Normalized Cuts Revisited: A Reformulation for Segmentation with Linear Grouping Constraints

Anders Eriksson
University of Adelaide, Australia.
anders.eriksson@adelaide.edu.au

Carl Olsson
Centre for Mathematical Sciences, Lund University, Sweden.
calle@maths.lth.se

Fredrik Kahl
Centre for Mathematical Sciences, Lund University, Sweden.
fredrik@maths.lth.se

***Abstract*— Indisputably Normalized Cuts is one of the most popular segmentation algorithms in pattern recognition and computer vision. It has been applied to a wide range of segmentation tasks with great success. A number of extensions to this approach have also been proposed, including ones that can deal with multiple classes or that can incorporate a priori information in the form of grouping constraints. However, what is common for all these methods is that they are noticeably limited in the type of constraints that can be incorporated and can only address segmentation problems on a very specific form. In this paper, we present a reformulation of Normalized Cut segmentation that in a unified way can handle linear equality constraints for an arbitrary number of classes. This is done by restating the problem and showing how linear constraints can be enforced exactly in the optimization scheme through duality. This allows us to add group priors, for example, that certain pixels should belong to a given class. In addition, it provides a principled way to perform multi-class segmentation for tasks like interactive segmentation. The method has been tested on real data showing good performance and improvements compared to standard normalized cuts.**

I. Image Segmentation

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, see [5], [10], [12], [18], [20], [22]. 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 [8], [18].

Currently 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 methods were made popular first through the Normalized Cut formulation of [20] and more recently by the energy minimization method of [6]. The algorithm in [6] for optimizing objective functions that are submodular has the advantage of solving many discrete problems exactly. However, not all segmentation problems can be formulated with submodular objective functions, nor is it possible to incorporate linear (or affine) equality constraints.

The work described here concerns the former approach, Normalized Cuts, the relevance of linear grouping constraints and how they can be included in this framework. A similar extension to include linear constraints for submodular objective functions was recently presented in [23].

Incorporating general linear constraints into the Normalized cut formulation was also attempted by [24]. In this work it was shown that by making additional assumptions about the segmentation the porblem can be further relaxed to a simpler, globally solvable minimization problem. This will however not solve the problem exactly and only return a lower bound solution to the Normalized cut relaxation.

It is not the aim of this paper to argue the merits of one segmentation method, or one cut metric, over another, nor do we here concern ourselves with how the actual grouping constraints are obtained. Instead we will focus on the optimization problem and show through Lagrangian relaxation and duality how one can, in a unified manner, handle such linear equality constraints *exactly* and also in what way these constraints influence the resulting segmentation.

In addition to the extension of normalized cuts, a key contribution of this paper is the development of an efficient algorithm for minimizing objective functions consisting of a ratio of quadratic functions subject to linear equality constraints. Similar objective functions have appeared in many other computer vision applications, for example, [3], [19]. Our framework has the potential to improve computational efficiency, in particular, for large-scale problems.

A. Problem Formulation

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 its complement is minimized, that is, the minimizer of

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

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

In an attempt to remedy this shortcoming, Normalized Cuts was introduced in [20]. 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 is defined as $assoc(A,V) = \sum_{i\in A, j\in V} w_{ij}$. It is then shown in [20] that by relaxing (2) a continuous underestimator of the (minimal) Normalized Cut can be efficiently computed. These techniques are then extended in [25] beyond graph bipartitioning to include multiple segments, and even further in [26] to handle certain types of linear equality constraints.

One can argue that the drawbacks of the original formulation for computing the Normalized Cut are that firstly, obtaining a discrete solution from the relaxed one can be problematic. Especially in multiclass segmentation where the relaxed solution is not unique but consists of an entire subspace. Then, the set of grouping constraints is restricted. Only homogeneous linear equality constraints can be directly included in the existing theory, which is of limited practical use. We will show that this excludes many visually relevant constraints. In [7] an attempt is made at solving a similar problem with general linear constraints. This approach does however involve dropping any discrete constraint all together, leaving one to question the quality or tightness of the obtained underestimator.

II. Normalized Cuts with Grouping Constraints

In this section we propose a reformulation of the relaxation of Normalized Cuts that in a unified way can handle all types of linear equality constraints for any number of partitions. First we show how we through duality theory reach the suggested relaxation. The following two sections then show why this formulation is well suited for dealing with general linear constraints and how this proposed approach can be applied to multiclass segmentation.

