

Fast Trust Region for Segmentation

Lena Gorelick¹

lenagorelick@gmail.com

Frank R. Schmidt²

schmidt@cs.uni-freiburg.de

Yuri Boykov¹

yuri@csd.uwo.ca

¹ Computer Vision Group
University of Western Ontario, Canada

² BIOSS Centre of Biological Signalling Studies
University of Freiburg, Germany

Abstract

Trust region is a well-known general iterative approach to optimization which offers many advantages over standard gradient descent techniques. In particular, it allows more accurate nonlinear approximation models. In each iteration this approach computes a global optimum of a suitable approximation model within a fixed radius around the current solution, a.k.a. trust region. In general, this approach can be used only when some efficient constrained optimization algorithm is available for the selected nonlinear (more accurate) approximation model.

In this paper we propose a Fast Trust Region (FTR) approach for optimization of segmentation energies with nonlinear regional terms, which are known to be challenging for existing algorithms. These energies include, but are not limited to, KL divergence and Bhattacharyya distance between the observed and the target appearance distributions, volume constraint on segment size, and shape prior constraint in a form of L_2 distance from target shape moments. Our method is 1-2 orders of magnitude faster than the existing state-of-the-art methods while converging to comparable or better solutions.

1. Introduction

In the recent years there is a general trend in computer vision towards using complex non-linear energies with higher-order regional terms for the task of image segmentation, co-segmentation and stereo [10, 7, 14, 2, 1, 11, 8]. In image segmentation such energies are particularly useful when there is a prior knowledge of the appearance model or the shape of an object being segmented.

In this paper we focus on segmentation energies that have the following form:

$$\min_{S \in \Omega} E(S) = R(S) + Q(S), \quad (1)$$

where S is a binary segmentation, $R(S)$ is a non-linear regional function, and $Q(S)$ is a standard length-

based smoothness term, e.g. quadratic submodular pseudo-boolean or continuous TV-based functional.

One straightforward approach to minimizing such energies could be based on gradient descent. In the context of level-set techniques the corresponding linear approximation model for $E(S)$ combines a first-order Taylor term for $R(S)$ with the standard curvature-flow term for $Q(S)$. Linear approximation model may work reasonably well for simple quadratic regional terms, e.g. area constraint $R(S) = (|S| - V)^2$ in [2]. However, it is well known that robust implementation of gradients descent for more complex regional constraints requires tiny time steps yielding slow running times and sensitivity to initialization [7, 1]. Significantly better optimization and speed are often achieved by methods specifically designed for particular regional constraints, e.g. see [1, 15, 16].

In this paper we propose a fast algorithm for minimizing general high-order energies like (1) based on more accurate non-linear approximation models and a general *trust region* framework for iterative optimization. We still compute a first-order approximation $U_0(S)$ for the regional term $R(S)$. However, we keep the exact quadratic pseudo-boolean (or TV-based) representation of $Q(S)$ instead of its linear (curvature flow) approximation. At each iteration we use non-linear approximation model

$$\tilde{E}(S) = U_0(S) + Q(S)$$

similar to those in [10, 14, 8]. Unlike [10, 14] we globally optimize such approximation models within a trust region $\|S - S_0\|_{L_2} \leq d$, which is a ball of certain radius d around current solution S_0 . The most closely related method is the *exact line-search* approach in [8]. At each iteration, they use a parametric max-flow technique to exhaustively explore solutions for all values of d and find the solution with the largest decrease of the original energy $E(S)$. We would like to point out that in general, the number of distinct solutions on the line in [8] can be exponential and we demonstrate that such exhaustive search is often too slow in practice.

Inspired by standard *backtracking* and *trust-region* frameworks [3, 17], at each step we find S^* minimizing approximation model $\tilde{E}(S)$ within ball $\|S - S_0\| \leq d$ of fixed radius d and adaptively adjust d for the next iteration depending on the current approximation quality $|E(S^*) - \tilde{E}(S^*)|$. As in [6, 8] we use a Lagrangian formulation for the trust-region sub-problem. One of our contributions is a derivation of a simple analytic relationship between radius d and Lagrange multiplier λ . This allows us to translate the standard adaptive scheme controlling the trust region radius d into an efficient adaptive scheme for the Lagrange multiplier λ . Consequently, we can use this fast scheme to replace the exhaustive search over λ in [8]. We demonstrate that our approach is not only faster by several orders of magnitude, but, surprisingly, also finds solutions as good as those obtained by the exhaustive *line search* [8].

The rest of the paper is as follows. Sec.2 reviews the trust region framework. Sec.3 details on the proposed Fast Trust Region approach to segmentation. Experimental results are presented in Sec.4 and our conclusions are given in Sec.5.

2. Overview of Trust Region Framework

Trust region is a general iterative optimization framework that in some sense is dual to the *gradient descent*, see Fig.1. While gradient descent fixes the direction of the step and then chooses the step size, trust region fixes the step size and then computes the optimal descent direction, as described below. At each iteration, an approximate model of the energy is constructed near the current solution. The model is only “trusted” within some small ball around the current solution, a.k.a. “trust region”. The global minimum of the approximate model within the trust region gives a new solution. This procedure is called *trust region sub-problem*. The size of the trust region is adjusted for the next iteration based on the quality of the current approximation.

Variants of trust region approach differ in the kind of approximate model used, optimizer for the trust-region sub-problem, and a merit function to decide on the acceptance of the candidate solution and adjustment of the next trust region size. For a detailed review of trust region methods see [17]. One interesting example of the general trust region framework is well-known *Levenberg–Marquardt* algorithm, which is commonly used in multi-view geometry to minimize non-linear re-projection errors [9].

