Efficiently Solving the Fractional Trust Region Problem

Anders P. Eriksson, Carl Olsson and Fredrik Kahl

Centre for Mathematical Sciences
Lund University, Sweden

Abstract. Normalized Cuts has successfully been applied to a wide range of tasks in computer vision, it is indisputably one of the most popular segmentation algorithms in use today. A number of extensions to this approach have also been proposed, ones that can deal with multiple classes or that can incorporate a priori information in the form of grouping constraints. It was recently shown how a general linearly constrained Normalized Cut problem can be solved. This was done by proving that strong duality holds for the Lagrangian relaxation of such problems. This provides a principled way to perform multi-class partitioning while enforcing any linear constraints exactly.
The Lagrangian relaxation requires the maximization of the algebraically smallest eigenvalue over a one-dimensional matrix sub-space. This is an unconstrained, piece-wise differentiable and concave problem. In this paper we show how to solve this optimization efficiently even for very large-scale problems. The method has been tested on real data with convincing results. [1]

1 Introduction

Image segmentation can be defined as the task of partitioning an image into disjoint sets. This visual grouping process is typically based on low-level cues such as intensity, homogeneity or image contours. Existing approaches include thresholding techniques, edge based methods and region-based methods. Extensions to this process includes the incorporation of grouping constraints into the segmentation process. For instance the class labels for certain pixels might be supplied beforehand, through user interaction or some completely automated process, [1, 2].

Perhaps the most successful and popular approaches for segmenting images are based on graph cuts. Here the images are converted into undirected graphs with edge weights between the pixels corresponding to some measure of similarity. The ambition is that partitioning such a graph will preserve some of the spatial structure of the image itself. These graph based methods were made popular first through the Normalized Cut formulation of [3] and more recently by the energy minimization method of [4]. This algorithm for optimizing objective functions that are submodular has the property of solving many discrete problems exactly. However, not all segmentation problems can

[1] This work has been supported by the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMErobot^{TM}, Swedish Foundation for Strategic Research (SSF) through the programmes Vision in Cognitive Systems II (VISCOS II) and Spatial Statistics and Image Analysis for Environment and Medicine.

be formulated with submodular objective functions, nor is it possible to incorporate all types of linear constraints.

In [5] it was shown how linear grouping constraints can be included in the former approach, Normalized Cuts. It was demonstrated how Lagrangian relaxation can in a unified can handle such linear constrains and also in what way they influence the resulting segmentation. It did not however address the practical issues of finding such solutions. In this paper we develop efficient algorithms for solving the Lagrangian relaxation.

2 Background.

2.1 Normalized Cuts.

Consider an undirected graph **G**, with nodes **V** and edges **E** and where the non-negative weights of each such edge is represented by an affinity matrix W, with only non-negative entries and of full rank. A min-cut is the non-trivial subset A of V such that the sum of edges between nodes in A and V is minimized, that is the minimizer of

$$cut(A,V) = \sum_{i \in A,\ j \in V \setminus A} w_{ij} \tag{1}$$

This is perhaps the most commonly used method for splitting graphs and is a well known problem for which very efficient solvers exist. It has however been observed that this criterion has a tendency to produced unbalanced cuts, smaller partitions are preferred to larger ones.

In an attempt to remedy this shortcoming, Normalized Cuts was introduced by [3]. It is basically an altered criterion for partitioning graphs, applied to the problem of perceptual grouping in computer vision. By introducing a normalizing term into the cut metric the bias towards undersized cuts is avoided. The Normalized Cut of a graph is defined as:

$$N_{cut} = \frac{cut(A,V)}{assoc(A,V)} + \frac{cut(B,V)}{assoc(B,V)} \tag{2}$$

where $A \cup B = V$, $A \cap B = \emptyset$ and the normalizing term defined as $assoc(A,V) = \sum_{i \in A j \in V} w_{ij}$ It is then shown in [3] that by relaxing (2) a continuous underestimator of the Normalized Cut can be efficiently computed.

