Fast Trust Region for Segmentation

Lena Gorelick [1]
lenagorelick@gmail.com

Frank R. Schmidt [2]
schmidt@cs.uni-freiburg.de

Yuri Boykov [1]
yuriboykov@gmail.com

[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 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 non-linear (more accurate) approximation model.

In this paper we propose a Fast Trust Region (FTR) approach for optimization of segmentation energies with non-linear 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 times 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 [9, 7, 13, 2, 1, 10, 8]. In image segmentation such energies are particularly useful when there is a prior knowledge regarding the appearance model or the shape of an object being segmented.

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

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

where S is a binary segmentaiton, $R(S)$ 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, 14, 15].

In this paper we propose a fast optimization algorithm for general high-order energies like (1) based on more accurate non-linear approximation models and a general *trust region* framework. 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

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

similar to those in [9, 13, 8]. Unlike [9, 13] we globally optimize this approximation model 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 backtracking line-search techniques [3] and standard trust-region optimization framework [16] we propose to minimize the approximation model within a ball of fixed radius d and adaptively adjust this radius from iteration to iteration. As in [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 in [8].

The rest of the paper is structured as follows. Section 1.1 reviews general trust region approach. Section 2 describes the details of the proposed Fast Trust Region approach. Experimental results are presented in Section 3 and finally we conclude in Section 4.

1.1. General Trust Region Approach

Trust region methods are a class of iterative optimization algorithms. In each iteration, an approximate model of the optimization problem is constructed near the current solution. The model is only "trusted" within a small region around the current solution called "trust region". This is intuitive, since approximations fit the original non-linear function only locally. The approximate model is then globally optimized within the trust region to obtain a candidate iterate solution. This step is often called *trust region sub-problem*. The size of the trust region is adjusted in each 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 step and a merit function to decide regarding the acceptance of the candidate solution and adjustment of the next trust region size. For a detailed review of trust region methods see [16].

Below we outline a general version of a trust region algorithm in the context of image segmentation (see pseudo-code in Algorithm 1). The goal is to minimize $E(S)$ in (1). Given the current solution S_k and distance d_k, the energy E is approximated using

$$\widetilde{E}(S) = U_0(S) + Q(S), \tag{2}$$

where $U_0(S)$ is the first order Taylor approximation of the non-linear term $R(S)$ near S_k. The trust region sub-problem is then solved by minimizing $\widetilde{E}$ within the region given by d_k (line 3). Namely,

$$S^* = \underset{||S-S_k||<d}{\operatorname{argmin}} \widetilde{E}(S) \tag{3}$$

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, the solution is updated in line 7 and the trust region is adjusted in line 9. It is common to set the parameter τ_1 in line 7 to zero, meaning that any candidate solution that decreases the actual energy gets accepted. The parameter τ_2 in line 9 is usually set to 0.25 [16]. Reduction ratio above this value corresponds to good approximation model allowing increase in the trust region size and thus larger step size.

Algorithm 1: GENERAL TRUST REGION APPROACH

1 **Repeat**
2 **//Solve Trust Region Sub-Problem**
3 $S^* \longleftarrow \operatorname{argmin}_{S\in\Omega,||S-S_k||\leq d_k} \widetilde{E}(S)$ (3)
4 $\Delta P = \widetilde{E}(S_k) - \widetilde{E}(S^*)$ //predicted reduction in energy
5 $\Delta A = E(S_k) - E(S^*)$ //actual reduction in energy
6 **//Update current solution**
7 $S_{k+1} \longleftarrow \begin{cases} S^* & \text{if } \Delta A/\Delta P > \tau_1 \\ S_k & \text{otherwise} \end{cases}$
8 **//Adjust the trust region**
9 $d_{k+1} \longleftarrow \begin{cases} d_k \cdot \alpha & \text{if } \Delta A/\Delta P > \tau_2 \\ d_k/\alpha & \text{otherwise} \end{cases}$
10 **Until Convergence**

2. Our approach

In the previous section, we discussed the general trust region approach. The central point of this approach is to solve a non-linear trust region sub-problem with the distance constraint (3). In Section 2.1 we show how this problem can be solved using unconstrained Lagrangian Formulation and state its properties. Then, in Section 2.2 we discuss the relationship between distance constraint d in (3) and Lagrange multiplier λ. In Section 2.3 we describe in detail our Fast Trust Region algorithm and in Section 2.4 we discuss its relation to gradient descent methods.

