Supplementary Material: Adaptive and Move Making Auxiliary Cuts for Binary Pairwise Energies

Lena Gorelick Yuri Boykov Olga Veksler
Computer Science Department
University of Western Ontario

Abstract

Below please find supplementary materials for our paper. First, we restate the optimization problem in Sec. 1. Because for some applications below it is more convenient to use a different form of energy, we show how to transform between the different formulations in Sec. 2.

We then provide full technical details for two applications: Multi-Region and Compact shape prior in Sections 3.1 and 3.2 respectively.

Finally, we provide additional experiments for the squared curvature regularization in Sec. 3.3, which were mentioned but not shown in the paper due to the lack of space.

1. Energy

We 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]

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

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

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

When energy (1) is *submodular*, *i.e.* $m_{pq} \leq 0 \quad \forall (p,q)$, 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] Note that such transformations are up to a constant, see Sec. 2.

2. Energy Transformation

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

$$E(S) = \sum_{p \in \Omega} D_p(s_p) + \sum_{(p,q) \in \mathcal{N}} V_{pq}(s_p, s_q), \tag{3}$$

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 (3) 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,

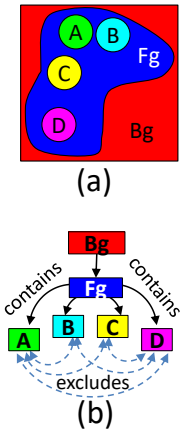
$$H = (h_p | p \in \Omega), \text{where } h_p = \sum_{(p,q) \in \mathcal{N}} (c_{pq} - a_{pq}).$$

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 (3) can be written in the form of (1) up to a constant $C = \sum_p D_p(0) + \sum_{(p,q) \in \mathcal{N}} a_{pq}$.



Five Binary Labels per Pixel in Our Graph					Img Pixel Label	Total Unary Cost $D_{(r,c)}(.)$
Fg	A	B	C	D		
0	0	0	0	0	Bg	$D_{r,c}$(Bg)
1	0	0	0	0	Fg	$D_{r,c}$(Fg)
1	1	0	0	0	A	$D_{r,c}$(A)
1	0	1	0	0	B	$D_{r,c}$(B)
1	0	0	1	0	C	$D_{r,c}$(C)
1	0	0	0	1	D	$D_{r,c}$(D)
Any other configuration						∞

(c)

Figure 1. Multi-region 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 multilabel cost. All other configurations have an infinite cost due to inclusion or exclusion violations.

3. Applications

3.1. 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 [3] for such multi-region objects. It 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 the non-submodular energy for MRI liver segmentation [4] that employs the multi-region model [3]. The image contains a liver with four distinct and mutually exclusive tumors. For completeness, we formally define the energy for our model using the form in (3). To convert this energy to the form in (1), see details in Sec. 2.

Given an image with N pixels, the liver is modeled by a graph with five layers of binary variables, corresponding to liver (Fg), and four tumors (A, B, C, D). See Fig. 1, (a-b) for a schematic illustration. Each layer has N nodes and each node has a corresponding binary variable. Inclusion of tumors within liver and exclusion constraints between tumors are implemented using submodular and supermodular inter-layer pairwise potentials respectively, see Fig. 1, (b). In addition, we use Potts regularization on each layer. 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, $l_p \in \{\text{Fg}, \text{A}, \text{B}, \text{C}, \text{D}\}$.

For Potts 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 $l \in \{\text{A}, \text{B}, \text{C}, \text{D}\}$, then

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

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. 1, (b). Let p and q be two nodes corresponding to the same pixel such that node p is in liver (Fg) layer and node q is in a tumor layer. That is $(r_q = r_p) \wedge (c_q = c_p)$ and $(l_p = \text{Fg}) \wedge (l_q \in \{\text{A}, \text{B}, \text{C}, \text{D}\})$. Inclusion pairwise potential $V^1_{p,q}$ 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

$$V^2_{p,q}(s_p, s_q) = \lambda_{sub} \cdot \begin{cases} \infty & \text{if } (s_p, s_q) = (0,1) \\ 0 & \text{otherwise.} \end{cases}$$

The tumor parts are mutually exclusive, see Fig. 1, (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 $l_p, l_q \in \{\text{A}, \text{B}, \text{C}, \text{D}\}$. Then the supermodular exclusion pairwise potential $V^3_{p,q}$ penalizes illegal configuration $(1, 1)$. Each pixel can only belong to one tumor. That is,

