direct’’ optimization with existing state-of-the-art optimization methods. Finally, in Sec. IV-D we discuss limitations of our direct approach and evaluate two possible solutions: Gradual Non-Submodularization in Sec. IV-D2 and trust region based on quadratic approximation in Sec. IV-D3.

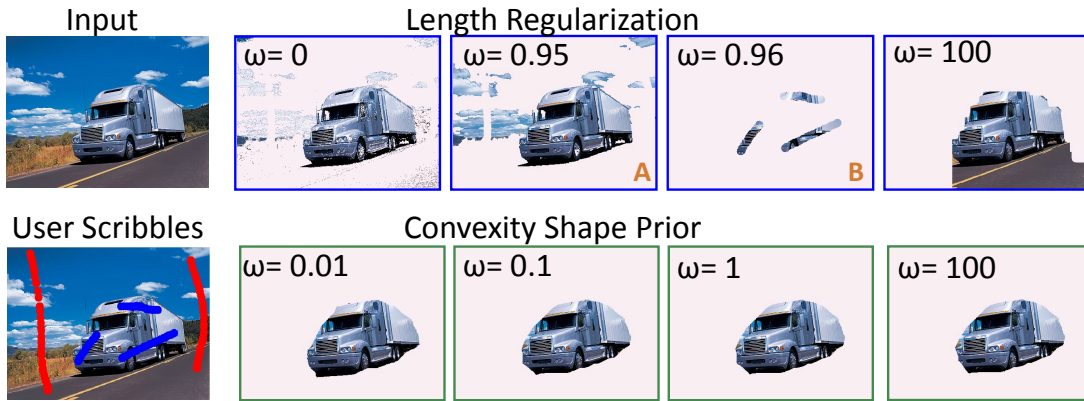


Fig. 6. Illustration of robustness to parameter ω : results for length regularization are shown with blue color and for convexity shape prior - with green. See text for details.

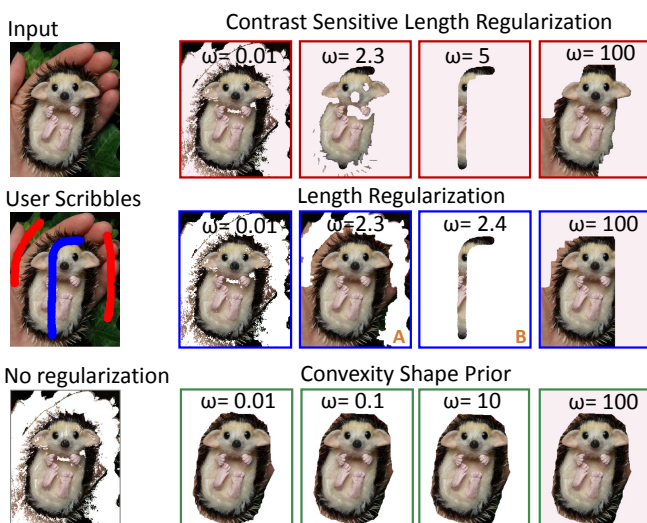


Fig. 7. Illustration of robustness to parameter ω : results for length regularization are shown with blue color and for convexity shape prior - with green. See text for details.

A. Direct approach: Trust Region with Linear Approximation

First we validate our method on synthetic images with noise $\mathcal{N}(0, 0.2)$, see Fig. 4. We assume given target appearance distributions for foreground and background, and combine standard log-likelihood data terms with the convexity shape prior

$$E(\mathbf{x}) = E_{\text{appearance}}(\mathbf{x}) + E_{\text{convexity}}(\mathbf{x}).$$

Here $E_{\text{appearance}}(\mathbf{x}) = \sum_{p \in \Omega} D_p(x_s)$ is the appearance term, $D_p(x_s) = -\log \text{Pr}(I_p | x_s)$ and $E_{\text{convexity}}(\mathbf{x})$ is as in (4). Fig. 4 demonstrates that our convexity prior removes noise, insures connectivity and fills in holes while preserving sharp corners.

Next, we use convexity prior in interactive segmentation of natural images with user scribbles. The convexity prior is especially useful for images where there is an overlap between the foreground and background appearance, see Figures 6-10. Such overlap often leads to holes in the foreground or

larger parts of the background erroneously segmented as the foreground. The convexity prior prevents such results. Length regularization is either too weak to remove the noise or too strong causing shrinking.

We now specify the details of our interactive segmentation. For appearance we use the recent submodular L_1 color separation term proposed in [26]. This term is based on L_1 distance between unnormalized histograms of foreground and background colors. Unlike standard appearance models, the color separation does not require re-estimation of the parameters and can be efficiently and globally optimized. We use 16 bins per color channel and combine the color separation term with the convexity prior, subject to hard constraints on the user scribbles

$$E(\mathbf{x}) = E_{L_1}(\mathbf{x}) + E_{\text{convexity}}(\mathbf{x}).$$

We then compare with the standard length regularization

$$E(\mathbf{x}) = E_{L_1}(\mathbf{x}) + \sum_{(pq) \in \mathcal{N}} \omega [x_s \neq x_t].$$

Figures 6-8 show segmentation results on two natural and one medical image. We vary the weight ω for our convexity prior in (3) and optimize as discussed in Sec. III. Similarly, we vary the weight ω for the length regularization and optimize with one graph-cut [26]. We show the resulting segmentations and compare them with the results obtained without regularization. We use green frames - for convexity, blue - for length. For length regularization we chose values ω for which the segmentations change significantly. The length regularization is either too weak to remove the noise or too strong and has a shrinking bias. Figure 10 shows similar results on additional natural images demonstrating the sensitivity of the length regularization and robustness of convexity shape prior.