Inspired by the ideas in [6, 8], we propose a trust region approach for minimizing high-order segmentation energies $E(S)$ in (1). The general idea outlined in Algorithm 1 is consistent with the standard *trust region* practices [3, 17]. Given current solution S_0 , energy $E(S)$ is approximated by

$$\tilde{E}(S) = U_0(S) + Q(S), \quad (2)$$

where $U_0(S)$ is the first order Taylor approximation of

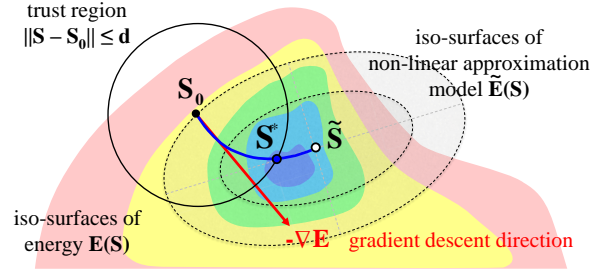


Figure 1. *Trust region* iteration $S_0 \rightarrow S^*$. Approximation $\tilde{E}(S)$ for energy $E(S)$ is constructed near S_0 . Solution S^* is obtained by minimizing \tilde{E} within a ball of radius d where the approximation is “trusted”. The step size d is adjusted for the next iterations depending on the approximation quality observed at S^* . The blue line shows the spectrum of possible *trust region* steps (moves). Small d gives steps aligned with *gradient descent* direction, while large d would give step \tilde{S} similar to Newton’s approach.

the non-linear term $R(S)$ near S_0 . The trust region sub-problem is then solved by minimizing \tilde{E} within a ball of given radius d (line 6)

$$S^* = \operatorname{argmin}_{\|S - S_0\| < d} \tilde{E}(S). \quad (3)$$

Once a candidate solution S^* is obtained, the quality of the approximation is measured using the ratio between the actual and predicted reduction in energy. Based on this ratio, current solution S_0 is updated in line 10 and the trust region size d is adjusted in line 12. It is common to set the parameter τ_1 in line 10 to zero so that any candidate solution decreasing the actual energy is accepted. The parameter τ_2 in line 12 is usually set to 0.25 [17]. Reduction ratio above τ_2 implies that approximation model \tilde{E} is sufficiently good within the current trust region and that the trust region size (step size) could be increased in the next iteration.

Algorithm 1: TRUST REGION FOR SEGMENTATION

- 1 $S_0 \leftarrow S_{init}, d \leftarrow d_{init}$
 - 2 **Repeat until convergence**
 - 3 //Compute approximation model (2) around S_0
 - 4 $U_0(S) \leftarrow$ Taylor exp. of $R(S)$ at S_0 //see details in [8]
 - 5 //Solve trust region sub-problem (3)
 - 6 $S^* \leftarrow \operatorname{argmin}_{S \in \Omega, \|S - S_0\| \leq d} \tilde{E}(S)$
 - 7 $\Delta P = \tilde{E}(S_0) - \tilde{E}(S^*)$ //predicted reduction in energy
 - 8 $\Delta A = E(S_0) - E(S^*)$ //actual reduction in energy
 - 9 //Update current solution
 - 10 $S_0 \leftarrow \begin{cases} S^* & \text{if } \Delta A / \Delta P > \tau_1 \\ S_0 & \text{otherwise} \end{cases}$
 - 11 //Adjust the trust region
 - 12 $d \leftarrow \begin{cases} d \cdot \alpha & \text{if } \Delta A / \Delta P > \tau_2 \\ d / \alpha & \text{otherwise} \end{cases}$
-

3. Our Algorithm

Constrained non-linear optimization (3) is a central problem for a general trust region approach. Section 3.1 shows how this problem can be solved using unconstrained Lagrangian formulation and states its properties. Section 3.2 discusses the relationship between trust region size d in (3) and Lagrange multiplier λ . Section 3.3 describe in detail our *Fast Trust Region* algorithm and Section 3.4 discusses its relation to gradient descent methods.

3.1. Lagrangian Formulation

Similarly to [8] we use the following unconstrained Lagrangian formulation for the trust region sub-problem (3):

$$L_\lambda(S) = \tilde{E}(S) + \frac{\lambda}{2} \text{dist}(\partial S_0, \partial S)^2, \quad (4)$$

where $\text{dist}(\cdot, \cdot)$ is a non-symmetric distance on the shape space defined on the segmentation's boundary as

$$\text{dist}(\partial S_0, \partial S) := \left[\int_{S_0} \min_{s \in S} \|s - s_0\|^2 ds \right]^{\frac{1}{2}}. \quad (5)$$

This distance can be approximated [6] using the integration of the signed distance function ϕ_0 of S_0 :

$$\text{dist}(\partial S_0, \partial S)^2 \approx \langle 2 \cdot \phi_0, S \rangle - \langle 2 \cdot \phi_0, S_0 \rangle. \quad (6)$$

The above approximation is linear and therefore $L_\lambda(S)$ can be minimized efficiently for any value of λ using graph-cut or TV-based methods. Below we state some basic properties of this Lagrangian formulation.

Property 1: Consider function $F : \mathbb{R}_+ \rightarrow \mathbb{R}$ defined via $F(\lambda) := \min_S L_\lambda(S)$. Each S induces a linear function $\lambda \mapsto L_\lambda(S)$ and F_λ is their lower envelope. Therefore, $F(\lambda)$ is a piece-wise linear concave function of λ with a finite (possibly exponential) number of break points [12].