To be able to include general linear constraints we reformulated the problem in the following way, (see [5] for details). With $d = W1$ and $D = \text{diag}(d)$ Normalized Cut cost can be written as

$$\inf_z \frac{z^T(D-W)z}{-z^T dd^T z + (1^T d)^2}, \text{ s.t. } z \in \{-1,1\}^n,\ Cz = b. \tag{3}$$

The above problem is a non-convex, NP-hard optimization problem. In [5] $z \in \{-1,1\}^n$ constraint was replaced with the norm constraint $z^T z = n$. This gives us the relaxed problem

$$\inf_z \frac{z^T(D-W)z}{-z^T dd^T z + (1^T d)^2}, \text{ s.t. } z^T z = n,\ Cz = b. \tag{4}$$

Even though this is a non-convex problem it was shown in [5] that it is possible to solve this problem exactly.

2.2 The Fractional Trust Region Subproblem

Next we briefly review the theory for solving (4). If we let $\hat{z}$ be the extended vector $\begin{bmatrix} z^T & z_{n+1} \end{bmatrix}^T$. Throughout the paper we will write $\hat{z}$ when we consider the extended variables and just z when we consider the original ones. With $\hat{C} = [C \quad -b]$ the linear constraints becomes $Cz = b$, and now form a linear subspace and can be eliminated in the following way. Let $N_{\hat{C}}$ be a matrix where its columns form a base of the nullspace of $\hat{C}$. Any $\hat{z}$ fulfilling $\hat{C}\hat{z} = 0$ can be written $\hat{z} = N_{\hat{C}}\hat{y}$, where $\hat{y} \in \mathbb{R}^{k+1}$. Assuming that the linear constraints are feasible we may always choose that basis so that $\hat{y}_{k+1} = \hat{z}_{n+1}$. Let $L_{\hat{C}} = {N_{\hat{C}}}^T \left[\begin{smallmatrix} (D-W) & 0 \\ 0 & 0 \end{smallmatrix}\right] N_{\hat{C}}$ and $M_{\hat{C}} = {N_{\hat{C}}}^T \left[\begin{smallmatrix} ((1^T d)D - dd^T) & 0 \\ 0 & 0 \end{smallmatrix}\right] N_{\hat{C}}$, both positive semidefinite, (see [5]). In the new space we get the following formulation

$$\inf_{\hat{y}} \frac{\hat{y}^T L_{\hat{C}} \hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}}, \text{ s.t. } \hat{y}_{k+1} = 1, \; ||\hat{y}||^2_{N_{\hat{C}}} = n + 1, \tag{5}$$

where $||\hat{y}||^2_{N_{\hat{C}}} = \hat{y}^T {N_{\hat{C}}}^T N_{\hat{C}} \hat{y}$. We call this problem the fractional trust region subproblem since if the denominator is removed it is similar to the standard trust region problem [6]. A common approach to solving problems of this type is to simply drop one of the two constraints. This may however result in very poor solutions. For example, in [7] segmentation with prior data was studied. The objective function considered there contained a linear part (the data part) and a quadratic smoothing term. It was observed that when $y_{k+1} \neq \pm 1$ the balance between that smoothing term and the data term was disrupted, resulting in very poor segmentations.

In [5] it was show that in fact this problem can be solved exactly, without excluding any constraints, by considering the dual problem.

Theorem 1. *If a minima of* (5) *exists its dual problem*

$$\sup_t \inf_{||\hat{y}||^2_{N_{\hat{C}}} = n+1} \frac{\hat{y}^T (L_{\hat{C}} + tE_{\hat{C}}) \hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}} \tag{6}$$

$$\text{where } E_{\hat{C}} = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right] - \frac{N^T_{\hat{C}} N_{\hat{C}}}{n+1} = N^T_{\hat{C}} \begin{bmatrix} -\frac{1}{n+1} I & 0 \\ 0 & 1 \end{bmatrix} N_{\hat{C}},$$

has no duality gap.

Since we assume that the problem is feasible and as the objective function of the primal problem is the quotient of two positive semidefinite quadratic forms a minima obviously exists. Thus we can apply this theorem directly and solve (5) through its dual formulation. We will use $F(t, \hat{y})$ to denote the objective function of (6), the Lagrangian of problem (5). By the dual function $\theta(t)$ we mean the solution of $\theta(t) = \inf_{||\hat{y}||^2_{N_{\hat{C}}} = n+1} F(t, \hat{y})$

The inner minimization of (6) is the well known generalized Rayleigh quotient, for which the minima is given by the algebraically smallest generalized eigenvalue [2] of

[2] By generalized eigenvalue of two matrices A and B we mean finding a $\lambda = \lambda^G(A, B)$ and v, $||v|| = 1$ such that $Av = \lambda Bv$ has a solution.