2.1. Lagrangian Formulation

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

$$L_\lambda(S) = \widetilde{E}(S) + \frac{\lambda}{2}\operatorname{dist}(\partial S_0, \partial S)^2, \tag{4}$$

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

$$\operatorname{dist}(\partial S_0, \partial S) := \left[\int_{S_0} \min_{s\in S} \|s - s_0\|^2 \, ds_0\right]^{\frac{1}{2}}. \tag{5}$$

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

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

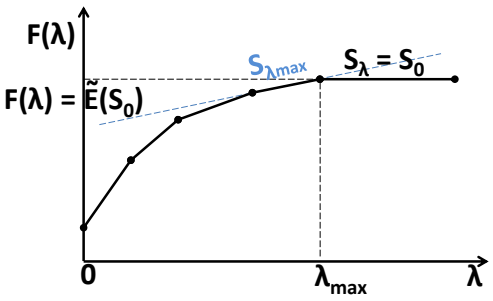

Figure 1. Each S induces a linear function $\lambda \mapsto L_\lambda(S)$. Their lower envelope yields the function $F(\lambda) = \min_S L_\lambda(S)$.

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 $F : \mathbb{R}_+ \to \mathbb{R}$ 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 [11].

Property 2: Let S_λ be the minimizer of $L_\lambda(S)$ in (4) and let $\lambda_{\max}$ 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, $F(\lambda) = \widetilde{E}(S_0) = \text{const}$. Since F is concave (Prop.1), F must also be monotonic non-decreasing function of λ with maximum at $\lambda_{\max}$ (see Figure 1).

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

$$\widetilde{E}(S_\lambda) \leq \widetilde{E}(S_0). \tag{7}$$

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

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

contradicting the optimality of S_λ. More generally,

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

2.2. 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 subproblem (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 $d = \operatorname{dist}(\partial S_0, \partial S_\lambda)$.

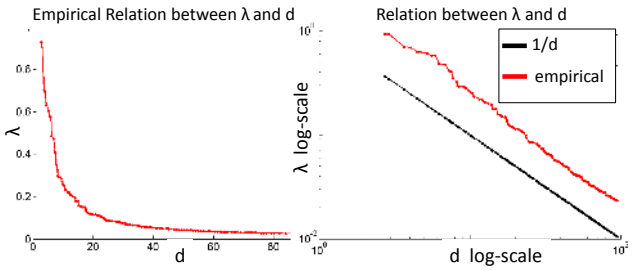

Figure 2. Empirical dependence between λ and d obtained in one typical iteration in our experiments (left). Using the log-scale in both λ and d it can be seen that the slope of empirical dependence is the same as that of $1/d$.

Figure 2(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 $\operatorname{dist}(\partial S_0, \partial S)$ is given by the approximation in (6), i.e.

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

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

$$0 = \nabla \widetilde{E}(S_\lambda) + \lambda \phi_0 \tag{9}$$

$$0 = \left\langle \nabla \widetilde{E}(S_\lambda), S_\lambda - S_0 \right\rangle + \lambda \langle \phi_0, S_\lambda - S_0 \rangle \tag{10}$$

$$\lambda = \frac{\left\langle \nabla \widetilde{E}(S_\lambda), S_0 - S_\lambda \right\rangle}{d^2} \approx \frac{\widetilde{E}(S_0) - \widetilde{E}(S_\lambda)}{d^2} \tag{11}$$

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 $S : \Omega \to \{0, 1\}$ is also a function of the form $S : \Omega \to \mathbb{R}$.