Property 2: Let S_λ be the minimizer of $L_\lambda(S)$ in (4) and let λ_{\max} be the maximal break point of F . Namely,

$$\lambda_{\max} = \sup \{ \lambda | S_\lambda \neq S_0 \}.$$

By definition, for any $\lambda > \lambda_{\max}$, $S_\lambda = S_0$ and, therefore, $F(\lambda) = \tilde{E}(S_0) = \text{const}$. Since F is concave (Prop.1), F must also be monotonic non-decreasing function of λ with maximum at λ_{\max} (see Figure 2).

Property 3: For any $\lambda > 0$ it holds that

$$\tilde{E}(S_\lambda) \leq \tilde{E}(S_0). \quad (7)$$

Assume that there is a λ such that $\tilde{E}(S_\lambda) > \tilde{E}(S_0)$. Then,

$$\begin{aligned} F(\lambda) = L_\lambda(S_\lambda) &= \tilde{E}(S_\lambda) + \frac{\lambda}{2} \text{dist}(\partial S_\lambda, \partial S_0)^2 \\ &> \tilde{E}(S_0) = L_\lambda(S_0), \end{aligned}$$

contradicting the optimality of S_λ . More generally,

Property 4: The function $\lambda \mapsto \tilde{E}(S_\lambda)$ is monotonic non-decreasing (see Remark 1 in [6]).

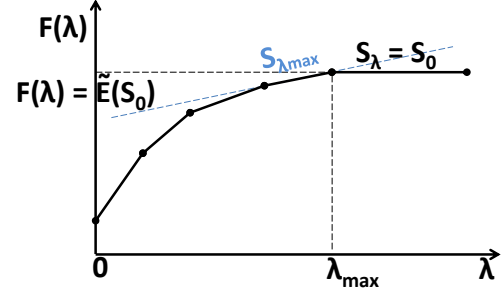


Figure 2. Each S induces a linear function $\lambda \mapsto L_\lambda(S)$. Their lower envelope yields the function $F(\lambda) = \min_S L_\lambda(S)$.

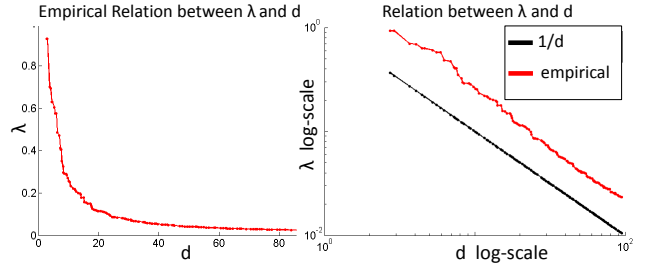


Figure 3. Empirical dependence between λ and d obtained in one typical iteration in our experiments (left). Using the log-scale in both λ and d it can be seen that the slope of empirical dependence is the same as that of $1/d$.

3.2. Relationship between λ and d

The standard trust region approach (see Algorithm 1) adaptively adjusts the distance parameter d . Since we use the Lagrangian formulation (4) to solve the trust region sub-problem (3), we do not directly control d . Instead, we control Lagrange multiplier λ . However, for each Lagrangian multiplier λ there is a corresponding distance d such that minimizer S_λ of (4) also solves (3) for that d . We can easily compute the corresponding value of $d = \text{dist}(\partial S_0, \partial S_\lambda)$. Figure 3(left) illustrates empirical dependence between λ and d obtained in one typical iteration in our experiments.

The relationship between λ and d can also be derived analytically. Consider the Lagrangian in (4) where $\text{dist}(\partial S_0, \partial S)$ is given by the approximation in (6), i.e.

$$L_\lambda(S) \approx \tilde{E}(S) + \lambda \langle \phi_0, S - S_0 \rangle. \quad (8)$$

Let S_λ be the minimizer of (8). Then it must satisfy

$$\begin{aligned} 0 &= \nabla \tilde{E}(S_\lambda) + \lambda \phi_0 \\ 0 &= \left\langle \nabla \tilde{E}(S_\lambda), S_\lambda - S_0 \right\rangle + \lambda \langle \phi_0, S_\lambda - S_0 \rangle \\ \lambda &= \frac{\left\langle \nabla \tilde{E}(S_\lambda), S_0 - S_\lambda \right\rangle}{d^2} \approx \frac{\tilde{E}(S_0) - \tilde{E}(S_\lambda)}{d^2} \end{aligned}$$

The last expression is obtained via a Taylor approximation. Note that the gradient that we used here is taken with

respect to the natural L_2 function space of relaxed segmentation. In particular, every segmentation $S : \Omega \rightarrow \{0, 1\}$ is also a function of the form $S : \Omega \rightarrow \mathbb{R}$.

Instead of writing \tilde{E} in a region-based form using the function S , we can also rewrite it in a contour-based form $\tilde{E}_C(\partial S) = \tilde{E}(S)$ using Green's formula. By applying again the Taylor approximation, we obtain

$$\begin{aligned} \lambda &\approx \frac{\langle \nabla \tilde{E}_C(\partial S_0), \partial S_\lambda - \partial S_0 \rangle}{d^2} \\ &\leq \frac{\|\nabla \tilde{E}_C(\partial S_0)\| \text{dist}(\partial S_0, \partial S_\lambda)}{d^2} = \frac{\|\nabla \tilde{E}_C(\partial S_0)\|}{d} \end{aligned} \quad (9)$$

We therefore can assume a proportionality between λ and $1/d$. This means that when the distance d_k is multiplied by a certain factor α , we instead divide λ by the same factor α . Figure 3(right) compares the empirical dependence shown on the left plot with the dependence given by $\lambda = 1/d$. Using log-scale for both d and λ , it can be seen that the slope of empirical dependence is the same as the slope of $1/d$ which justifies our heuristic.