Starting off with the definition of Normalized Cuts in (2), the cost of partitioning an image with affinity matrix W into two disjoint sets, A and B, can be written as

$$N_{cut} = \frac{\sum_{\substack{i\in A\\ j\in B}} w_{ij}}{\sum_{\substack{i\in A\\ j\in V}} w_{ij}} + \frac{\sum_{\substack{i\in B\\ j\in A}} w_{ij}}{\sum_{\substack{i\in B\\ j\in V}} w_{ij}}. \tag{3}$$

Let $z \in \{-1,1\}^n$ be the class label vector, W the $n\times n$-matrix with entries w_{ij}, d the $n\times 1$-vector containing the row sums of W, and D the diagonal $n\times n$-matrix with d on the diagonal. A 1 is used to denote vectors of all ones. We can write (3) as

$$\begin{aligned} N_{cut} &= \frac{\sum_{i,j} w_{ij}(z_i - z_j)^2}{2\sum_i (1+z_i)d_i} + \frac{\sum_{i,j} w_{ij}(z_i - z_j)^2}{2\sum_i (1-z_i)d_i} = \\ &= \frac{z^T(D-W)z}{d^T(1+z)} + \frac{z^T(D-W)z}{d^T(1-z)} = \\ &= \frac{2d^T 1\big(z^T(D-W)z\big)}{1^T dd^T 1 - z^T d^T d^T z} = \frac{2d^T 1\big(z^T(D-W)z\big)}{z^T((1^T d)D - dd^T)z}. \end{aligned} \tag{4}$$

In the last inequality we used the fact that $1^T d = z^T Dz$. When we include general linear constraints on z on the form $Cz = b$, $C \in \mathbb{R}^{m\times n}$, the optimization problem associated with this partitioning cost becomes

$$\begin{aligned} \inf_z \quad & \frac{z^T(D-W)z}{z^T((1^T d)D - dd^T)z} \\ \text{s.t.} \quad & z \in \{-1,1\}^n \\ & Cz = b. \end{aligned} \tag{5}$$

The above problem is a non-convex, NP-hard optimization problem. Therefore we are led to replace the discrete $z \in \{-1,1\}^n$ constraint with the norm constraint $z^T z = n$. This gives us the relaxed problem

$$\begin{aligned} \inf_z \quad & \frac{z^T(D-W)z}{z^T((1^T d)D - dd^T)z} \\ \text{s.t.} \quad & z^T z = n \\ & Cz = b. \end{aligned} \tag{6}$$

This is also a non-convex problem. However, as we shall see in section III, we are able to solve this problem exactly. Next we will write problem (6) in homogenized form. The reason for doing this will become clear later on. Let L and M be the $(n+1)\times(n+1)$ matrices

$$L = \begin{bmatrix} (D-W) & 0 \\ 0 & 0 \end{bmatrix}, \quad M = \begin{bmatrix} ((1^T d)D - dd^T) & 0 \\ 0 & 0 \end{bmatrix}, \tag{7}$$

and

$$\hat{C} = [C \quad -b] \tag{8}$$

the homogenized constraint matrix. The relaxed problem (6) can now be written

$$\begin{aligned} \inf_z \quad & \frac{[z^T\ 1]L\left[\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right]}{[z^T\ 1]M\left[\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right]} \\ \text{s.t.} \quad & z^T z = n \\ & \hat{C}\left[\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right] = 0. \end{aligned} \tag{9}$$

Finally we add the artificial variable z_{n+1}. 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 variables. The relaxed problem (6) in its homogenized form is

$$\begin{aligned} \inf_{\hat{z}} \quad & \frac{\hat{z}^T L \hat{z}}{\hat{z}^T M \hat{z}} \\ \text{s.t.} \quad & \hat{z}_{n+1}^2 - 1 = 0 \\ & \hat{z}^T \hat{z} = n+1 \\ & \hat{C}\hat{z} = 0. \end{aligned} \tag{10}$$

Note that the first constraint is equivalent to $\hat{z}_{n+1} = 1$. If $\hat{z}_{n+1} = -1$ then we may change the sign of $\hat{z}$ to obtain a solution to our original problem.

The homogenized constraints $\hat{C}\hat{z} = 0$ 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 $\hat{C}$. Let

$k+1$ be the dimension of the nullspace. Any $\hat{z}$ fulfilling $\hat{C}\hat{z}=0$ can be written $\hat{z}=N_{\hat{C}}\hat{y}$, where $\hat{y}\in\mathbb{R}^{k+1}$. As in the case with the z-variables, $\hat{y}$ is the vector containing all variables whereas y is a vector containing all but the last variable. Assuming that the linear constraints are feasible we may always choose a basis such that $\hat{y}_{k+1}=\hat{z}_{n+1}=1$. We set

$$L_{\hat{C}}=N_{\hat{C}}^T L N_{\hat{C}} \text{ and } M_{\hat{C}}=N_{\hat{C}}^T M N_{\hat{C}}. \tag{11}$$

In the new variables, the following formulation is obtained.

$$\begin{aligned} \inf_{\hat{y}} \quad & \frac{\hat{y}^T L_{\hat{C}}\hat{y}}{\hat{y}^T M_{\hat{C}}\hat{y}} \\ \text{s.t.} \quad & \hat{y}_{k+1}^2-1=0 \\ & \hat{y}^T N_{\hat{C}}^T N_{\hat{C}}\hat{y}=||\hat{y}||^2_{N_{\hat{C}}}=n+1. \end{aligned} \tag{12}$$

We will use $f(\hat{y})$ to denote the objective function of this problem. A common approach for solving this kind of problem is to simply drop one of the two constraints. This may however result in very poor solutions. We shall see that we can in fact solve this problem exactly without excluding any constraints.