$(L_{\hat{C}} + tE_{\hat{C}})$ and $M_{\hat{C}}$. Letting $\lambda_{\min}(t)(\cdot,\cdot)$ denote the smallest generalized eigenvalue of two entering matrices, we can also write problem (6) as

$$\sup_t \lambda_{\min}(L_{\hat{C}} + tE_{\hat{C}}, M_{\hat{C}}). \tag{7}$$

These two dual formulations will from here on be used interchangeably, it should be clear from the context which one is being referred to. In this paper we will develop methods for solving the outer maximization efficiently.

3 Efficient Optimization

3.1 Subgradient Optimization

First we present a method, similar to that used in [8] for minimizing binary problems with quadratic objective functions, based on subgradients for solving the dual formulation of our relaxed problem. We start off by noting that as $\theta(t)$ is a pointwise infimum of functions linear in t it is easy to see that this is a concave function. Hence the outer optimization of (6) is a concave maximization problem, as is expected from dual problems. Thus a solution to the dual problem can be found by maximizing a concave function in one variable t. Note that the choice of norm does not affect the value of θ it only affects the minimizer $\hat{y}^*$.

It is widely known that the eigenvalues are analytic (and thereby differentiable) functions as long as they are distinct. Thus, to be able to use a steepest ascent method we need to consider subgradients. Recall the definition of a subgradient [9, 8].

Definition 1 *If a function $g : \mathbb{R}^{k+1} \mapsto \mathbb{R}$ is concave, then $v \in \mathbb{R}^{k+1}$ is a subgradient to g at σ_0 if*

$$g(\sigma) \leq g(\sigma_0) + v^T(\sigma - \sigma_0), \quad \forall \sigma \in \mathbb{R}^{k+1}. \tag{8}$$

One can show that if a function is differentiable then the derivative is the only vector satisfying (8). We will denote the set of all subgradients of g at a point t_0 by $\partial g(t_0)$. It is easy to see that this set is convex and if $0 \in \partial g(t_0)$ then t_0 is a global maximum. Next we show how to calculate the subgradients of our problem.

Lemma 1. *If $\hat{y}_0$ fulfills $F(\hat{y}_0, t_0) = \theta(t_0)$ and $||\hat{y}_0||^2_{N_{\hat{C}}} = n + 1$. Then*

$$v = \frac{\hat{y}_0^T E_{\hat{C}} \hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0} \tag{9}$$

is a subgradient of θ at t_0. If θ is differentiable at t_0, then v is the derivative of θ at t_0.

Proof.

$$\theta(t) = \min_{||\hat{y}||^2_{N_{\hat{C}}}=1} \frac{\hat{y}^T(L_{\hat{C}} + tE_{\hat{C}})\hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}} \leq \frac{\hat{y}_0^T(L_{\hat{C}} + tE_{\hat{C}})\hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0} =$$

$$= \frac{\hat{y}_0^T(L_{\hat{C}} + t_0E_{\hat{C}})\hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0} + \frac{\hat{y}_0^T E_{\hat{C}} \hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0}(t - t_0) = \theta(t_0) + v^T(t - t_0) \tag{10}$$

A Subgradient Algorithm Next we present an algorithm based on the theory of subgradients. The idea is to find a simple approximation of the objective function. Since the function θ is concave, the first order Taylor expansion $\theta_i(t)$, around a point t_i, always fulfills $f_i(t) \leq f(t)$. If $\hat{y}_i$ solves $\inf_{||\hat{y}||^2_{N_{\hat{C}}}=n+1} F(\hat{y}, t_i)$ and this solution is unique then the Taylor expansion of θ at t_i is

$$\theta_i(t) = F(\hat{y}_i, t_i) + v^T(t - t_i). \tag{11}$$

Note that if $\hat{y}_i$ is not unique f_i is still an overestimating function since v is a subgradient.