We experiment both with contrast sensitive length (ω depends on pixel pair (p, q) as in [4]) and pure length, see Fig. 7. There is no significant difference in their performance. The same sensitivity to the parameter ω is observed; compare the red (contrast sensitive) frames with the blue ones (pure length).

Our model is parameter-free due to infinity cost constraints. In practice, we have to choose finite ω . There is almost no

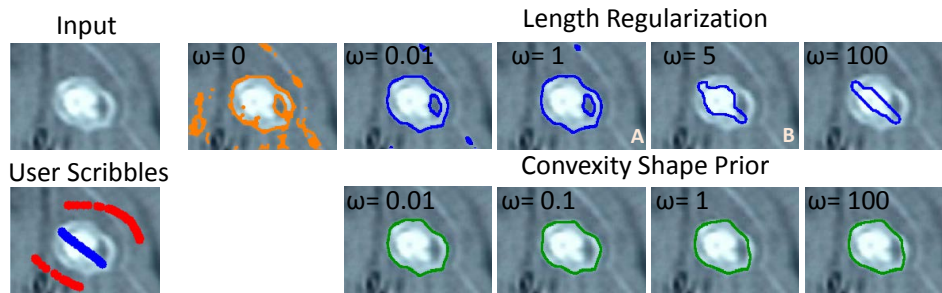


Fig. 8. Robustness to parameter ω : results for length regularization are shown with blue color and for convexity shape prior - with green. See text for details.



Fig. 9. Robustness to scale: segmentation results obtained for original images and their 25% and 11% scaled versions. We used $\omega = 10$ and 16 bins per color channel in all experiments. User scribbles were scaled accordingly for each experiment.

variation in segmentation results for different values of ω , once the value is high enough, making it a very robust regularizer. In fact, for each image we can compute finite ω such that violating a single constraint is more expensive than the initial solution. In cases where using such large ω leads to poor local minimum, gradually increasing the weight ω (annealing) can potentially escape to a better solution (see Gradual Non-Submodularization in sec. IV-D2).

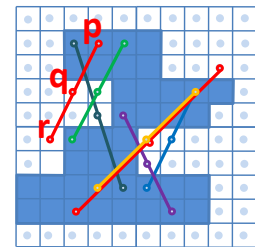
There is also very little variation in segmentation results with change of scale. Figure 9 shows segmentation results obtained on several natural images and their 25% and 11% scaled versions using the same parameters for ω and appearance. User scribbles were scaled accordingly in each experiment.

B. Alternative convexity models

Below we discuss some variations of our convexity model.

1) Central Cliques Model:

One natural question regarding our model is whether we need *all* the triple cliques on a line to enforce convexity. Indeed, it is sufficient to use a smaller subset consisting only of *central* triple cliques (p, q, r) , i.e. $|p - q| = |q - r|$,



see example on the right. This reduces the number of triple cliques from $O(n^3)$ to $O(n^2)$ for a line with n pixels. However, our dynamic programming procedures for evaluation (Sec. II-A) and for approximation (Sec. III-B) are no longer applicable. Brute force computation takes $O(n^2)$ operations per line with n pixels, as opposed to linear time with our dynamic programming for the full model. Nonetheless, we compare between our full model and the central cliques model, see Fig.11. Since the resulting segmentations have no convexity violations their energies can be directly compared. The energy is slightly better with the central cliques, but its running time is 25-30 times longer. The difference in time will be even more significant for larger images.

2) *Local Convexity Model*: Another natural variation of our model is the local convexity model, in which we penalize local spatially symmetric configurations 1-0-1 within a fixed size neighborhood around each pixel. The number of triple cliques is linear in the number of image pixels and the neighborhood size. Similarly to our full model, the local convexity model can be optimized using our trust region method. Our dynamic programming is not directly applicable.

The local model can only enforce local convexity. Namely, it is blind to violations of convexity that lie at a scale larger than the neighborhood size. Therefore the results can contain holes, multiple connected components and large scale concavities. Consider a 100×100 synthetic image example in Fig. 13. With a small neighborhood size, the number of triple cliques is small and computation is efficient, but the results are far from convex (Fig. 13, top-right). Moreover, small neighborhood size yields very coarse discretization of orientation. As we increase the neighborhood size, the number of convexity violations is reduced (Fig. 13, bottom row), eventually reaching a convex segmentation for a large enough neighborhood. However, the