III. Lagrangian Relaxation and Strong Duality

In this section we will show how to solve (6) using Lagrangian duality. We start by generalizing a lemma from [17] for trust region problems.

Lemma 1: Let $y^TA_2y+2b_2^Ty+c_2$ be a positive semidefinite quadratic form. If there exists a y with $y^TA_3y+2b_3^Ty+c_3<0$, then, the primal problem

$$\inf_y \frac{y^TA_1y+2b_1^Ty+c_1}{y^TA_2y+2b_2^Ty+c_2},\ \text{s.t.}\ y^TA_3y+2b_3^Ty+c_3\le 0 \tag{13}$$

and the dual problem

$$\sup_{\lambda\ge 0}\inf_y \frac{y^T(A_1+\lambda A_3)y+(b_1+\lambda b_3)^Ty+c_1+\lambda c_3}{y^TA_2y+2b_2^Ty+c_2} \tag{14}$$

has no duality gap.

Proof: Since $y^TA_2y+2b_2^Ty+c_2\ge 0$, the primal problem can be written as

$$\begin{aligned} \inf \quad & \gamma_1 \\ \text{s.t.} \quad & y^T(A_1-\gamma_1A_2)y+2(b_1-\gamma_1b_2)^Ty+c_1-\gamma_1c_2\le 0 \\ & y^TA_3y+2b_3^Ty+c_3\le 0. \end{aligned} \tag{15}$$

Let $\mathcal{M}(\lambda,\gamma)$ be the matrix

$$\mathcal{M}(\lambda,\gamma)=\begin{bmatrix} A_1+\lambda A_3-\gamma A_2 & b_1+\lambda b_3-\gamma b_2 \\ (b_1+\lambda b_3-\gamma b_2)^T & c_1+\lambda c_3-\gamma c_2 \end{bmatrix}. \tag{16}$$

The dual problem can be written

$$\begin{aligned} \sup_{\lambda\ge 0}\inf_{\gamma_2,y} \quad & \gamma_2 \\ \text{s.t.} \quad & \begin{bmatrix} y \\ 1 \end{bmatrix}^T \mathcal{M}(\lambda,\gamma_2)\begin{bmatrix} y \\ 1 \end{bmatrix}\le 0. \end{aligned} \tag{17}$$

Since (17) is dual to (15) we have that for their optimal values, $\gamma_2^*\le\gamma_1^*$ must hold. To prove that there is no duality gap we must show that $\gamma_2^*=\gamma_1^*$. We do this by considering the following problem,

$$\begin{aligned} \sup_{\gamma_3,\lambda\ge 0} \quad & \gamma_3 \\ \text{s.t.} \quad & \mathcal{M}(\lambda,\gamma_3)\succeq 0. \end{aligned} \tag{18}$$

Here $\mathcal{M}(\lambda,\gamma_3)\succeq 0$ means that $\mathcal{M}(\lambda,\gamma_3)$ is positive semidefinite. We note that if $\mathcal{M}(\lambda,\gamma_3)\succeq 0$ then there is no y fulfilling

$$\begin{bmatrix} y \\ 1 \end{bmatrix}^T \mathcal{M}(\lambda,\gamma_3)\begin{bmatrix} y \\ 1 \end{bmatrix}+\epsilon\le 0 \tag{19}$$

for any $\epsilon>0$. Therefore we must have that the optimal values fulfill $\gamma_3^*\le\gamma_2^*\le\gamma_1^*$. To complete the proof we show that $\gamma_3^*=\gamma_1^*$. We note that for any $\gamma\le\gamma_1^*$ we have that

$$\begin{aligned} & y^TA_3y+2b_3^Ty+c_3\le 0\ \Rightarrow \\ & y^T(A_1-\gamma A_2)y+2(b_1-\gamma b_2)^Ty+c_1-\gamma c_2\ge 0. \end{aligned} \tag{20}$$

However, according to the S-procedure [4], this is true if and only if there exists $\lambda\ge 0$ such that $\mathcal{M}(\lambda,\gamma)\succeq 0$. Therefore (γ,λ) is feasible for problem (18) and thus $\gamma_3^*=\gamma_1^*$. ∎

We note that for a fixed γ the problem

$$\begin{aligned} \inf_y \quad & y^T(A_1-\gamma A_2)y+2(b_1-\gamma b_2)^Ty+c_1-\gamma c_2 \\ \text{s.t.} \quad & y^TA_3y+2b_3^Ty+c_3\le 0 \end{aligned} \tag{21}$$

only has an interior solution if $A_1-\gamma A_2$ is positive semidefinite. If A_3 is positive semidefinite then we may subtract $k(y^TA_3y+2b_3^Ty+c_3)$ for any $(k>0)$ from the objective function to obtain boundary solutions. This gives us the following corollary.