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

$$\begin{aligned} \lambda \approx& \frac{\left\langle \nabla \widetilde{E}_C(\partial S_0), \partial S_\lambda - \partial S_0 \right\rangle}{d^2} \\ \leq& \frac{\left\| \nabla \widetilde{E}_C(\partial S_0) \right\| \operatorname{dist}(\partial S_0, \partial S_\lambda)}{d^2} = \frac{\left\| \nabla \widetilde{E}_C(\partial S_0) \right\|}{d} \end{aligned}$$

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 2(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.

2.3. Fast Trust Region (FTR)

In this section we describe our Fast Trust Region (FTR) algorithm, see Algorithm 2 below. This algorithm follows from the standard trust region framework, see Algorithm 1, but uses Lagrangian formulation described in Section 2.1 instead of constrained optimization in 3. The relationship between Lagrange multiplier λ and distance d established in Section 2.2 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 4). The algorithm either decreases the energy by accepting the candidate solution (line 18) or reduces the trust region (line 21). When the trust region is so small that $S_k = S_\lambda$ (line 7), one more attempt is made using $\lambda_{\max}$ (see Property 2). If no reduction in actual energy is achieved using $S_{\lambda_{\max}}$ (line 14), we have arrived at local minimum [6] and the algorithm stops (line 15).

Following recommendations for standard trust region methods [16], we set parameter $\tau_2 = 0.25$ in line 21. Reduction ratio $\Delta A/\Delta P$ above τ_2 implies good approximation quality, allowing increase of the trust region.

Algorithm 2: FAST TRUST REGION ALGORITHM

1 $k \longleftarrow 0$, $S_k \longleftarrow S_{\text{init}}$, $\lambda \longleftarrow \lambda_{\text{init}}$, convergedFlag $\longleftarrow 0$
2 **repeat**
3 **//Solve trust region sub-problem**
4 $S_\lambda \longleftarrow \operatorname{argmin}_S L_\lambda$ (4) //minimize with curr step-size
5 $\Delta P = \widetilde{E}(S_k) - \widetilde{E}(S_\lambda)$ //predicted reduction in energy
6 $\Delta A = E(S_k) - E(S_\lambda)$ //actual reduction in energy
7 **If** $\Delta P = 0$ //(meaning $S_\lambda = S_k$ and $\lambda > \lambda_{\max}$)
8 $\lambda \longleftarrow \lambda_{\max}$ //make smallest possible step
9 **//Solve trust region sub-problem**
10 $S_\lambda \longleftarrow \operatorname{argmin} L_\lambda$
11 $\Delta P = \widetilde{E}(S_k) - \widetilde{E}(S_\lambda)$ //predicted reduction
12 $\Delta A = E(S_k) - E(S_\lambda)$ //actual reduction
13 **//Update current solution**
14 $S_{k+1} \longleftarrow \begin{cases} S_\lambda & \text{if } \Delta A > 0 \\ S_k & \text{otherwise} \end{cases}$
15 convergedFlag$\longleftarrow (S_{k+1} = S_k)$ //local minima
16 **Else** //(meaning $S_\lambda \neq S_k$ and $\lambda \leq \lambda_{\max}$)
17 **//Update current solution**
18 $S_{k+1} \longleftarrow \begin{cases} S_\lambda & \text{if } \Delta A > 0 \\ S_k & \text{otherwise} \end{cases}$
19 **End**
20 **//Adjust the trust region**
21 $\lambda \longleftarrow \begin{cases} \lambda/\alpha & \text{if } \Delta A/\Delta P > \tau_2 \\ \lambda \cdot \alpha & \text{otherwise} \end{cases}$
22 **until** convergedFlag

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

2.4. 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 we use. In particular, we are interested in a relationship between our approach and a level-set approach. Like in Section 2.2 we express the energy $\widetilde{E}(S)$ as a function of segmentation boundary ∂S. We denote this energy by $\widetilde{E}_C$ and it holds $\widetilde{E}_C(\partial S) = \widetilde{E}(S)$. Now let S_λ be the minimizer of

$$L_\lambda(S) = \widetilde{E}_C(\partial S) + \frac{\lambda}{2}\operatorname{dist}(\partial S_0, \partial S)^2. \qquad (12)$$