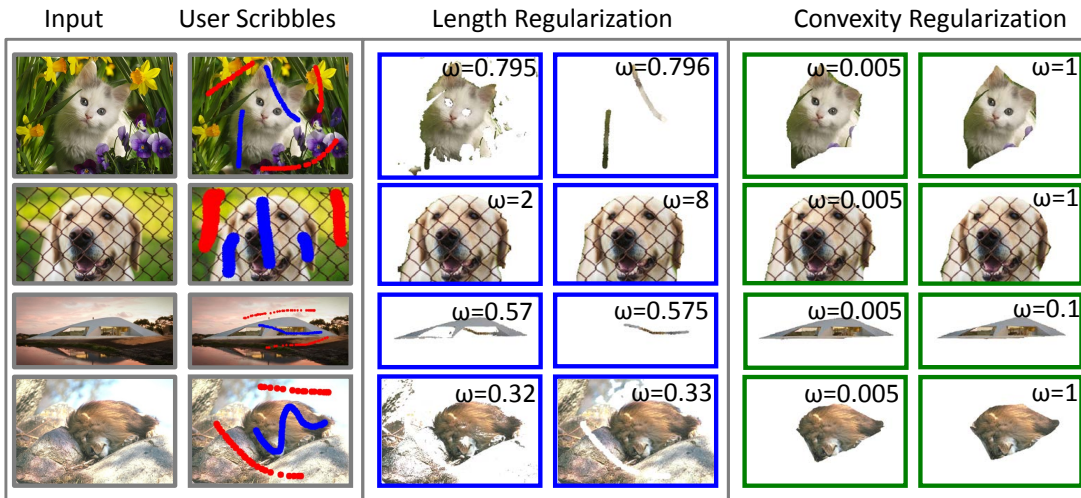


Fig. 10. Additional results comparing length and convexity regularizers. Except for the second row, all images demonstrate sensitivity to length weight ω .

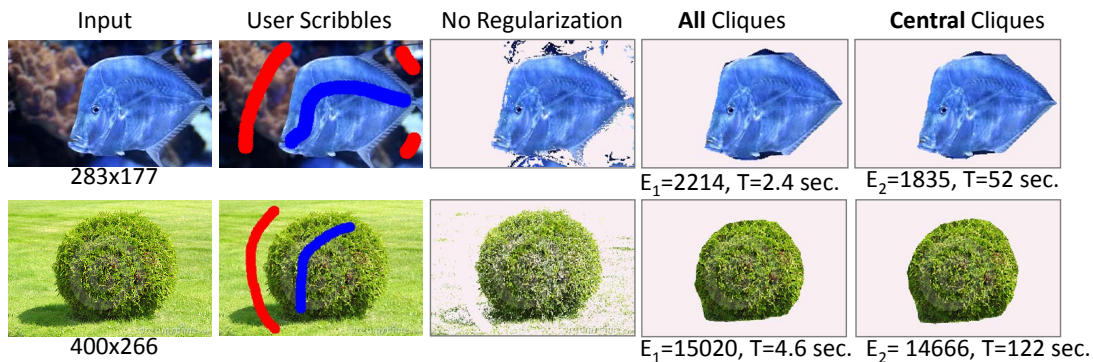


Fig. 11. Comparison between the full convexity model with all triple cliques vs. central cliques convexity model. The later does not allow efficient evaluation and approximation of $E_{convexity}$ using dynamic programming and therefore is much slower.

model becomes very slow to optimize.

C. Alternative Optimization Methods

The most related optimization method to ours is LSA [9], which is also based on trust region framework. However, it was designed for non-submodular energies with only *pairwise* cliques. For this class of energies, LSA reports state-of-the-art results [9]. LSA approximates the energy by replacing non-submodular pairwise cliques with their Taylor expansions while preserving all submodular pairwise cliques. Even though our convexity prior is not pairwise, it is possible to reduce each triple clique to several pairwise potentials using an additional auxiliary node [15] and optimize them with LSA. We call this reduced version r-LSA. Reducing all triple cliques would result in a prohibitively large number of auxiliary nodes and pairwise cliques for our full convexity model. Even for the central clique model the number of resulting pairwise cliques is quite large. For an $n \times n$ image central cliques model with m orientations produces $O(mn^3)$ pairwise potentials, which is very costly both to optimize and to evaluate. Nonetheless, we tried this approach on a small 91×122 image. The first two rows in Figure 12 compare r-LSA approach to our method.

We apply both methods to the central clique model⁵ and vary ω . Note that r-LSA is an order of magnitude slower than the slow version of our method. As the value of ω increases, r-LSA fails to obtain satisfactory solutions. We believe that there could be serious drawbacks in splitting clique $\phi(x_s, x_t, x_v)$ into individual submodular and supermodular parts and then approximating the supermodular part. One sign of a problem is that there are infinitely many such decompositions and it is not clear which one would give a better approximation.

Our full model with all triple cliques is also prohibitively expensive for standard optimization methods designed for non-submodular energies, such as QPBO [2] and TRWS [14]. However, we can use these methods to optimize the more compact central clique model as well. The last two rows in Figure 12 show segmentation results of QPBO and TRWS for several values of ω . For values of ω that are not sufficiently large to enforce convexity, all four methods, QPBO, TRWS, r-LSA and Trust Region, return globally optimum, but useless solutions. However, when ω is large enough, QPBO, TRWS

⁵Even though the full model is more efficient for our method, for this experiment we use central cliques to have identical energies for direct comparison.
