Corollary 1: Let $y^TA_2y+2b_2^Ty+c_2$ be a positive semidefinite quadratic form, and A_3 be positive semidefinite. If there exists a y with $y^TA_3y+2b_3^Ty+c_3<0$, then the primal problem

$$\inf_y \frac{y^TA_1y+2b_1^Ty+c_1}{y^TA_2y+2b_2^Ty+c_2},\ \text{s.t.}\ y^TA_3y+2b_3^Ty+c_3=0 \tag{22}$$

and the dual problem

$$\sup_\lambda\inf_y \frac{y^T(A_1+\lambda A_3)y+(b_1+\lambda b_3)^Ty+c_1+\lambda c_3}{y^TA_2y+2b_2^Ty+c_2} \tag{23}$$

has no duality gap.

Next we will show how to solve a problem on a form related to (12). Let

$$\hat{A}_1=\begin{bmatrix} A_1 & b_1 \\ b_1^T & c_1 \end{bmatrix},\ \hat{A}_2=\begin{bmatrix} A_2 & b_2 \\ b_2^T & c_2 \end{bmatrix},\ \hat{A}_3=\begin{bmatrix} A_3 & b_3 \\ b_3^T & c_3 \end{bmatrix}.$$

Theorem 1: If $\hat{A}_2$ and $\hat{A}_3$ are positive semidefinite, then the primal problem

$$\begin{aligned} \inf_{y^TA_3y+2b_3^Ty+c_3=n+1} & \frac{y^TA_1y+2b_1^Ty+c_1}{y^TA_2y+2b_2^Ty+c_2}= \\ & = \inf_{\substack{\hat{y}^T\hat{A}_3\hat{y}=n+1 \\ y_{n+1}^2=1}} \frac{\hat{y}^T\hat{A}_1\hat{y}}{\hat{y}^T\hat{A}_2\hat{y}} \end{aligned} \tag{24}$$

and its dual

$$\sup_t \inf_{\hat{y}^T\hat{A}_3\hat{y}=n+1} \frac{\hat{y}^T\hat{A}_1\hat{y}+ty_{n+1}^2-t}{\hat{y}^T\hat{A}_2\hat{y}} \tag{25}$$

has no duality gap.

Proof: Let γ^* be the optimal value of problem (12). Then

$$\begin{aligned} \gamma^* &= \inf_{\substack{\hat{y}^T\hat{A}_3\hat{y}=n+1 \\ y_{n+1}^2=1}} \frac{\hat{y}^T\hat{A}_1\hat{y}}{\hat{y}^T\hat{A}_2\hat{y}} \\ &= \sup_t \inf_{\substack{\hat{y}^T\hat{A}_3\hat{y}=n+1 \\ y_{n+1}^2=1}} \frac{\hat{y}^T\hat{A}_1\hat{y}+ty_{n+1}^2-t}{\hat{y}^T\hat{A}_2\hat{y}} \\ &\ge \sup_t \inf_{\hat{y}^T\hat{A}_3\hat{y}=n+1} \frac{\hat{y}^T\hat{A}_1\hat{y}+ty_{n+1}^2-t}{\hat{y}^T\hat{A}_2\hat{y}} \\ &\ge \sup_{t,\lambda} \inf_{\hat{y}} \frac{\hat{y}^T\hat{A}_1\hat{y}+ty_{n+1}^2-t+\lambda(\hat{y}^T\hat{A}_3\hat{y}-(n+1))}{\hat{y}^T\hat{A}_2\hat{y}} \\ &= \sup_{s,\lambda} \inf_{\hat{y}} \\ &\quad \frac{\hat{y}^T\hat{A}_1\hat{y}+sy_{n+1}^2-s+\lambda(y^TA_3y+y_{n+1}2b_3^Ty+c_3-(n+1))}{\hat{y}^T\hat{A}_2\hat{y}}= \\ &= \sup_\lambda \inf_{y_{n+1}^2=1} \frac{\hat{y}^T\hat{A}_1\hat{y}+\lambda(y^TA_3y+2b_3^Ty+c_3-(n+1))}{\hat{y}^T\hat{A}_2\hat{y}} \end{aligned}$$

$$= \sup_\lambda \inf_y \frac{y^T A_1 y + 2b_1^T y + c_1 + \lambda(y^T A_3 y + 2b_3^T y + c_3 - (n+1))}{y^T A_2 y + 2b_2^T y + c_2}$$
$$= \gamma^*, \quad (26)$$

where we let $s = t + c_3\lambda$. In the last two equalities, Corollary 1 was used twice. The third row of the above proof gives us that