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 parametrization of S_0 induces a parametrization of S_λ. For the minimizer S_λ it holds that

$$\begin{aligned} 0 =& \nabla \widetilde{E}_C(\partial S_\lambda) + \lambda(\partial S_\lambda - \partial S_0) \\ \partial S_\lambda =& \partial S_0 - t\nabla \widetilde{E}(S_\lambda), \end{aligned} \qquad (13)$$

where $t = 1/\lambda$ is a step size. Note that Equation (13) 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 (12) 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 $-\nabla \widetilde{E}(S_0)$ but rather $-\nabla \widetilde{E}(S_\lambda)$ instead. We show that this is nonetheless a direction in which the energy decreases. By rewriting (13), we obtain

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

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.

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

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 $\widetilde{E}(\cdot)$ decreases, this approach converges.

We show in Section 3 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.

3. 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 2.4, 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 6-8). This behavior of simulated gradient descent method also conforms to the conclusions made in [2, 1] regarding gradient descent based on level-sets.

3.1. Volume Constraint

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

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

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 3(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 4 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

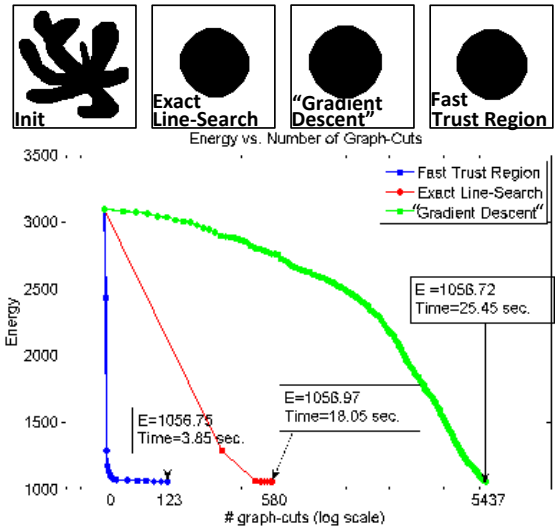

Figure 3. Synthetic example for volume constraint: $\lambda_{\text{Smooth}} = 1$, $\lambda_{\text{Shape}} = 0.0001$. Target volume is the size of initial segmentation.

volume constraint that penalizes deviations from the allowable range, namely

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

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, $\widetilde{E}(S) = U_0(S) + Q(S) + D(S)$. Again, all three methods (FTR, exact line-search and simulated gradient descent) converge to good solutions (see Figure 4) 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).

3.2. Shape Prior with Geometric Shape Moments

Below, we perform image segmentation with shape prior constraint in a form of 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 $R(S)$

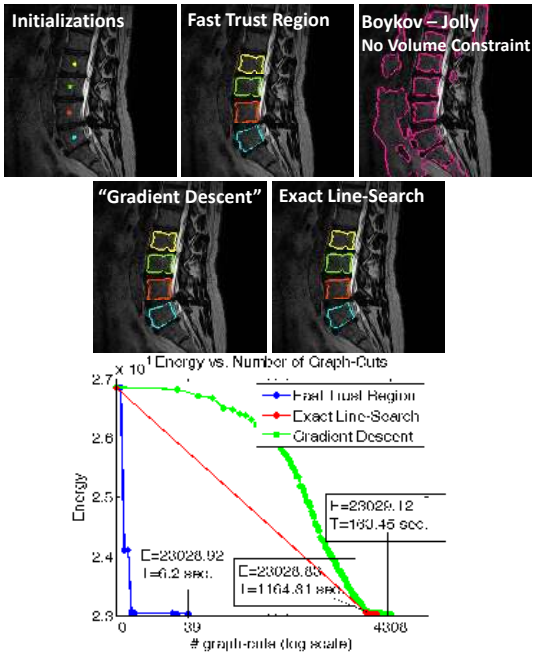

Figure 4. Four examples of vertebrae segmentation with range volume constraint, color coded (yellow, green, red, cyan). $\lambda_{\text{Smooth}} = 0.02$, $\lambda_{\text{Shape}} = 0.01, \lambda_{\text{Smooth}} = 0.1$. We used $V_{\min} = 890$ and $V_{\max} = 1410$])