3.3. Fast Trust Region (FTR)

In this section we describe our Fast Trust Region (FTR) algorithm. It is based on the high-level principles of the trust region framework presented in Algorithm 1, but uses Lagrangian formulation (4) instead of constrained optimization in (3). The relationship between Lagrange multiplier λ and distance d established in Section 3.2 allows us to translate the standard adaptive scheme for d in Algorithm 1 into an adaptive scheme for λ in our Algorithm 2. Note that we use parameter $\tau_1 = 0$ (see Algorithm 1) so that any decrease in energy is accepted.

We can show that our algorithm converges: in each iteration the method solves the trust region sub-problem with the given multiplier λ (Line 6). The algorithm either decreases the energy by accepting the candidate solution (line 20) or reduces the trust region (Line 23). When the trust region is so small that $S_\lambda = S_0$ (Line 9), one more attempt is made using λ_{\max} (see Property 2). If no reduction in actual energy is achieved using $S_{\lambda_{\max}}$ (Line 16), we have arrived at local minimum [6] and the algorithm stops (Line 17).

Following recommendations for standard trust region methods [17], we set parameter $\tau_2 = 0.25$ in Line 23. Reduction ratio $\Delta A/\Delta P$ above τ_2 implies good approximation quality, allowing increase of the trust region.

3.4. Relationship to Gradient Descent

A trust region approach can be seen as a generalization of a gradient descent approach. In this section we will revisit this relationship in the case of the specific energy that

Algorithm 2: FAST TRUST REGION

```

1  $S_0 \leftarrow S_{\text{init}}, \lambda \leftarrow \lambda_{\text{init}}, \text{convergedFlag} \leftarrow 0$ 
2 Repeat until convergedFlag
3   //Compute approximation model (2) around  $S_0$ 
4    $U_0(S) \leftarrow$  Taylor exp. of  $R(S)$  at  $S_0$  //details in [8]
5   //Solve trust region sub-problem
6    $S_\lambda \leftarrow \text{argmin}_S L_\lambda$  //Lagrangian  $L_\lambda$  in (4)
7    $\Delta P = \tilde{E}(S_0) - \tilde{E}(S_\lambda)$  //predicted reduction in energy
8    $\Delta A = E(S_0) - E(S_\lambda)$  //actual reduction in energy
9   If  $\Delta P = 0$  //((meaning  $S_\lambda = S_0$  and  $\lambda > \lambda_{\max}$ )
10     $\lambda \leftarrow \lambda_{\max}$  //make smallest possible step
11    //Solve trust region sub-problem
12     $S_\lambda \leftarrow \text{argmin}_S L_\lambda$ 
13     $\Delta P = \tilde{E}(S_0) - \tilde{E}(S_\lambda)$  //predicted reduction
14     $\Delta A = E(S_0) - E(S_\lambda)$  //actual reduction
15    //Update current solution
16     $S_0 \leftarrow \begin{cases} S_\lambda & \text{if } \Delta A > 0 \\ S_0 & \text{otherwise} \end{cases}$ 
17    convergedFlag  $\leftarrow (\Delta A \leq 0)$  //local minima
18  Else //((meaning  $S_\lambda \neq S_0$  and  $\lambda \leq \lambda_{\max}$ )
19    //Update current solution
20     $S_0 \leftarrow \begin{cases} S_\lambda & \text{if } \Delta A > 0 \\ S_0 & \text{otherwise} \end{cases}$ 
21  End
22  //Adjust the trust region
23   $\lambda \leftarrow \begin{cases} \lambda/\alpha & \text{if } \Delta A/\Delta P > \tau_2 \\ \lambda \cdot \alpha & \text{otherwise} \end{cases}$ 

```

24 we use $\alpha = 10$, $\tau_2 = 0.25$; λ_{\max} is defined in Property 2.

we use. In particular, we are interested in a relationship between our approach and a level-set approach. Like in Section 3.2 we express the energy $\tilde{E}(S)$ as a function of segmentation boundary ∂S . We denote this energy by \tilde{E}_C and it holds $\tilde{E}_C(\partial S) = \tilde{E}(S)$. Now let S_λ be the minimizer of

$$L_\lambda(S) = \tilde{E}_C(\partial S) + \frac{\lambda}{2} \text{dist}(\partial S_0, \partial S)^2. \quad (10)$$

According to the definition (5) there is a vector field V on the boundary ∂S_0 such that $\partial S_0 + V = \partial S_\lambda$. Below we denote this vector field by $\partial S_\lambda - \partial S_0$. Note that this notation implies that every parameterization of S_0 induces a parameterization of S_λ . For the minimizer S_λ it holds that

$$\begin{aligned} 0 &= \nabla \tilde{E}_C(\partial S_\lambda) + \lambda(\partial S_\lambda - \partial S_0) \\ \partial S_\lambda &= \partial S_0 - t \nabla \tilde{E}(S_\lambda), \end{aligned} \quad (11)$$

where $t = 1/\lambda$ is a step size. Note that Equation (11) is an update step that may arise during gradient descent approaches using the level-set formulation. There are differences between our approach and the level-set framework. First, we minimize (10) globally using graph-cut or TV approaches, while level-set methods only make small steps.