One can assume that the function θ_i approximates θ well in a neighborhood around $t = t_i$ if the smallest eigenvalue is distinct. If it is not we can expect that there is some t_j such that $\min(\theta_i(t), \theta_j(t))$ is a good approximation. Thus we will construct a function $\bar{\theta}$ of the type

$$\bar{\theta}(t) = \inf_{i \in I} F(\hat{y}_i, t_i) + v^T(t - t_i) \tag{12}$$

that approximates θ well. That is, we approximate θ with the point-wise infimum of several first-order Taylor expansions, computed at a number of different values of t, an illustration can be seen in fig. 1. We then take the solution to the problem $\sup_t \bar{\theta}(t)$, given by

$$\begin{array}{l} \sup_{t,\alpha} \alpha \\ \alpha \leq F(\hat{y}_i, t_i) + v^T(t - t_i), \ \forall i \in I, \ t_{min} \leq t \leq t_{max}. \end{array} \tag{13}$$

as an approximate solution to the original dual problem. Here, the fixed parameters t_{min}, t_{max} are used to express the interval for which the approximation is believed to be valid. Let t_{i+1} denote the optimizer of (13). It is reasonable to assume that $\bar{\theta}$ approximates θ better the more Taylor approximations we use in the linear program. Thus, we can improve $\bar{\theta}$ by computing the first-order Taylor expansion around t_{i+1}, add it to (13) and solve the linear program again. This is repeated until $|t_{N+1} - t_N| < \epsilon$ for some predefined $\epsilon > 0$, and t_{N+1} will be a solution to $\sup_t \theta(t)$.

3.2 A Second Order Method

The algorithm presented in the previous section uses first order derivatives only. We would however like to employ higher order methods to increase efficiency. This requires calculating second order derivatives of (6). Most formulas for calculating the second derivatives of eigenvalues involves all of the eigenvectors and eigenvalues. However, determining the entire eigensystem is not feasible for large scale systems. We will show that it is possible to determine the second derivative of an eigenvalue function by solving a certain linear system only involving the corresponding eigenvalue and eigenvector.

The generalized eigenvalues and eigenvectors fulfills the following equations

$$((L_{\hat{C}} + tE_{\hat{C}}) - \lambda(t)M_{\hat{C}})\hat{y}(t) = 0 \tag{14}$$

$$||\hat{y}(t)||^2_{N_{\hat{C}}} = n + 1. \tag{15}$$

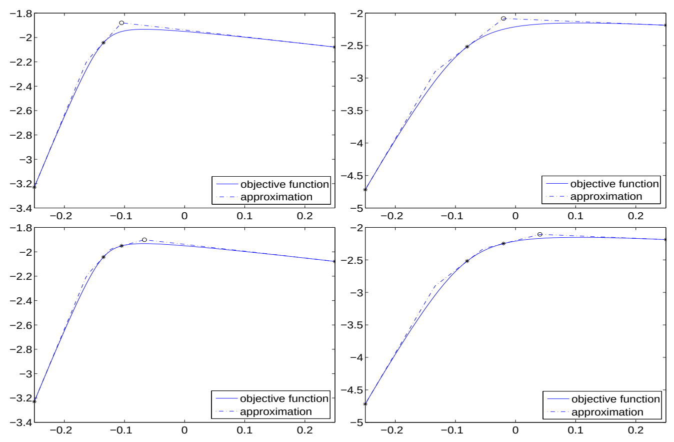

Fig. 1. Approximations of two randomly generated objective functions. Top: Approximation after 1 step of the algorithm. Bottom: Approximation after 2 steps of the algorithm.

To emphasize the dependence on t we write $\lambda(t)$ for the eigenvalue and $\hat{y}(t)$ for the eigenvector. By differentiating (14) we obtain

$$(E_{\hat{C}} - \lambda'(t)M_{\hat{C}})\hat{y}(t) + ((L_{\hat{C}} + tE_{\hat{C}}) - \lambda(t)M)\hat{y}'(t) = 0. \tag{16}$$

This $(k+1) \times (k+1)$ linear system in $\hat{y}'(t)$ will have a rank of k, assuming $\lambda(k)$ is a distinct eigenvalue. To determine $\hat{y}'(t)$ uniquely we differentiate (15), obtaining

$$\hat{y}^T(t) N_{\hat{C}}{}^T N_{\hat{C}} \hat{y}'(t) = 0. \tag{17}$$

Thus, the derivative of the eigenvector $\hat{y}'(t)$ is determined by the solution to the linear system

$$\begin{bmatrix} (L_{\hat{C}}+tE_{\hat{C}})-\lambda(t)M_{\hat{C}} \\ \hat{y}^T(t)N_{\hat{C}}{}^T N_{\hat{C}} \end{bmatrix} \hat{y}'(t) = \begin{bmatrix} (-E_{\hat{C}}+\lambda'(t)M_{\hat{C}})\hat{y}(t) \\ 0 \end{bmatrix} \tag{18}$$