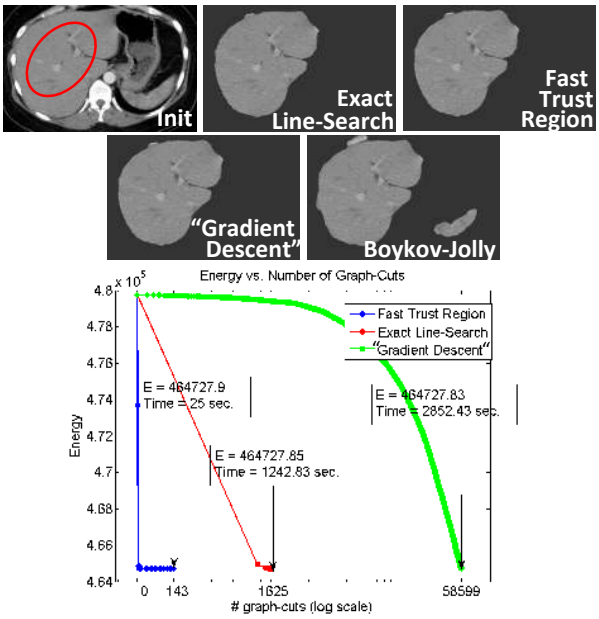

Figure 5. Liver segmentation using shape prior constraint in a form of L^2 distance from the target shape moments. As in [10] we compute geometric shape moments and appearance models for the user provided input (ellipse). We used moments of up to order 2 excluding volume and log-likelihood appearance 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

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

Figure 5 shows an example of liver segmentation with the above 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 [10]. 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.

3.3. Matching Target Appearance

In the experiments below we apply FTR to optimize segmentation energies where the goal is to match a given target appearance distribution using either Kullback-Leibler divergence [8] or Bhattacharrya distance [1, 8] between the segment and target appearance distributions. The images in the experiments below are taken from [12]. Again, we approximate $R(S)$ near S_0 using the first order Taylor approximation $U_0(S) = \langle g, S \rangle$ which results 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 the background were obtained from the ground truth segments. We used 100 bins per color channel.

Figure 6 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

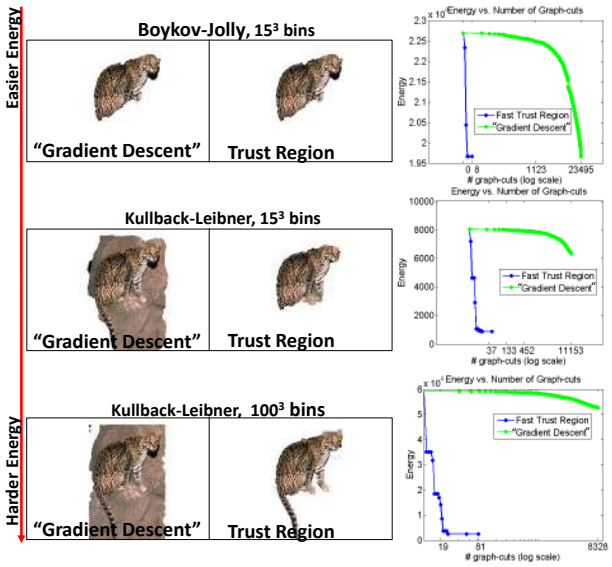

Figure 6. Matching target appearance distribution using linear log-likelihood data term style Boykov-Jolly [4] and 15 bins per color channel (top row), KL divergence with 15 bins per channel (middle row) and KL divergence with 100 bins per color channel (bottom row). $\lambda_{\text{Smooth}} = 0.01$, $\lambda_{\text{App}} = 80$. Target appearance model is set using the ground truth segmentation.