$$\mu^* = \sup_t \inf_{\hat{y}^T \hat{A}_3 \hat{y} = n+1} \frac{\hat{y}^T \hat{A}_1 \hat{y} + t y_{n+1}^2 - t}{\hat{y}^T \hat{A}_2 \hat{y}} =$$
$$= \sup_t \inf_{\hat{y}^T \hat{A}_3 \hat{y} = n+1} \frac{\hat{y}^T \hat{A}_1 \hat{y} + t y_{n+1}^2 - t\frac{\hat{y}^T \hat{A}_3 \hat{y}}{n+1}}{\hat{y}^T \hat{A}_2 \hat{y}} =$$
$$= \sup_t \inf_{\hat{y}^T \hat{A}_3 \hat{y} = n+1} \frac{\hat{y}^T \left(\hat{A}_1 + t\left(\left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right] - \frac{\hat{A}_3}{n+1} \right) \right) \hat{y}}{\hat{y}^T \hat{A}_2 \hat{y}}. \quad (27)$$

■

Finally, since strong duality holds, we can state the following corollary.

Corollary 2: If t^* and $\hat{y}^*$ solves (25), then $(\hat{y}^*)^T \hat{N} \hat{y}^* = n+1$ and $y_{k+1}^* = 1$. That is, $\hat{y}^*$ is an optimal feasible solution to (24).

IV. The Dual Problem and Constrained Normalized Cuts

Returning to our relaxed problem (12) we start off by introducing the following lemma.

Lemma 2: L and M as defined in (7) are both $(n+1)\times(n+1)$ positive semidefinite matrices of rank $n-1$. Their 2-dimensional nullspaces are spanned by $n_1 = [\,1 \ldots 1\ 0\,]^T$ and $n_2 = [\,0 \ldots 0\ 1\,]^T$. Consequently, $L_{\hat{C}}$ and $M_{\hat{C}}$ as defined in (11) are also positive semidefinite.

Proof: L is the zero-padded positive semidefinite Laplacian matrix of the affinity matrix W and is hence also positive semidefinite. For M it suffices to show that the matrix $(1^T d)D - dd^T$ is positive semidefinite,

$$v^T((1^T d)D - dd^T)v = \textstyle\sum_i d_i \sum_j d_j v_j^2 - \left(\sum_i d_i v_i\right)^2$$
$$= \textstyle\sum_{i,j} d_i d_j v_j (v_j - v_i) = \sum_i d_i d_i v_i (v_i - v_i) +$$
$$+ \textstyle\sum_{i,j<i} d_i d_j v_j (v_j - v_i) + d_j d_i v_i (v_i - v_j) =$$
$$\textstyle\sum_{i,j<i} d_i d_j (v_j - v_i)^2 \geq 0, \ \forall v \in \mathbb{R}^n. \quad (28)$$

The last inequality comes from $d_i > 0$ for all i which means that $(1^T d)D - dd^T$, and thus also M, are positive semidefinite.

The second statement follows since both $Ln_i = Mn_i = 0$ for $i = 1, 2$.

Next, since

$$v^T L v \geq 0, \ \forall v \in \mathbb{R}^n \Rightarrow v^T L v \geq 0, \ \forall v \in Null(\hat{C}) \Rightarrow$$
$$\Rightarrow w^T {N_{\hat{C}}}^T L N_{\hat{C}}{}^T w \geq 0, \ \forall w \in \mathbb{R}^k \Rightarrow$$
$$\Rightarrow w^T L_{\hat{C}} w \geq 0, \ w \in \mathbb{R}^k,$$

it holds that $L_{\hat{C}} \succeq 0$, and similarly for $M_{\hat{C}}$. ■

Assuming that the original problem is feasible then we have that, as $f(\hat{y})$ of problem (24) is the quotient of two positive semidefinite quadratic forms and therefore $f(\hat{y})$ is non-negative, a minimum for the relaxed Normalized Cut problem will exist. Theorem 1 states that strong duality holds for a program on the form (24). Consequently, we can apply the theory from the previous section directly and solve (12) through its dual formulation. Let

$$E_{\hat{C}} = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right] - \frac{N_{\hat{C}}^T N_{\hat{C}}}{n+1} = N_{\hat{C}}^T \left[\begin{smallmatrix} -\frac{I}{n+1} & 0 \\ 0 & 1 \end{smallmatrix}\right] N_{\hat{C}} \quad (29)$$

and let $\theta(\hat{y}, t)$ denote the Lagrangian function. The dual problem is then

$$\sup_t \inf_{||\hat{y}||^2_{N_{\hat{C}}} = n+1} \theta(\hat{y}, t) = \frac{\hat{y}^T (L_{\hat{C}} + tE_{\hat{C}})\hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}}. \quad (30)$$

The inner minimization is the well known generalized Rayleigh quotient, for which the minimum is given by the algebraically smallest generalized eigenvalue[1] of $(L_{\hat{C}} + tE_{\hat{C}})$ and $M_{\hat{C}}$. Letting $\lambda_{\min}^G(t)$ and $v_{\min}^G(t)$ denote the smallest generalized eigenvalue and corresponding generalized eigenvector of $(L_{\hat{C}} + tE_{\hat{C}})$ and $M_{\hat{C}}$, we can write problem (30) as

$$\sup_t \ \lambda_{\min}^G(L_{\hat{C}} + tE_{\hat{C}}, M_{\hat{C}}). \quad (31)$$