Another difference is that our trust region approach does not follow $-\nabla\tilde{E}(S_0)$ but rather $-\nabla\tilde{E}(S_\lambda)$ instead. We show that this is nonetheless a direction in which the energy decreases. By rewriting (11), we obtain

$$\partial S_0 = \partial S_\lambda + t\nabla\tilde{E}(S_\lambda),$$

which proves that S_0 can be seen as a *gradient ascent* step starting from S_λ if $t = 1/\lambda$ is small enough. Obviously S_λ becomes a gradient descent step for such small t .

In practice, we cannot make infinitesimally small steps because the minimal step size is given by $t = 1/\lambda_{\max}$. According to (7), $S_{\lambda_{\max}}$ is a descent step of the energy, i.e.

$$\tilde{E}(S_{\lambda_{\max}}) \leq \tilde{E}(S_0).$$

We use this property in order to simulate a gradient descent approach. Starting with segmentation S_0 , at iteration $k + 1$, we set $S_{k+1} = S_{\lambda_{\max}}$ computed with respect to segmentation S_k . Since $\tilde{E}(\cdot)$ decreases, this approach converges.

We show in Section 4 that such a *simulated gradient descent* approach is not only much slower than our trust-region approach, but also less reliable as it is prone to get stuck in a weaker local minimum.

4. Applications

In this section we apply our method to segmentation of natural and medical images. We selected several examples of segmentation energies with non-linear regional constraints. These include volume constraint, shape prior in a form of L_2 distance from target shape moments, as well as Kullback-Leibler divergence and Bhattacharyya distance between the segment and target appearance distributions.

We compare the performance of our Fast Trust Region approach with the exact line-search algorithm proposed in [8] and simulated gradient descent described in Section 3.4, because these are the most related general algorithms for minimization of non-linear regional segmentation energies. Our implementation of the above methods is based on graph-cuts, therefore we compare the energy as a function of number of graph-cuts performed. We use the floating point precision in the standard code for graph-cuts [5].

While the running time of simulated gradient descent could potentially be improved by using level-sets implementation, it would still be prone to getting stuck in weak local minimum when optimizing complex energies (see Figures 7-9). This behavior of simulated gradient descent method also conforms to the conclusions made in [2, 1] regarding gradient descent based on level-sets.

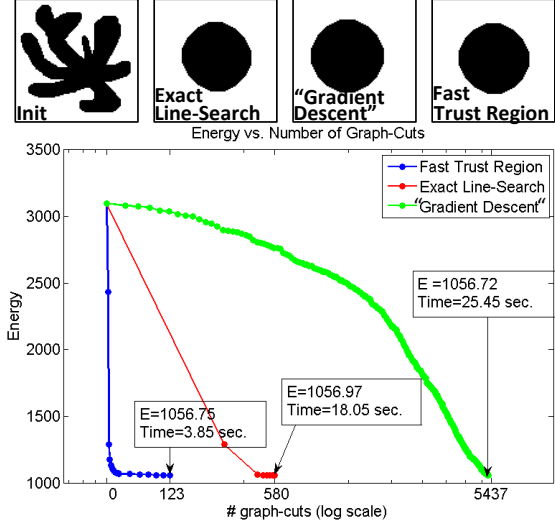


Figure 4. Synthetic example with volume constraint: $\lambda_{\text{Smooth}} = 1$, $\lambda_{\text{Shape}} = 0.0001$. Target volume is the size of initial segmentation.

4.1. Volume Constraint

Below, we perform image segmentation with volume constraint. Namely, $E(S) = R(S) + Q(S)$ where

$$R(S) = \frac{1}{2}((1_\Omega, S) - V)^2,$$

V is a given target volume and $Q(S)$ is a 16-neighborhood quadratic length term,

$$Q(S) = \lambda \sum_{(p,q)} w_{pq} \cdot \delta(s_p \neq s_q).$$

We approximate $R(S)$ near S_0 using the first order Taylor approximation $U_0(S) = \langle g, S \rangle$. For our volume constraint, this results in

$$g(x, y) \equiv \langle 1_\Omega, S_0 \rangle - V.$$

This is a relatively simple energy and Figure 4(top) shows that FTR as well as exact line-search [8] and simulated gradient descent converge to good local minimum solutions (circle), with FTR being significantly faster (bottom).

Figure 5 shows four examples of vertebrae segmentation with volume constraint. The color coded segmentations (yellow, green, red, cyan) are performed separately but shown together due to the lack of space. Since the volume varies considerably across vertebrae we use a range volume constraint that penalizes deviations from the allowable range, namely

$$R(S) = \begin{cases} 1/2(\langle 1_\Omega, S \rangle - V_{\max})^2 & \text{if } |S| \geq V_{\max} \\ 1/2(\langle 1_\Omega, S \rangle - V_{\min})^2 & \text{if } |S| \leq V_{\min} \\ 0 & \text{otherwise.} \end{cases}$$