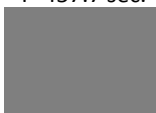

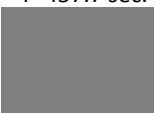






	$\omega=0.1$	$\omega=0.18$	$\omega=0.19$	$\omega=1$	$\omega=5$	$\omega=10$
Ours no DP	 E=3569.3 T=1.2 sec.	 E=6424.7 T=1.6 sec.	 E=6781.7 T=1.9 sec.	 E=26653 T=10.9 sec.	 E=34925 T=8.3 sec.	 E=38320 T=18.9 sec.
r-LSA	 E=3569.3 T=38.4 sec.	 E=6424.7 T=40 sec.	 E=6781.7 T=42.2 sec.	 E=26612 T=457.7 sec.	 E=46305 T=417.9 sec.	 E=356930 T=457.7 sec.
QPBO	 E=3569.3 T=38.3 sec.	 E=6424.7 T=389.8 sec.	 none T=444 sec.	 none T=426.08 sec.	 none T=433 sec.	 none T=481.69 sec.
TRWS	 E=3569.3 T=247.8 sec.	 E=6424.7 T=307.7 sec.	 E=6781.7 T=540.9 sec.	 E=68345 T=8270 sec.	 E=163065 T=8596.4 sec.	 E=259880 T=8377 sec.

Fig. 12. Comparison between our method without dynamic programming (no DP), r-LSA, QPBO and TRWS on the central clique model with 8 orientations. We use thresholded appearance terms for initialization when needed. As ω increases, making the energy more difficult, QPBO was not able to label any pixel (shown in gray) and TRWS did not converge after 5000 iterations, which took several hours. For ω large enough to enforce convexity, all methods except ours fail.

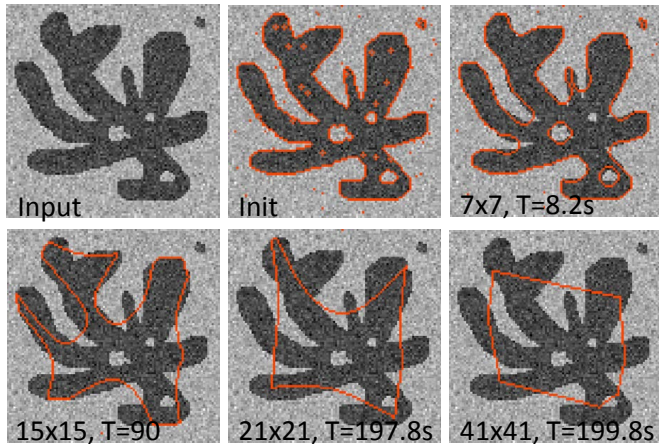


Fig. 13. Local Convexity Model: penalizes local 1-0-1 configurations within a fixed neighborhood around each pixel. Small neighborhoods are efficient to optimize but have very coarse discretization of orientation. Moreover, they are sensitive to noise and might result in multiple disconnected components. Larger neighborhood are slow to optimize.

and r-LSA fail to obtain a satisfactory result. Our trust region approach obtains good results in all cases.

D. Optimization Limitations and Proposed Solutions

1) *Local Minimum*: Trust region framework is a local iterative optimization and therefore we can only guarantee a local minimum [10]. Figure 14 demonstrates some sensitivity with respect to initialization. A trivial initialization with all

pixels in the foreground, denoted by “init 1” and delineated by the red contour, leads to a local minimum. Initializing with the maximum likelihood label per pixel, denoted by “init 2” results in solution with better energy. Empirically, we obtain better results starting with maximum likelihood labels. This is consistent for all the experiments, both on synthetic and real images.

2) *Gradual Non-Submodularization*: As shown in the experiments with the natural images, segmentation with convexity shape prior yields robust results for different values of the weight ω , once they are high enough. However, we could design a synthetic example (Fig. 15) where caution should be used when selecting the weight ω in (3). High values of ω tend to cause more aggressive optimization steps, and might lead to a solution that is convex but far from a global optimum (Fig. 15, top row, third column). Using low values of ω allows deviation from convexity (Fig. 15, bottom-left). To obtain a convex solution that avoids such local minima, the weight ω of the convexity shape prior can be increased gradually, similarly to the idea of “Graduated Non-Convexity” [1]. We can start with a very low weight, where the energy is mostly guided by the submodular appearance terms and is easier to optimize. Then the result is used for initialization of the next round, in which the weight ω is slightly increased. See an example of such *gradual non-submodularization* on a synthetic image in Fig.15 (bottom row).