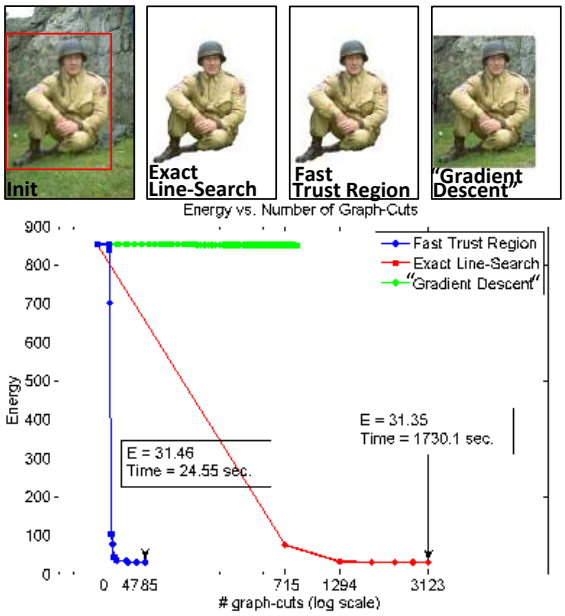

Figure 7. Matching target appearance distribution using Kullback-Leibler Divergence and 100 bins per color channel: $\lambda_{\text{Smooth}} = 0.01$, $\lambda_{\text{App}} = 100$. Target appearance model is set using the ground truth segmentation.

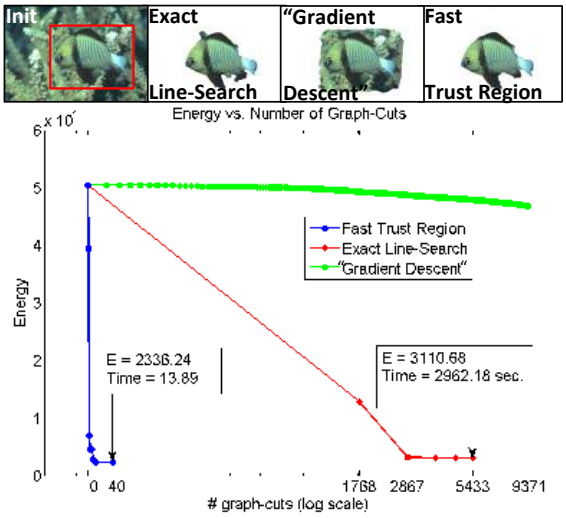

Figure 8. Matching target appearance distribution using Bhattacharyya distance and 100 bins per color channel. We used $\lambda_{\text{Smooth}} = 1$, $\lambda_{\text{App}} = 1000$. Target appearance model is set using the ground truth segmentation.

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).

Figures 7-8 show additional examples with KL divergence and Bhattacharrya distance respectively, using 100 bins per color channel and regularizing with contrast sensitive quadratic length term $Q(S)$. Note, the simulated gradient descent method is unable to reduce the energy, while exact line-search is about 100 times slower than the proposed FTR.

Finally, Figures 9-10 show the practical robustness of the FTR algorithm to the reduction ratio threshold τ_2.

4. Conclusions

In this paper we propose a Fast Trust Region (FTR) algorithm for image segmentation. We use the Lagrangian framework for the trust-region sub-problem and derive a simple analytic relationship between distance constraint d and Lagrange multiplier λ. This relationship allows us to adjust the trust region distance constraint implicitly by controlling the Lagrange multiplier λ. We demonstrate that our adaptive scheme for λ significantly speeds up (up to a factor of 100) over the exact line-search [8], while getting comparable solutions in practice. Moreover, 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, avoid weak local minima.