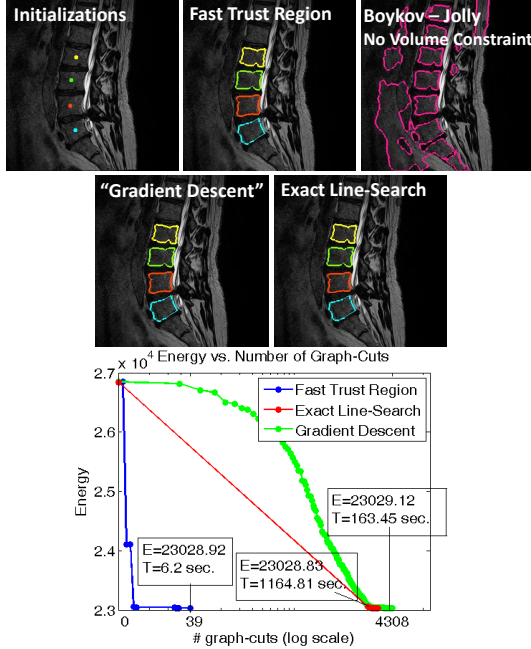


Figure 5. Four examples of vertebrae segmentation with range volume constraint, color coded (yellow, green, red, cyan). We used $V_{\min} = 890$, and $V_{\max} = 1410$, $\lambda_{\text{Smooth}} = 0.02$, $\lambda_{\text{Shape}} = 0.01$, $\lambda_{\text{App}} = 0.1$ and appearance models with 32 bins.

In this example, in addition to the volume constraint and contrast-sensitive quadratic length term we make use of Boykov-Jolly style log-likelihoods [4] based on color histograms. Namely, $E(S) = R(S) + Q(S) + D(S)$, where $D(S)$ is a standard log-likelihood unary term. In this case, $\tilde{E}(S) = U_0(S) + Q(S) + D(S)$. Again, all three methods (FTR, exact line-search and simulated gradient descent) converge to good solutions (see Figure 5) with FTR being significantly faster. The plot shows convergence behavior for the vertebrae marked in red. The volume constraint strongly controls the resulting segmentation compared to the one obtained without the constraint (top-right).

4.2. Shape Prior with Geometric Shape Moments

Below, we perform image segmentation with shape prior using L_2 distance between segment and target geometric shape moments. Our energy is defined as $E(S) = R(S) + Q(S) + D(S)$. Here, $D(S)$ is a standard log-likelihood unary term based on color histograms, $Q(S)$ is a contrast-sensitive quadratic length term and $R(S)$ is given by

$$R(S) = \frac{1}{2} \sum_{p+q \leq d} (\langle x^p y^q, S \rangle - m_{pq})^2,$$

with m_{pq} denoting the target geometric moment of order $d = p + q$. The first order Taylor approximation of $R(S)$

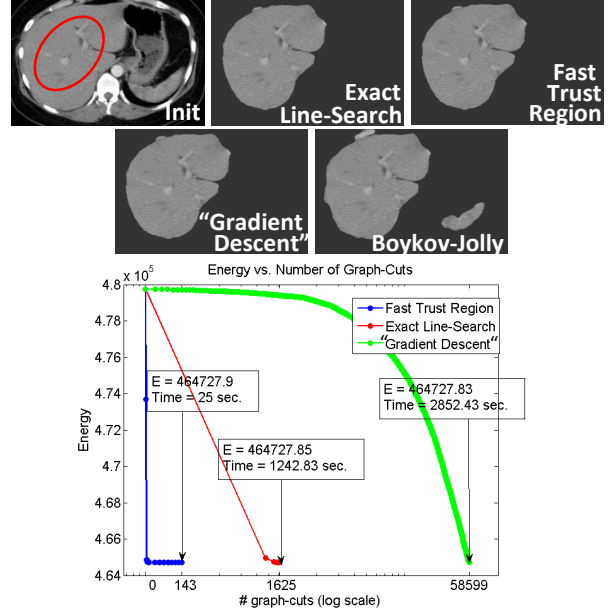


Figure 6. Liver segmentation with the shape prior. The target shape moments and appearance model are computed for the user provided input (ellipse). We used moments of up to order 2 excluding volume and appearance models with 100 bins. ($\lambda_{\text{Smooth}} = 5$, $\lambda_{\text{Shape}} = 0.01$ and $\lambda_{\text{App}} = 1$)

near S_0 results in $U_0(S) = \langle g, S \rangle$ where

$$g(x, y) = \sum_{p+q \leq d} [\langle x^p y^q, S_0 \rangle - m_{pq}] x^p y^q.$$

Figure 6 shows an example of liver segmentation with the shape prior constraint. The target shape moments as well as the foreground and background appearance models are computed from an input ellipse (top-left) provided by user as in [11]. We used moments of up to order $d = 2$ (including the center of mass and shape covariance but excluding the volume). This energy can be optimized quite well with the exact line-search and the simulated gradient descent methods, but it is 10 to 100 times faster to do so with FTR (bottom). Shape prior constraint controls the resulting segmentation compared to the best segmentation obtained without shape prior (top-right).

So far we have demonstrated that FTR is a fast optimization method. In the next experiments we show that as the segmentation energy becomes more complex, FTR becomes more advantageous since Gradient Descent often gets stuck in weak local minimum while exact line-search is too slow.