For our discrete energy, gradual increase in weight ω makes the energy gradually more non-submodular, more non-linear and, therefore, more difficult. Similar improvements over

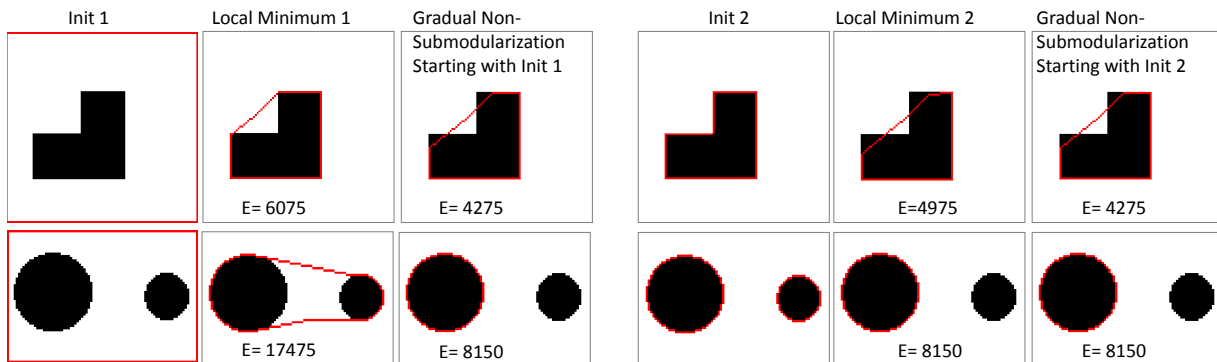


Fig. 14. Local optimization of convexity shape prior might yield different segmentation results for different initializations. We used an 11×11 mask to discretize orientation, $\omega = 100$ and given target appearance models $(\mu_{fg} = 0, \sigma_{fg} = 0.1)$, $(\mu_{bg} = 1, \sigma_{bg} = 0.1)$. Gradual Non-Submodularization yields global minimum solutions (verified by geometrical arguments) using any initialization (see text for details).

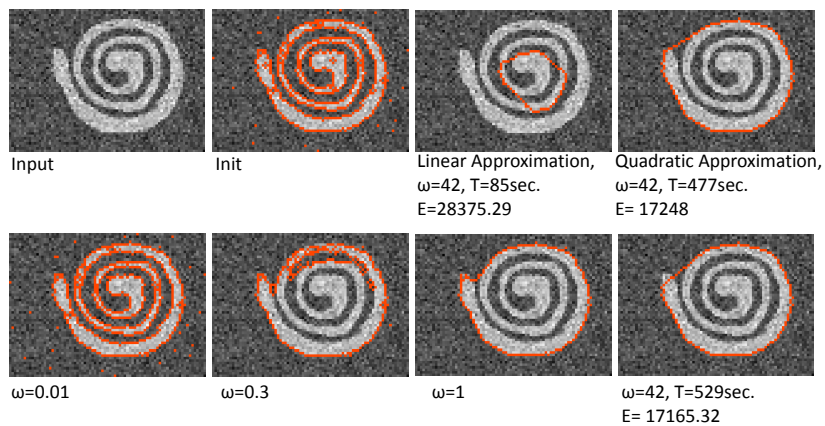


Fig. 15. Gradual Non-Submodularization with Linear Approximation vs. Quadratic Approximation. First row: input image, initial solution based on data terms, direct optimization with linear approximation resulting in weak local minimum, and direction optimization with quadratic approximation resulting in a satisfying solution. Second row: Gradual non-submodularization. Each increase in ω starts new optimization using previous result as initialization. We used standard log-likelihood data terms: $(\mu_{fg} = 0, \sigma_{fg} = 0.1)$, $(\mu_{bg} = 1, \sigma_{bg} = 0.1)$.

direct aggressive optimization are obtained in other synthetic (see Fig. 14) and medical (see Fig. 16) images.

In Fig. 16 we apply convexity shape prior for liver segmentation on an MRI image. Here again, high values of ω result in a convex but not accurate solution (top-right). Values of ω that are too low allow deviation from convexity (bottom-left). *Gradual non-submodularization* (bottom row) escapes such bad local minima and results in a solution with a lower energy.

Gradual non-submodularization can also alleviate the sensitivity of our trust region method to initialization. In Fig. 14 we were able to obtain the same global optimum starting with either “init 1” or “init 2” for both synthetic examples.

To further evaluate the advantage of gradual non-submodularization we collected a database of 51 natural images. We manually segmented each image to obtain ground truth and convexified the resulting foreground segment to insure that ground truth is convex. We performed two sets of experiments: using 5×5 orientations stencil (i.e. 8 orientations) and 11×11 stencil (i.e. 40 orientations.) In each experiment we run both, our direct approach with $\omega = 10$ and the gradual non-submodularization approach with the schedule

$\omega = \{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. All methods were initialized with the solution that minimizes submodular L_1 color separation term proposed in [26] subject to user scribbles constraint. Recall that L_1 color separation does not require re-estimation of the parameters and can be efficiently and globally optimized. We used 16 bins per color channel.

We then compared the results in terms of mean running time (in sec.), distance from the ground truth (percentage of misclassified pixels) and energy (percentage of images in which gradual approach obtains strictly lower energy. Table I summarizes the results. While having a slightly longer mean running time, gradual approach achieves more accurate results with respect to the ground truth for both 5×5 and 11×11 . In terms of energy, gradual approach achieves lower energy in more than 84% of the images.