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

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

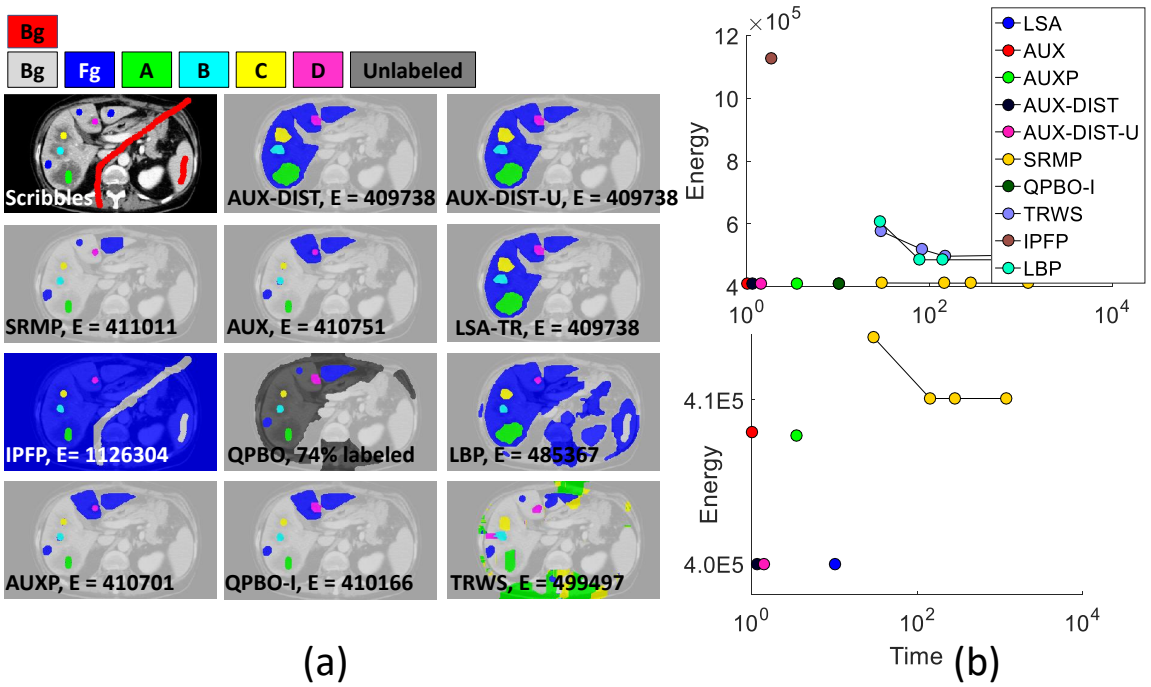


Figure 2. Multi-region liver segmentation. (a) input image, user scribbles - liver (blue) and tumors (green, yellow, cyan, magenta) along with segmentation results, (b) energy vs. time plot (top) with zoom in the (bottom). We set weights $\lambda_{sub} = \lambda_{sup} = 100$ and $\lambda_{\text{Potts}} = 25$.

illegal, *i.e*. having infinite cost. Figure 1, (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 (3) 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,

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

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.

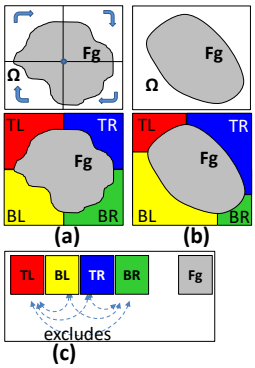
Figure 2 (a-b) shows the results. We use scribbles for appearance and as hard constraints. The top plot compares the methods in terms of energy and running time. The bottom plot zooms in on the most interesting part. Most methods arrived at poor solutions that have violations of inclusion and exclusion constraints. LSA-TR, AUX-DIST, AUX-DIST-U achieve the same lowest energy, with AUX-DIST being an order of magnitude faster.

3.2. Generalized Compact Shape Prior

In this section we provide technical details for the generalized compact shape prior in [4]. It is formulated as multilabel energy and is subsequently reduced to a binary non-submodular pairwise energy using reduction similar to that in Sec. 3.1. This new model generalizes *compact* shape prior proposed in [2]. Compact shape prior is useful in industrial part detection and medical image segmentation applications.