It can easily be shown that the minimizer of the inner problem of (30), is given by a scaling of the generalized eigenvector, $\hat{y}(t) = (||v_{\min}^G(t)||_{N_{\hat{C}}}) v_{\min}^G(t)$. The relaxed Normalized Cut problem can thus be solved by finding the maximum of (31). As the objective function is the point-wise infimum of functions linear in t, it is a concave function, as is expected from dual problems. So solving (31) means maximizing a concave function in one variable t, this can be carried out using standard methods for one-dimensional optimization.

Unfortunately, the task of solving large scale generalized eigenvalue problems can be demanding, especially when the matrices involved are dense, as the case is here. This can however be remedied. By exploiting the unique matrix structure we can rewrite the generalized eigenvalue problem as a standard one. First we note that the generalized eigenvalue problem $Av = \lambda Bv$ is equivalent to the standard eigenvalue problem $B^{-1}Av = \lambda v$, if B is non-singular. Furthermore, in large scale applications it is reasonable to assume that the number of variables $n+1$ is much greater than the number of constraints m. Then the base for the null space of the homogenized linear constraints $N_{\hat{C}}$ can then be written on the form $N_{\hat{C}} = \left[\begin{smallmatrix} c & c_0 \\ I & \end{smallmatrix}\right]$. Now we can write

$$M_{\hat{C}} = \left[\begin{smallmatrix} c & c_0 \\ I & \end{smallmatrix}\right]^T \left(\left[\begin{smallmatrix} ((1^T d)D - dd^T) & 0 \\ 0 & 0 \end{smallmatrix}\right] \right) \left[\begin{smallmatrix} c & c_0 \\ I & \end{smallmatrix}\right] =$$
$$= \left\{ \begin{matrix} D := \left[\begin{smallmatrix} D_1 & 0 \\ 0 & D_2 \end{smallmatrix}\right] \\ d := \left[\begin{smallmatrix} d_1 \\ d_2 \end{smallmatrix}\right] \end{matrix} \right\} = \underbrace{\left[\begin{smallmatrix} D_2 & 0 \\ 0 & c_0^T D_1 c_0 + 1 \end{smallmatrix}\right]}_{\tilde{D}} +$$
$$+ \underbrace{\left[\begin{smallmatrix} c^T & cd_1 + d_2 & 0 \\ c_0^T & c_0^T d_1 & 1 \end{smallmatrix}\right]}_{V} \underbrace{\left[\begin{smallmatrix} D_1 & & \\ & 1 & \\ & & -1 \end{smallmatrix}\right]}_{S} \left[\begin{smallmatrix} c & & c_0 \\ d_1^T c^T + d_2^T & & d_1^T c_0 \\ 0 & & 1 \end{smallmatrix}\right] =$$
$$= \tilde{D} + VSV^T. \quad (32)$$

Hence, $M_{\hat{C}}$ is the sum of a positive definite, diagonal matrix $\tilde{D}$ and a low-rank correction VSV^T. As a direct result of the Woodbury matrix identity [11] we can express the inverse of $M_{\hat{C}}$ as

$${M_{\hat{C}}}^{-1} = (\tilde{D} + VSV^T)^{-1} =$$
$$= \tilde{D}^{-1} \left(I - V(S^{-1} + V^T \tilde{D}^{-1} V)^{-1} V \tilde{D}^{-1} \right). \quad (33)$$

Despite the potentially immense size of the entering matrices, this inverse can be efficiently computed since $\tilde{D}$ is diagonal and the size of the square matrices S and $(S^{-1} + V^T \tilde{D}^{-1} V)$ are both typically manageable and therefore easily inverted. Our

[1] A generalized eigenvalue of two matrices A and B is a scalar $\lambda = \lambda^G(A, B)$ such that for a vector v with $||v|| = 1$, the equation $Av = \lambda Bv$ has a solution.

generalized eigenvalue problem then turns into the problem of finding the smallest algebraic eigenvalue of the matrix $M_{\hat{C}}{}^{-1}L_{\hat{C}}$. The dual problem becomes

$$\sup_t \quad \lambda_{\min}\big((\tilde{D}^{-1}(I - V(S^{-1} + V^T \quad \tilde{D}^{-1}V)^{-1}V\tilde{D}^{-1}) \\ N_{\hat{C}}{}^T(L_{\hat{C}} + tE_{\hat{C}})N_{\hat{C}}\big). \tag{34}$$

Not only does this reformulation provide us with the more familiar, standard eigenvalue problem but it will also allow for very efficient computations of multiplications of vectors to this matrix. This is a crucial property, since, even though $M_{\hat{C}}{}^{-1}(L_{\hat{C}} + tE_{\hat{C}})$ is still dense, it is the product and sum of diagonal ($\tilde{D}^{-1}$, $E_{\hat{C}}$), sparse ($L_{\hat{C}}$, $N_{\hat{C}}$) and low rank matrices (V, S^{-1}). It is a very structured matrix to which iterative eigensolvers can successfully be applied. We will return to this in section VI-C.