Figure 17 shows several images from the database. The database can be downloaded from <http://vision.csd.uwo.ca/data/>. Rows 1, 4-6 show examples where gradual method obtains lower energy than the direct approach with both 5×5 and 11×11 orientation stencils. Second row shows an example where direct and gradual approach converge to the same solution. Third row shows an example where in case of 11×11

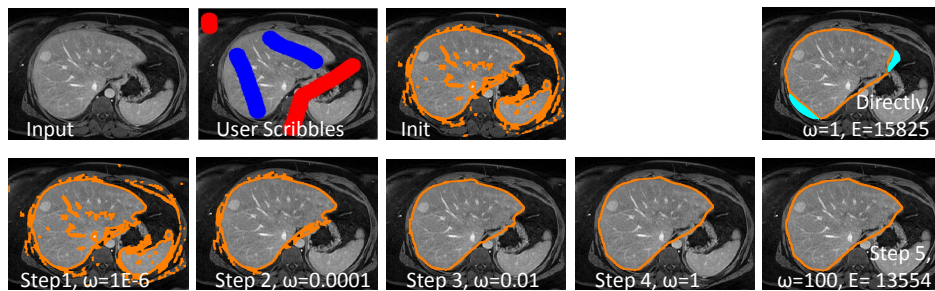


Fig. 16. First row: liver MRI image, user scribbles, initial solution based on data terms, direct optimization with $\omega = 1$ resulting in weak local minimum. Second row: Gradual non-submodularization. See text for details. Here for appearance we used L_1 color separation from [26].

		Direct Approach	Gradual Non-Subm.	Gradual Approach Achieves Lower Energy in
5×5	Error	2.5%	2.1%	86%(44/51)
	Runtime	10.9s.	15.4s.	
11×11	Error	2.5%	1.9%	84%(43/51)
	Runtime	110.7s.	147.4s.	

TABLE I

QUANTITATIVE COMPARISON OF DIRECT AND GRADUAL NON-SUBMODULARIZATION APPROACH IN TERMS OF MEAN RUNNING TIME, MEAN NUMBER OF MISCLASSIFIED PIXELS WITH RESPECT TO THE GROUND TRUTH. THE LAST COLUMN SHOWS THE PERCENTAGE OF IMAGES IN THE DATABASE FOR WHICH GRADUAL APPROACH ACHIEVED LOWER ENERGY.

gradual method converges to a solution with less violations of convexity but worse color separation resulting in overall higher energy. Such cases comprise less than 15% of the database. The visual difference between 5×5 and 11×11 stencils is not significant and in most cases 5×5 stencil is sufficient to obtain good results on natural images. Note that convexity model with 5×5 is relatively coarse and might result in solutions that have large scale concavities (second row) or several disconnected component (last row) that do not violate any of the triple constraints. In such cases, we advise to use higher resolution of orientation.

3) *Quadratic Approximation Models*: Another possible way to improve optimization is to use more accurate approximation models. In Sec. III-C we introduced quadratic approximation model based on parallel pairwise ICM. For our dense convexity model, this approximation is very inefficient. First, it requires rebuilding the graph in each iteration for incorporating new quadratic potentials for each triple clique. Second, evaluation of quadratic approximation cannot benefit from the speed-up obtained with the linear dynamic programming. However, quadratic approximation is more accurate than the linear approximation model and can yield better segmentation results. Figure 15 shows an illustrative example of a synthetic binary image with artificially added noise. We show solutions obtained with linear (top row, third column) and quadratic (top-right) approximations and juxtapose them with gradual non-submodularization (bottom row).

To further evaluate the quality of our quadratic approximation model we collected an additional set of images. Due

to the slow running time we chose very small 50×50 yet challenging synthetic binary images with artificially added noise. In nine out of ten images, quadratic approximation yielded solutions with lower energy. The average running time was 27.15 and 36.75 sec. for linear and quadratic approximation models respectively. Note that the small difference in running time is due to the small size of the images. Figure 18 shows input images (leftmost column), initial solutions (second column), results obtained with linear approximation (third column) and results obtained with quadratic approximation (rightmost column). Though for our dense convexity model quadratic approximation is very inefficient, it might be useful for other high-order energies where the number of high-order cliques is not as high.

V. CONCLUSION AND FUTURE WORK

We propose convexity prior as a new regularizer and develop efficient discrete optimization based on trust region and dynamic programming. Our convexity prior does not have shrinking bias and is robust w.r.t. parameter ω ,

In the future, we plan to explore meaningful extensions of strict convexity. For example, we can explore contrast sensitive convexity, which is similar to contrast sensitive length. The penalty for convexity violation by triplet (p_s, p_t, p_v) can carry a smaller weight if there is a high contrast on a line connecting pixels p_s and p_v . This formulation is straightforward, but the main difficulty will be extending our dynamic programming algorithms to handle this case. Another direction is to extend our model to handle objects with multiple convex parts.

VI. ACKNOWLEDGEMENTS

We are thankful for generous support by Canadian NSERC Discovery and RTI Programs

REFERENCES

- [1] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA, USA: MIT Press, 1987.
- [2] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, p. 2002, 2001.
- [3] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *International Conference on Computer Vision*, 2003, pp. 26–33.
- [4] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *IEEE International Conference on Computer Vision (ICCV)*, 2001.