If we assume differentiability at t, the second derivative of $\theta(t)$ can now be found by computing $\frac{d}{dt}\theta'(t)$, where $\theta'(t)$ is equal to the subgradient v given by (9).

$$\theta''(t) = \frac{d}{dt}\theta'(t) = \frac{d}{dt}\frac{\hat{y}(t)^T E_{\hat{C}}\hat{y}(t)}{\hat{y}(t)^T M_{\hat{C}}\hat{y}(t)} = \frac{2}{\hat{y}(t)^T M_{\hat{C}}\hat{y}(t)}\hat{y}^T(t)\left(E_{\hat{C}} - \theta'(t)M_{\hat{C}}\right)\hat{y}'(t) \tag{19}$$

A Modified Newton Algorithm Next we modify the algorithm presented in the previous section to incorporate the second derivatives. Note that the second order Taylor expansion is not necessarily an over-estimator of θ. Therefore we can not use the the second derivatives as we did in the previous section.

Instead, as we know θ to be infinitely differentiable when the smallest eigenvalue $\lambda(t)$ is distinct, strictly convex around its optima t^*, Newton's method for unconstrained optimization can be applied. It follows from these properties of $\theta(t)$ that Newton's method [9] should be well behaved on this function and that we could expect quadratic convergence in a neighborhood of t^*. All of this, under the assumption that θ is differentiable in this neighborhood. Since Newton's method does not guarantee convergence we have modified the method slightly, adding some safeguarding measures.

At a given iteration of the Newton method we have evaluated $\theta(t)$ at a number of points t_i. As θ is concave we can easily find upper and lower bounds on t^* ($t_{\min}$, $t_{\max}$) by looking at the derivative of the objective function for these values of $t = t_i$.

$$t_{\max} = \min_{i;\theta'(t_i)\leq 0} t_i, \text{ and } t_{\min} = \max_{i;\theta'(t_i)\geq 0} t_i \tag{20}$$

At each step in the Newton method a new iterate is found by approximating the objective function is by its second-order Taylor approximation

$$\theta(t) \approx \theta(t_i) + \theta'(t_i)(t - t_i) + \frac{\theta''(t_i)}{2}(t - t_i)^2. \tag{21}$$

and finding its maxima. By differentiating (21) it is easily shown that its optima, as well as the next point in the Newton sequence, is given by

$$t_{i+1} = -\frac{\theta'(t_i)}{\theta''(t_i)} + t_i \tag{22}$$

If t_{i+1} is not in the interval $[t_{\min}, t_{\max}]$ then the second order expansion can not be a good approximation of θ, here the safeguarding comes in. In these cases we simply fall back to the first-order method of the previous section. If we successively store the values of $\theta(t_i)$, as well as the computed subgradients at these points, this can be carried out with little extra computational effort. Then, the upper and lower bounds $t_{\min}$ and $t_{\max}$ are updated, i is incremented by 1 and the whole procedure is repeated, until convergence.

If the smallest eigenvalue $\lambda(t_i)$ at an iteration is not distinct, then $\theta''(t)$ is not defined and a new Newton step can not be computed. In these cases we also use the subgradient gradient method to determine the subsequent iterate. However, empirical studies indicate that non-distinct smallest eigenvalues are extremely unlikely to occur.

4 Experiments

A number of experiments were conducted in an attempt to evaluate the suggested approaches. As we are mainly interested in maximizing a concave, piece-wise differentiable function, the underlying problem is actually somewhat irrelevant. However, in order to emphasize the intended practical application of the proposed methods, we ran the subgradient- and modified Newton algorithms on both smaller, synthetic problems as well as on larger, real-world data. For comparison purposes we also include the results of a golden section method [9], used in [5], as a baseline algorithm.

First, we evaluated the performance of the proposed methods on a large number of synthetic problems. These were created by randomly choosing symmetric, positive definite, 100×100 matrices. As the computational burden lies in determining the generalized eigenvalue of the matrices $L_{\hat{C}} + tE_{\hat{C}}$ and $M_{\hat{C}}$ we wish to reduce the number of such calculations. Figure 2 shows a histogram of the number of eigenvalue evaluations for the subgradient-, modified Newton method as well as the baseline golden section search.