The compact shape prior in [2] 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 3, (a) shows an example of an object (along with user provided center) that can be segmented using the model in [2]. Allowed orientations for each quadrant are shown with blue arrows. In contrast, the generalized model [4] 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, the generalized model explicitly divides the background into four regions as in Fig. 3, (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



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	$D_{r,c}(\text{fg}) + 3D_{(r,c)}(\text{fg})$
0	1	1	1	bg (TL)	$D_{r,c}(\text{bg}) + 3D_{(r,c)}(\text{fg})$
1	0	1	1	bg (BL)	$D_{r,c}(\text{bg}) + 3D_{(r,c)}(\text{fg})$
1	1	0	1	bg (TR)	$D_{r,c}(\text{bg}) + 3D_{(r,c)}(\text{fg})$
1	1	1	0	bg (BR)	$D_{r,c}(\text{bg}) + 3D_{(r,c)}(\text{fg})$
Any other configuration					∞

(d)

Figure 3. Compact Shape Prior Illustration: (a-top) the model in [2], (a-bottom) the multilabel model in [4], (b-top) - an input silhouette that can be modeled with the generalized model but not with the model in [2] (see text for details), (b-bottom) demonstrates how we split the image into five regions in the generalized model, (c) schematic representation of the geometric exclusion constraints between the layers of our graph for the generalized model. (d) unary terms for each layer used in our graph.

blue arrows. For example, the red region can have discontinuity only in the up-right orientation.

The generalized model includes the model proposed in [2] as a special case when the transitions between different background labels are horizontally and vertically aligned as in (a-bottom). However, it is more general because the discontinuities between the background regions do not need to align. For example, the object in (b-top) cannot be segmented using the model in [2] but is easily modeled using the generalized compact shape prior (b-bottom). Below we formally define the energy for our model using the form in (3). To convert this energy to the form in (1) see details in Sec. 2.

Given an image with N pixels, we construct a graph with four binary layers: top-left (TL), top-right (TR), bottom-left (BL), bottom-right (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 schematic illustration of the inter-layer exclusion constraints in 3, (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, $l \in \{\text{TL}, \text{TR}, \text{BL}, \text{BR}\}$.

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,

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

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 constraints between the different background labels. For example the red region (TL) in Fig. 3, (a-bottom) cannot overlap any of the other background regions (TR, BL, BR). Modeling such interactions results in supermodular pairwise potentials.

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 $l_p, l_q \in \{\text{TR}, \text{TL}, \text{BR}, \text{BL}\}$. Then the supermodular exclusion pairwise potential $V^{\text{ex}}_{p,q}$ penalizes illegal configuration $(0,0)$. That is

$$V^{\text{ex}}_{pq}(s_p, s_q) = \begin{cases} \infty & \text{if } (s_p, s_q) = (0,0) \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

To interpret the optimal solution on our graph in terms of binary image segmentation, we consider a quadruple of corresponding binary graph nodes on layers TR, TL, BR and BL. We assign image pixel to foreground object (F) if all its corresponding graph nodes have label one, and to the background (B) otherwise, see table in Fig. 3, (d). As in [2], the generalized model can incorporate any unary term in (3) defined on image pixels, *e.g.* appearance terms. We now explain how to define the corresponding unary terms on the nodes of our four layer 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 $\{p = (r_p, c_p, l_p) | (r_p = r) \wedge (c_p = c)\}$. 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 number of supermodular terms V^{ex} in (4) 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, 6], however there is no known algorithm for finding the optimal switching for energies that are not permuted-submodular .

The generalized 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, the compact shape prior 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 [5]. 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 4, (left) shows an example of an input image with a hot-air balloon along with the user scribbles and the resulting appearance terms for each image pixel. Blue colors denote preference for the background and cyan-red colors - preference for the foreground. 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 $\lambda_{sub} = 250$ and $\lambda_{sup} = 500$ for the 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 shape prior. For each compared method we show the final image segmentation.

Figure 4, (right) 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 on monotonicity and coherence of the segment. This is due the high weight and large number of the supermodular terms. LSA-TR and AUX-DIST-U-EXP are the only methods that could optimize such energy, with AUX-DIST-U-EXP obtaining the lowest energy in shorter time.

3.3. Squared Curvature

Below we provide additional experiments for the Squared Curvature application, in which we compare the proposed extensions to standard optimization methods.

In Fig. 5, top we compare the best three extensions to AUX and AUXP and LSA-TR. All local optimization methods start with the maximum likelihood solution based on the appearance terms. When the weight of supermodular curvature terms increases, the proposed methods consistently outperform LSA-TR (blue line), AUX (red) and AUXP (green).