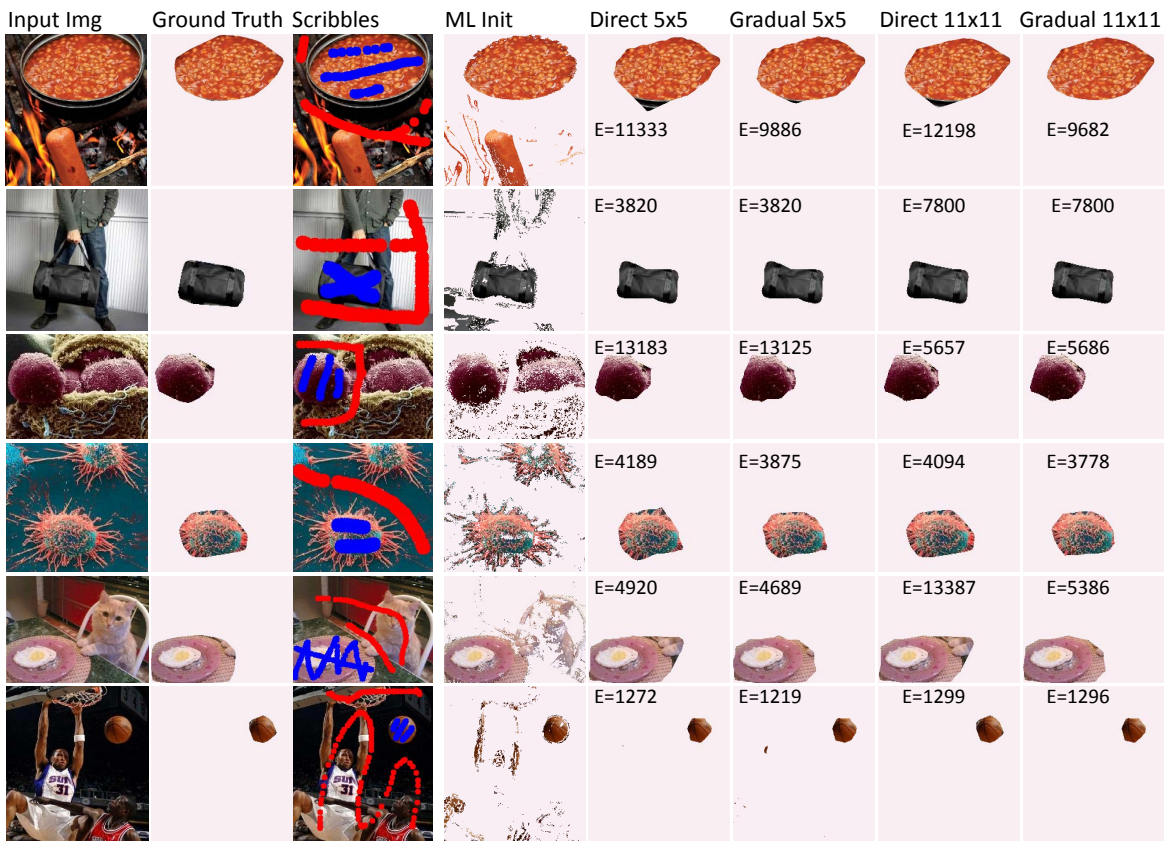


Fig. 17. Qualitative comparison of direct and gradual approach using 5×5 and 11×11 orientation stencils. See text for details.