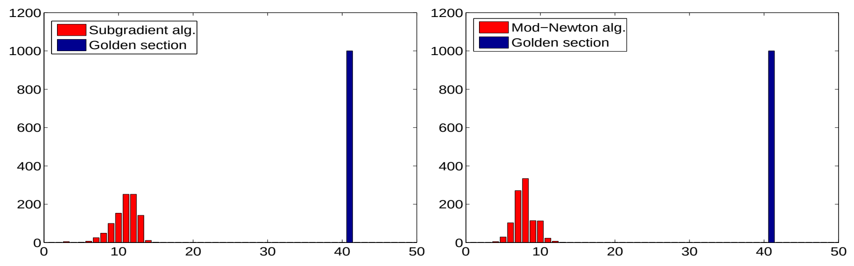

Fig. 2. Histogram of the number of function evaluations required for 1000-synthetically generated experiments using a golden section method (blue) and the subgradient algorithm (red).

The two gradient methods clearly outperforms the golden section search. The difference between the subgradient- and modified Newton is not as discernible. The somewhat surprisingly good performance of the subgradient method can be explained by the fact that far away from t^* the function $\theta(t)$ is practically linear and an optimization method using second derivatives would not have much advantage over one that uses only first order information.

Finally, we applied our methods to two real world examples. The underlying motivation for investigating an optimization problem of this form was to segment images with linear constraints using Normalized Cuts. The first image can be seen in fig. 3, the linear constraints included were hard constraints, that is the requirement that that certain pixels should belong to the foreground or background. One can imagine that such constraints are supplied either by user interaction in a semi-supervised fashion or by some automatic preprocessing of the image. The image was gray-scale, approximately 100×100 pixels in size, the associated graph was constructed based on edge information as described in [10]. The second image was of traffic intersection where one wishes to segment out the small car in the top corner. We have a probability map of the image, giving the likelihood of a certain pixel belonging to the foreground. Here the graph representation is based on this map instead of the gray-level values in the image. The approximate size and location of the vehicle is know and included as linear constraint into the segmentation process. The resulting partition can be seen in fig. 4.

In both these real world cases, the resulting segmentation will always be the same, regardless of approach. What is different is the computational complexity of the different methods. Once again, the two gradient based approaches are much more efficient than a golden section search, and their respective performance comparable. As the methods differ in what is required to compute, a direct comparison of them is not a straight forward procedure. Comparing the run time would be pointless as the degree to which the implementations of the individual methods have been optimized for

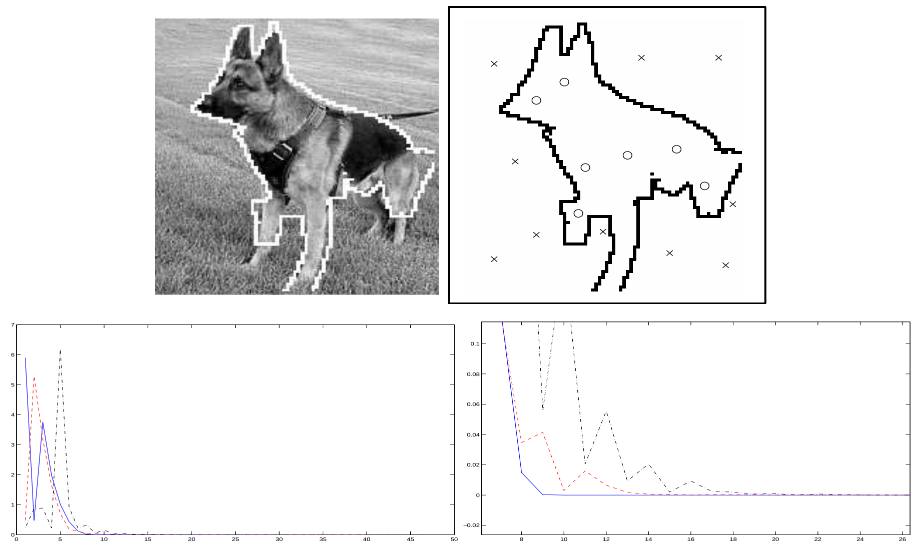

Fig. 3. Top: Resulting segmentation (left) and constraints applied (right). Here an X means that this pixel belongs to the foreground and an O to the background. Bottom: Convergence of the modified Newton (solid), subgradient (dashed) and the golden section (dash-dotted) algorithms. The algorithms converged after 9, 14 and 23 iteration respectively.

speed differ greatly. However, as it is the eigenvalue computations that are the most demanding we believe that comparing the number of such eigenvalue calculations will be a good indicator of the computational requirements for the different approaches. It can be seen in fig. 3 and 4 how the subgradient methods converges quickly in the initial iterations only to slow down as it approaches the optima. This is in support of the above discussion regarding the linear appearance of the function $\theta(t)$ far away from the optima. We therfore expect the modified Newton method to be superior when higher accuracy is required.