We demonstrate that FTR is an efficient and robust opti-

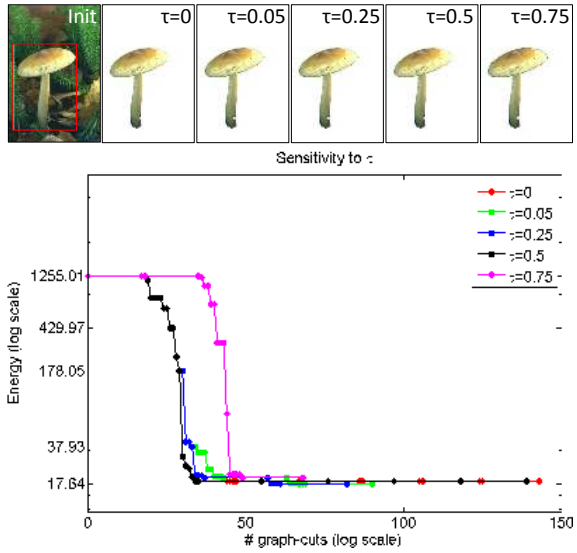

Figure 9. Robustness to reduction ratio τ_2 when matching target appearance distribution using Kullback-Leibler Divergence and 100 bins per color channel. We used $\lambda_{\text{Smooth}} = 0.01$, $\lambda_{\text{App}} = 100$. Target appearance model is set using the ground truth segmentation using 100 bins per color channel.

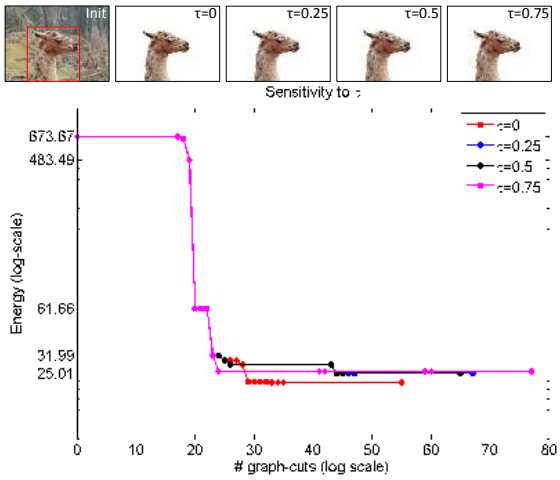

Figure 10. Robustness to reduction ratio τ_2 when matching target appearance distribution using Kullback-Leibler divergence and 100 bins per color channel. We used $\lambda_{\text{Smooth}} = 0.01$, $\lambda_{\text{App}} = 100$. Target appearance model is set using the ground truth segmentation using 100 bins per color channer.

mization method. In the future we plan to explore possible applications of FTR for optimization of novel complex energies.

References

[1] I. Ayed, H. Chen, K. Punithakumar, I. Ross, and L. Shuo. Graph cut segmentation with a global constraint: Recovering region distribution via a bound of the Bhattacharyya measure. In *Computer Vision and Pattern Recognition (CVPR)*, June 2010. 1, 5, 6

[2] I. Ben Ayed, A. Li, S. ; Islam, G. Garvin, and C. R. Area prior constrained level set evolution for medical image segmentation. In *Proc. SPIE, Medical Imaging 2008: Image Processing*, 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. 5, 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

[7] D. Freedman and T. Zhang. Active contours for tracking distributions. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 13, April 2004. 1

[8] L. Gorelick, F. 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, 5, 6, 7

[9] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *IEEE Int. Conf. on Computer Vision (ICCV)*, October 2003. 1

[10] 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

[11] 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

[12] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. In *ACM SIGGRAPH*, 2004. 6

[13] 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

[14] 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

[15] O. J. Woodford, C. Rother, and V. Kolmogorov. A Global Perspective on MAP Inference for Low-Level Vision. In *IEEE Int. Conf. on Computer Vision (ICCV)*, October 2009. 1

[16] Y. Yuan. A review of trust region algorithms for optimization. In *Proceedings of the Fourth Int. Congress on Industrial & Applied Mathematics (ICIAM)*, 99. 2, 4