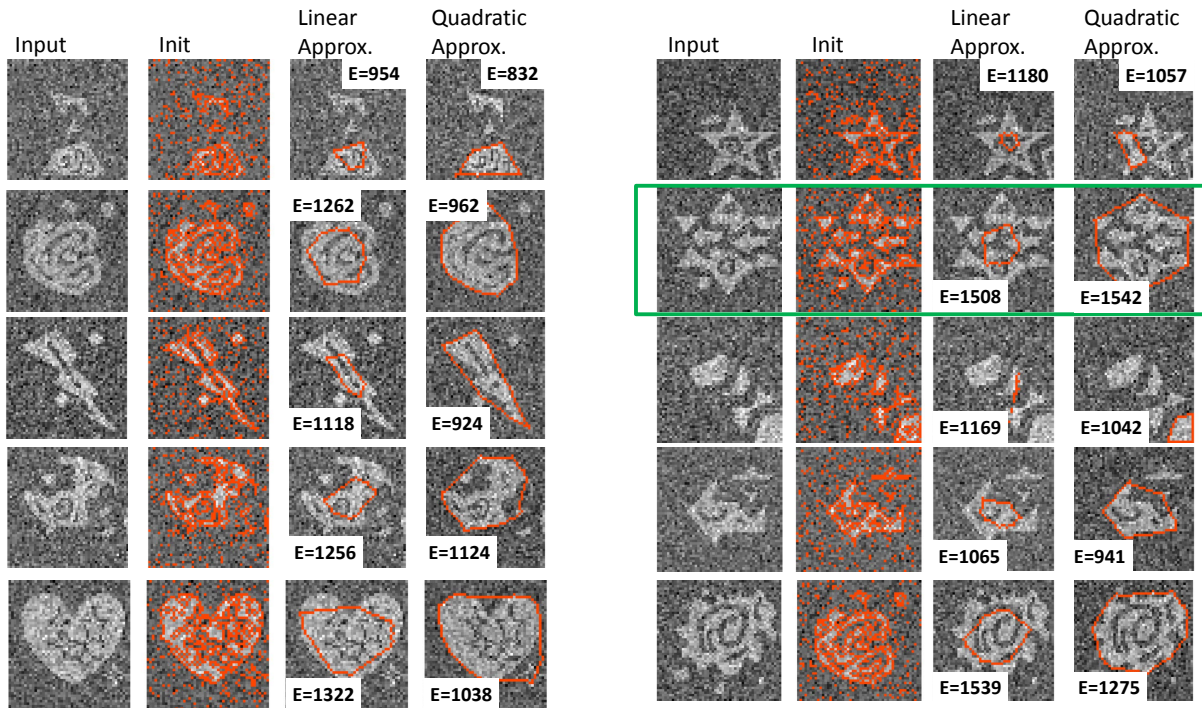


Fig. 18. Qualitative comparison of linear and quadratic approximation using 11×11 orientation stencils. Left-to-right: Input image, initial solution using maximum likelihood label per pixel based on the appearance terms, solution with linear approximation, solution with quadratic approximation. We used standard log-likelihood data terms: $(\mu_{fg} = 0, \sigma_{fg} = 0.4)$, $(\mu_{bg} = 1, \sigma_{bg} = 0.4)$. Quadratic approximation obtained lower energy for 9 out of 10 images. Green frame highlights the negative example.

- [5] 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.
- [6] K. Bredies, T. Pock, and B. Wirth, "Convex relaxation of a class of vertex penalizing functionals," *J. Math. Imaging and Vision*, vol. 47, no. 3, pp. 278–302, 2013.
- [7] P. Felzenszwalb and O. Veksler, "Tiered scene labeling with dynamic programming," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [8] S. Fujishige, "Submodular functions and optimization," *Annals of Discrete Mathematics*, 1991.
- [9] L. Gorelick, Y. Boykov, O. Veksler, I. Ben Ayed, and A. Delong, "Submodularization for binary pairwise energies," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 1154–1161.
- [10] 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-ftp-abs.shtml>
- [11] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010. [Online]. Available: <http://research.microsoft.com/apps/pubs/default.aspx?id=122786>
- [12] S. Jegelka and J. Bilmes, "Submodularity beyond submodular energies: Coupling edges in graph cuts," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1897–1904.
- [13] P. Kohli, A. Osokin, and S. Jegelka, "A principled deep random field model for image segmentation," in *CVPR*. IEEE, 2013, pp. 1971–1978.
- [14] V. Kolmogorov and T. Schoenemann, "Generalized sequential tree-reweighted message passing," *arXiv:1205.6352*, 2012.
- [15] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, 2004.
- [16] 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.
- [17] X. Liu, O. Veksler, and J. Samarabandu, "Order-preserving moves for graph-cut based optimization," *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, pp. 1182–1196, 2010.
- [18] Z. Liu, D. Jacobs, and R. Basri, "The role of convexity in perceptual completion: beyond good continuation," *Vision Research*, vol. 39, pp. 4244–4257, 1999.
- [19] P. Mamassian and M. Landy, "Observer biases in the 3d interpretation of line drawings," *Vision Research*, vol. 38, pp. 2817–2832, 1998.
- [20] C. Nieuwenhuis, E. Töppe, L. Gorelick, O. Veksler, and Y. Boykov, "Efficient regularization of squared curvature," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 4098–4105.
- [21] S. Nowozin and C. H. Lampert, "Global interactions in random field models: A potential function ensuring connectedness," *SIAM J. Imaging Sciences*, vol. 3, no. 4, pp. 1048–1074, 2010.
- [22] C. Olsson, J. Ulen, Y. Boykov, and V. Kolmogorov, "Partial enumeration and curvature regularization," in *International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.
- [23] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "Global solutions of variational models with convex regularization," *SIAM Journal on Imaging Sciences*, vol. 3, pp. 1122–1145, 2010.
- [24] T. Schoenemann, F. Kahl, S. Masnou, and D. Cremers, "A linear framework for region-based image segmentation and inpainting involving curvature penalization," *Int. Journal of Computer Vision*, 2012.
- [25] E. Strelakovsky and D. Cremers, "Generalized ordering constraints for multilabel optimization," in *International Conference on Computer Vision (ICCV)*, 2011.
- [26] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, "Grabcut in one cut," in *International Conference on Computer Vision*, 2013.
- [27] O. Veksler, "Star shape prior for graph-cut image segmentation," in *European Conference on Computer Vision (ECCV)*, 2008, pp. 454–467.
- [28] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. [Online]. Available: <http://research.microsoft.com/apps/pubs/default.aspx?id=80485>
- [29] J. Winn and J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 37–44.
- [30] Y. Yuan, "A review of trust region algorithms for optimization," in *The Fourth International Congress on Industrial & Applied Mathematics (ICIAM)*, 1999.



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 with applications to medical imaging.



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 research 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 (department of Radio Engineering and Cybernetics) 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.



Claudia Nieuwenhuis received the MS (Diplom) degree in computer science from the Technical University of Ilmenau, Germany, in 2006 and the MS (Diplom) degree in mathematics from the University of Hagen, Germany, in 2008. In 2009, she received the PhD degree in computer science from the University of Heidelberg, Germany. After receiving the PhD degree she was a postdoctoral fellow at the University of Frankfurt, Technical University of Munich, university of Western Ontario and University of California Berkeley.