In the bottom of Fig. 5 we compare to other standard optimization methods such as QPBO, TRWS, SRMP, LBP and IPFP. All standard methods are significantly inferior even to the worst of the proposed extensions, AUX-DIST-U-EXP.

References

[1] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123:2002, 2001. 1, 5

[2] P. Das, O. Veksler, V. Zavadsky, and Y. Boykov. Semiautomatic segmentation with compact shape prior. *Image Vision Computing*, 27(1-2):206–219, 2009. 3, 4

[3] 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*, pages 285–292, 2009. 2

[4] L. Gorelick, Y. Boykov, O. Veksler, I. B. Ayed, and A. Delong. Local submodularization for binary pairwise energies. In *PAMI*, page accepted, 2017. 2, 3, 4

[5] 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*, pages 675–690, 2014. 5

[6] 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*, pages 28–38, 2007. 5

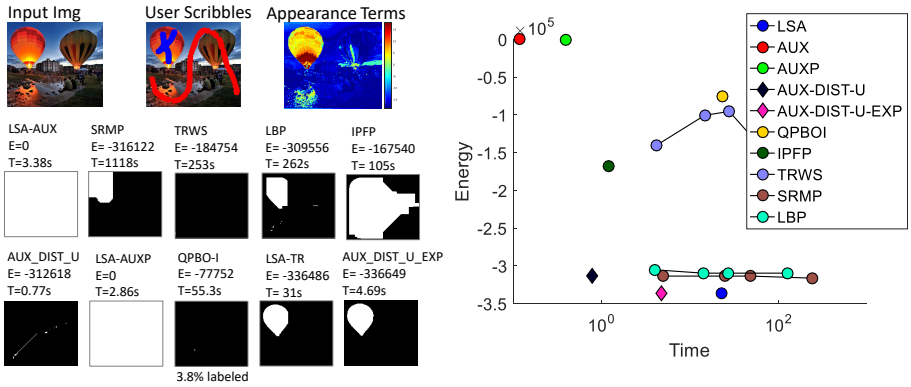


Figure 4. Compact Shape Prior: Left - segmentations results. The first row shows input image, user scribbles and resulting appearance terms. Red colors show preference to foreground and blue colors show preference to background. The remaining rows show for each method the final image segmentation. Right - comparison with other methods in terms of energy and the running time (shown in log-scale).

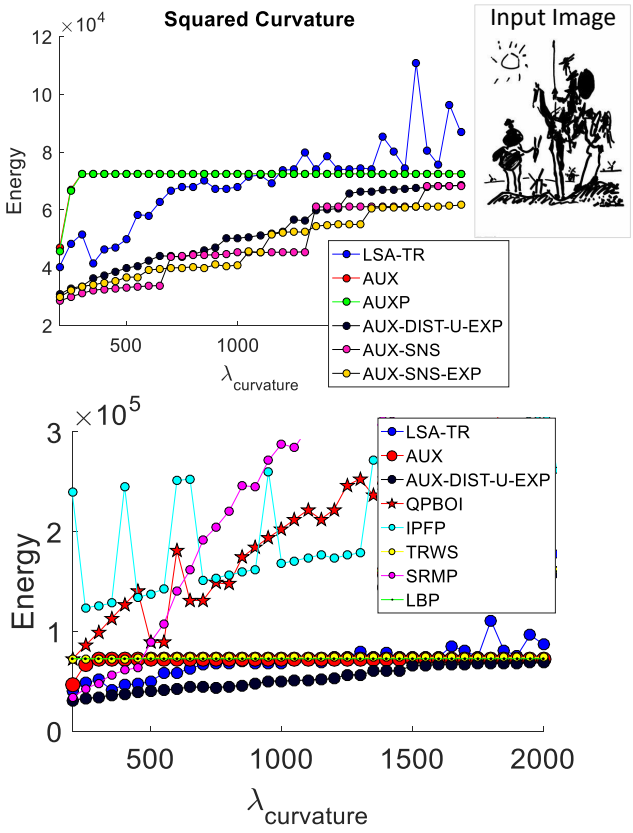


Figure 5. Squared curvature model. We used Gaussian with $(\mu = 0, \sigma = 0.2)$ and $(\mu = 1, \sigma = 0.2)$ for the foreground and background appearance models and 7×7 stencil for angular resolution. Top - comparison with AUX, AUXP and LSA-TR. Bottom - comparison with other standard optimization methods. All standard methods are significantly inferior even to the worst of the proposed extensions, AUX-DIST-U-EXP