4.3. Matching Target Appearance

In the experiments below we apply FTR to optimize segmentation energies where the goal is to match a given target

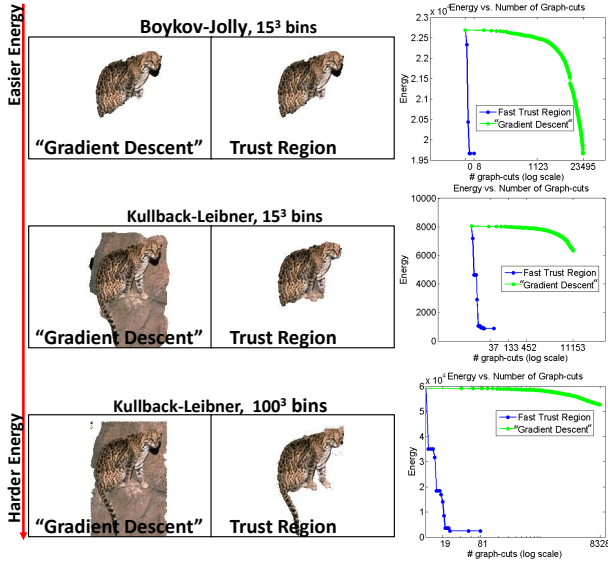


Figure 7. Matching target appearance: via standard log-likelihood data term style Boykov-Jolly [4] with 15 bins per color channel (top), KL divergence with 15 bins per channel (middle) and KL divergence with 100 bins per channel (bottom). Target appearance model is set using the ground truth, $\lambda_{\text{Smooth}} = 0.01$, $\lambda_{\text{App}} = 80$.

appearance distribution using either Kullback-Leibler divergence [8] or Bhattacharyya distance [1, 8] between the segment and target appearance distributions. The images in the experiments below are taken from [13]. We approximate $R(S)$ near S_0 using the first order Taylor approximation $U_0(S) = \langle g, S \rangle$ resulting in the following scalar functions:

$$g(x, y) = \sum_{i=1}^k \left[\log \left(\frac{\langle f_i, S_0 \rangle}{\langle 1, S_0 \rangle q_i} \right) + 1 \right] \cdot \left[\frac{f_i(x, y)}{\langle 1, S_0 \rangle} - \frac{\langle f_i, S_0 \rangle}{\langle 1, S_0 \rangle^2} \right]$$

for the KL divergence and

$$g(x, y) = \frac{\sum_{i=1}^k \sqrt{\frac{\langle f_i, S_0 \rangle q_i}{\langle 1, S_0 \rangle^3}} - \sqrt{\frac{q_i}{\langle 1, S_0 \rangle \langle f_i, S_0 \rangle}} f_i(x, y)}{2 \sum_{i=1}^k \sqrt{\frac{\langle f_i, S_0 \rangle q_i}{\langle 1, S_0 \rangle}}}$$

for the Bhattacharyya distance. Here, f_i is an indicator function of pixels belonging to bin i and q_i is the target probability of bin i . The target appearance distributions for the object and background were obtained from the ground truth segments. We used 100 bins per color channel.

Figure 7 illustrates the superior performance of FTR compared to the simulated gradient descent method. As the energy becomes more complex, either due to addition of the non-linear regional term (moving from the first row to the second) or due to the increased number of bins used to fit appearance distribution (moving from the second row to the third), the Gradient Descent approach gets stuck in weak local minimum while our FTR efficiently converges to good solutions (see right column for comparing the energy).

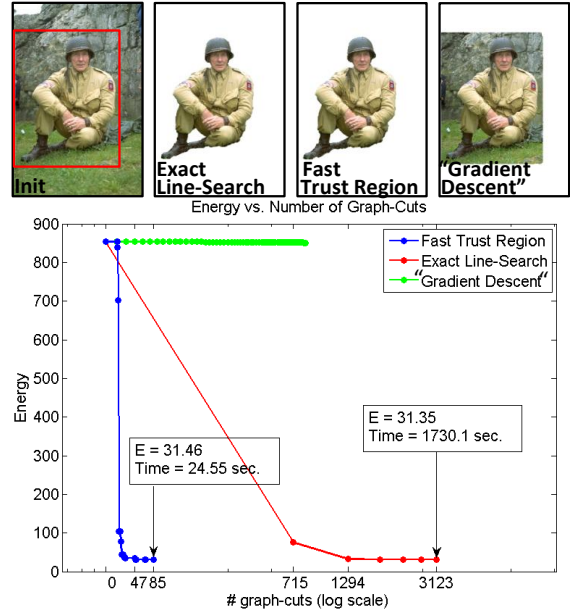


Figure 8. Matching target appearance using KL divergence and 100 bins per channel. Target appearance model is set using the ground truth segmentation. We used $\lambda_{\text{Smooth}} = 0.01$, $\lambda_{\text{App}} = 100$

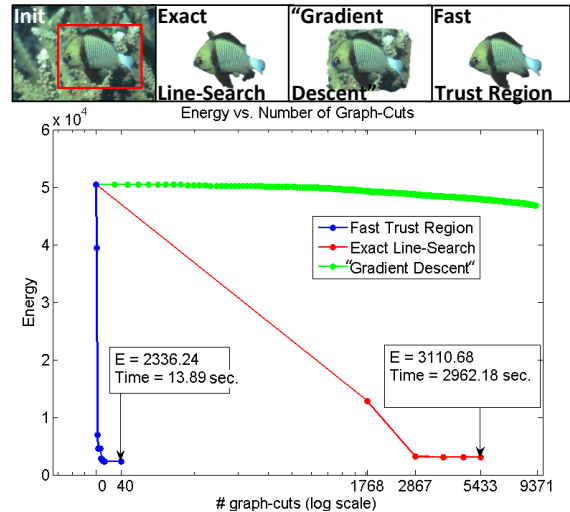


Figure 9. Matching target appearance using Bhattacharyya distance and 100 bins per channel. Target appearance model is set using the ground truth segmentation, $\lambda_{\text{Smooth}} = 1$, $\lambda_{\text{App}} = 1000$.