In certain cases it might however occur that the quadratic form in the denominator is only positive semidefinite and thus singular. These cases are easily detected and must be treated separately. As we then can not invert $M_{\hat{C}}$ and rewrite the problem as a standard eigenvalue problem we must instead work with generalized eigenvalues, as defined in (31). This is preferably avoided as this is typically a more computationally demanding formulation, especially since the entering matrices are dense. Iterative methods for finding generalized methods for structured matrices such as $L_{\hat{C}} + tE$ and $M_{\hat{C}}$, do however exist [21]. Note that the absence of linear constraints is such a special instance. However, in that case homogenization is completely unnecessary. Problem (6) with $Cz = b$ removed is an standard unconstrained generalized Rayleigh quotient and the solution is given by the generalized eigenvalue $\lambda_G^T(D - W, (1^T d)D - dd^T)$.

Now, if t^* and $\hat{y}^* = (||v_{\min}^G(t^*)||_{N_{\hat{C}}})v_{\min}^G(t^*)$ are the optimizers of (30), Corollary 2 certifies that $(y^*)^T N_{\hat{C}}^T N_{\hat{C}} y^* = n+1$ and that $\hat{y}_{k+1}^* = 1$. With $\hat{z}^* = \begin{bmatrix} z^* \\ \hat{z}_{n+1}^* \end{bmatrix} = N_{\hat{C}}\hat{y}^*$ and $\hat{z}_{n+1} = \hat{y}_{n+1}$, we have that z^* prior to rounding is the minimizer of (6). Thus we have shown how to, through Lagrangian relaxation, solve the relaxed, linearly constrained Normalized Cut problem exactly.

Finally, the solution to the relaxed problem must be discretized in order to obtain a solution to the original binary problem (5). This is typically carried out by applying some rounding scheme to the solution.

A. Multi-Class Constrained Normalized Cuts

Multi-class Normalized Cuts is a generalization of (2) for an arbitrary number of partitions,

$$N_{cut}^k = \sum_{l=1}^{k} \frac{cut(A_l, V)}{assoc(A_l, V)}. \tag{35}$$

If one minimizes (35) in an iterative fashion, by, given the current k-way partition, finding a new partition while keeping all but two partitions fixed. This procedure is known as the $\alpha-\beta$-swap when used in graph cuts applications, [6]. The associated subproblem at each iteration then becomes

$$\tilde{N}_{cut}^k = \frac{cut(A_i, V)}{assoc(A_i, V)} + \frac{cut(A_j, V)}{assoc(A_j, V)} + \sum_{l \neq i,j} \frac{cut(A_l, V)}{assoc(A_l, V)} = \\ = \frac{cut(A_i, V)}{assoc(A_i, V)} + \frac{cut(A_j, V)}{assoc(A_j, V)} + c, \tag{36}$$

where pixels not labeled i or j are fixed. Consequently, minimizing the multi-class subproblem can be treated similarly to the bipartition problem. At each iteration we have a problem on the form

$$\begin{aligned} \inf_z \quad & f(z) = \tfrac{z^T(D-W)z}{-z^T dd^T z + (1^T d)^2} \\ \text{s.t.} \quad & z \in \{-1, 1\}^n \\ & Cz = b, \end{aligned} \tag{37}$$

where W, D, C and b will be dependent on the current partition and choice of labels to be kept fixed. These matrices are obtained by removing rows and columns corresponding to pixels not labeled i or j, the linear constraints must also be similarly altered to only involve pixels not currently fixed. Given an initial partition, randomly or otherwise, iterating over the possible choices until convergence ensures a multi-class segmentation that satisfies all constraints. There is however no guarantee that this method will avoid getting trapped in local minimum and producing a sub-optimal solution, but during the experimental validation this procedure always produced satisfactory results.

V. Solving Large-Scale Hermitian Eigenvalue Problems

In an attempt to keep the paper self-contained, this section will give an brief overview to one of the most important methods available for computing eigenvalues and eigenvectors of large matrices. This technique, known as the Lanczos algorithm, is based on projections onto Krylov subspaces. This trivial extension of the simple power iteration turns out to give one of the most powerful methods for extracting eigenvalues of large Hermitian matrices. The method was first introduced in 1950 [13] as a way of reducing an entire matrix to tridiagonal form. Unfortunately, due to issues with round-off errors, the method failed miserably in this capacity. However, twenty years later it was discovered that despite this shortcoming the Lanczos algorithm is still an effective tool for computing extremal eigenvalues and their eigenvectors.