In conclusion we have proposed two methods for efficiently optimizing a piece-wise differentiable function using both first- and second order information applied to the task of partitioning images. Even though it is difficult to provide a completely accurate comparison between the suggested approaches it is obvious that the Newton based method is superior.

References

1. Rother, C., Kolmogorov, V., Blake, A.: ”GrabCut”: interactive foreground extraction using iterated graph cuts. In: ACM Transactions on Graphics. (2004) 309–314
2. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: International Conference on Computer Vision, Vancouver, Canada (2001) 05–112

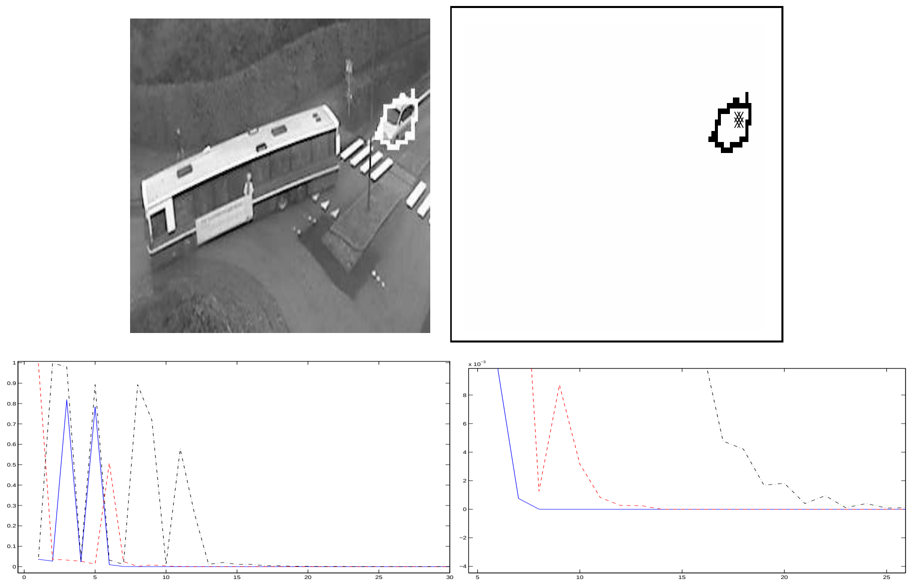

Fig. 4. Top: Resulting segmentation (left) and constraints applied, in addition to the area requirement used (area = 50 pixels) (right). Here the X in the top right part of the corner means that this pixel belongs to the foreground. Bottom: Convergence of the modified Newton (solid), subgradient (dashed) and the golden section (dash-dotted) algorithms. The algorithms converged after 9, 15 and 23 iteration respectively.

3. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Analysis and Machine Intelligence **22**(8) (2000) 888–905
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Analysis and Machine Intelligence **23**(11) (2001) 1222–1239
5. Eriksson, A., Olsson, C., Kahl, F.: Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. In: International Conference on Computer Vision, Rio de Janeiro, Brazil (2007)
6. Sorensen, D.: Newton's method with a model trust region modification. SIAM Journal on Nummerical Analysis **19**(2) (1982) 409–426
7. Eriksson, A., Olsson, C., Kahl, F.: Image segmentation with context. In: Proc. Conf. Scandinavian Conference on Image Analysis, Ahlborg, Denmark (2007)
8. Olsson, C., Eriksson, A., Kahl, F.: Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. In: Proc. Conf. Computer Vision and Pattern Recognition, Mineapolis, USA (2007)
9. Bazaraa, Sherali, Shetty.: Nonlinear Programming, Theory and Algorithms. Wiley (2006)
10. Malik, J., Belongie, S., Leung, T.K., Shi, J.: Contour and texture analysis for image segmentation. International Journal of Computer Vision **43**(1) (2001) 7–27