Figures 8-9 show additional examples with KL divergence and Bhattacharyya distance respectively, using 100 bins per color channel and regularizing with contrast sensitive quadratic length term $Q(S)$. The simulated gradient descent is unable to reduce the energy, while exact line-search is about 100 times slower than the proposed FTR.

Finally, Figure 10 show the practical robustness of the FTR algorithm to the reduction ratio threshold τ_2 .

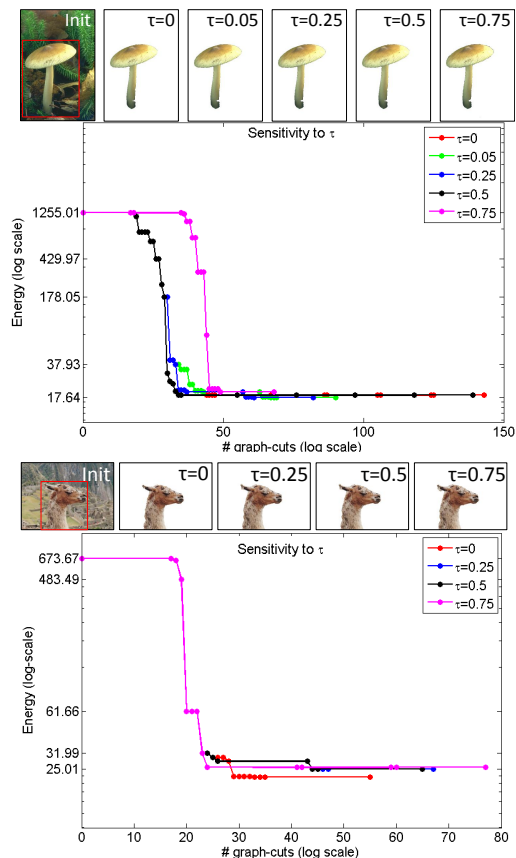


Figure 10. Robustness to reduction ratio τ_2 with KL divergence, 100 bins per channel, $\lambda_{\text{smooth}} = 0.01$ and $\lambda_{\text{App}} = 100$. Target appearance model is set using the ground truth segmentation.

5. Conclusions and Extensions

We show that the proposed Fast Trust Region (FTR) method is a robust and practically efficient algorithm for a very general class of high-order segmentation functionals. We use a Lagrangian formulation of the trust region sub-problem and derive a simple analytic relationship between step size d and Lagrange multiplier λ . This relationship allows to control the trust region size via λ . Our adaptive scheme for λ significantly speeds up (up to a factor of 100) over the *exact line-search* [8], while getting comparable solutions. We analyze the relationship between FTR and classical gradient descent approaches based on level-sets. In contrast to local linear updates in gradient decent methods, FTR incorporates long-range non-linear steps that, in practice, can avoid weak local minima.

Extensions of FTR can explore other schemes for adjusting λ between the iterations, e.g., one can use scalar $\|\nabla \tilde{E}_C(\partial S_0)\|$ in $d \sim 1/\lambda$ relation (9). Following [6], FTR can use non-Euclidean trust regions, e.g., ellipsoids as in *Levenberg-Marquardt*. Approximate models better than (2) can be used in case their efficient minimizers are known.

References

- [1] I. Ben Ayed, H. Chen, K. Punithakumar, I. Ross, and S. Li. Graph cut segmentation with a global constraint: Recovering region distribution via a bound of the Bhattacharyya measure. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2010. 1, 5, 7
- [2] I. Ben Ayed, S. Li, A. Islam, G. Garvin, and R. Chhem. Area prior constrained level set evolution for medical image segmentation. In *SPIE, Medical Imaging*, March 2008. 1, 5
- [3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 2004. 2
- [4] Y. Boykov and M.-P. Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2001. 6, 7
- [5] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(9):1124–1137, 2004. 5
- [6] Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong. An Integral Solution to Surface Evolution PDEs via Geo-Cuts. *ECCV, LNCS 3953*, 3:409–422, May 2006. 2, 3, 4, 8
- [7] D. Freedman and T. Zhang. Active contours for tracking distributions. *IEEE Transactions on Image Processing*, 13, April 2004. 1
- [8] L. Gorelick, R. Schmidt, Y. Boykov, A. Delong, and A. Ward. Segmentation with Non-Linear Regional Constraint via Line-Search cuts. In *European Conf. on Computer Vision (ECCV)*, October 2012. 1, 2, 3, 4, 5, 7, 8
- [9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 2
- [10] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Int. Conf. on Comp. Vision (ICCV)*, October 2003. 1
- [11] M. Klodt and D. Cremers. A convex framework for image segmentation with moment constraints. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011. 1, 6
- [12] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of Parametric Maxflow in Computer Vision. In *IEEE Int. Conf. on Computer Vision (ICCV)*, November 2007. 3
- [13] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. In *ACM SIGGRAPH*, 2004. 7
- [14] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *Computer Vision and Pattern Recognition (CVPR)*, June 2006. 1
- [15] T. Werner. High-arity Interactions, Polyhedral Relaxations, and Cutting Plane Algorithm for Soft Constraint Optimisation (MAP-MRF). In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2008. 1
- [16] O. J. Woodford, C. Rother, and V. Kolmogorov. A Global Perspective on MAP Inference for Low-Level Vision. In *Int. Conf. on Computer Vision (ICCV)*, October 2009. 1
- [17] Y. Yuan. A review of trust region algorithms for optimization. In *Proceedings of the Fourth International Congress on Industrial & Applied Mathematics (ICIAM)*, 1999. 2, 4