The Krylov subspaces associated with a square symmetric matrix $A \in \mathbb{R}^{n \times n}$ for a vector q_1 is defined as

$$\mathcal{K}_k = \text{span}\left\{q_1, Aq_1, A^2q_1, ..., A^{k-1}q_1\right\}, \tag{38}$$

This simple type of subspace, which is uniquely determined by A and q_1, is of considerable importance for numerous iterative methods for extracting eigenvalues. It turns out that the eigenvalues of the projection of a symmetric matrix A onto a Krylov subspace approximates the actual eigenvalues of A very well. In addition, there exists an orthogonal base Q_k for $\mathcal{K}_k$ that reduces A to a tridiagonal form.

One way of finding an orthogonal base Q_k for a Krylov subspace of a general square matrix A is through a Gram-Schmidt-like procedure known as the Arnoldi method, [1]. This base $Q_k = [\, q_1 \; q_2 \; \cdots \; q_k \,]$ is orthogonal by construction. It can also be shown that each vector $q_j = p_{j-1}(A)q_1$, where p_{j-1} is a (j-1)-th degree polynomial, and that Q_k consequently spans $\mathcal{K}_k$. This base also has the property that it reduces A to an upper Hessenberg matrix which means that the matrix has zero entries below the first subdiagonal. That is

$$Q_k^T A Q_k = H_k, \tag{39}$$

where H_k is upper Hessenberg. Further, if A is symmetric then

$$H_k^T = (Q_k^T A Q_k)^T = Q_k^T A Q_k = H_k \tag{40}$$

and hence H_k must be tridiagonal. Consequently, the orthogonal base Q_k reduces A to the tridiagonal form

$$T_k = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \dots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & \dots & 0 \\ 0 & \beta_3 & \alpha_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \beta_k & \alpha_k \end{bmatrix}. \tag{41}$$

So for Hermitian matrices the Arnoldi method can then be simplified into what is known as the symmetric Lanczos method, [16].

Algorithm 5.1
Lanczos Method for Symmetric Eigenvalue Problems

Begin with $q_0 = 0$ and user supplied starting vector r_0.
$\beta_0 = ||r_0||$
for $i = 1, 2, \dots$ **until** convergence
$q_i = r_{i-1}/\beta_{i-1}$ (1)
$p = Aq_i$ (2)
$\alpha_i = q_i^T p$ (3)
$r_i = p - \alpha_i q_i - \beta_{i-1} q_{i-1}$ (4)
$\beta_i = ||r_i||$ (5)
end

In exact arithmetic arithmetic this algorithm will produce an orthogonal base Q_k for $\mathcal{K}_k$ that also tridiagonalizes A. However, in reality this orthogonality is usually lost in later iteration, owing to round-off errors. In practical Lanczos algorithms some form of reorthogonalizing step is therefore typically incorporated, see [16] for more on this topic.

Originally, the Lanczos method was a procedure for tridiagonalizing a matrix A, it is however its connection to the eigenvalues of A that makes it so interesting. Let $\theta^{(k)}$ and $s^{(k)}$ denote the solution to the resulting tridiagonal eigensystem $T_k s = \theta s$ after k iterations. Since $\mathcal{K}_1 \subset \mathcal{K}_2 \subset \dots \subset \mathcal{K}_n = \mathbb{R}^n$ it follows from the Cauchy interlacing theorem that

$$\lambda_1 = \lambda_1(A) = \theta_1^{(n)} \leq \dots \leq \theta_1^{(2)} \leq \theta_1^{(1)}. \tag{42}$$

Consequently, a side effect of the Lanczos algorithm (5.1) is that it will will produce a decreasing sequence of eigenvalues $\theta_1^{(k)}$ that approaches λ_1. A natural consequence is then to take $\theta^{(k)}$ and $s^{(k)}$ as approximations of the eigenvalues and eigenvectors of A, with $\lambda = \theta^{(k)}$ and $v = Q_k s^{(k)}$. By continuing the Lanczos method, solving the $k \times k$ system $T_k s = \theta s$ at each iteration and terminating when the norm of the residual ($||(AQ - \theta^{(k)})s^{(k)}||$) is sufficiently small, we can obtain eigenvalues that are arbitrarily close to λ_1. From computational point of view, a crucial property here is also the tridiagonality of T_k, since such such eigensystems can be solved extremely efficiently. This clearly motivates the use of Krylov subspaces.

The Lanczos procedure 5.1 can easily be extended to handle generalized eigenvalue problem for positive definite symmetric matrices A and M. We present this algorithm here without any further discussion, see [16].

Algorithm 5.2
Lanczos Method for Generalized Eigenvalue Problems

Begin with user supplied starting vector u_0.
$r_0 = Mu_0$
$\beta_0 = \sqrt{u_0^T r_0}$
$p_0 = 0$
for $i = 1, 2, \dots$ **until** convergence
$q_i = u_{i-1}/\beta_{i-1}$ (1)
$\bar{u}_i = Aq_i - p_{i-1}\beta_{i-1}$ (2)
$\alpha_i = q_i^T \bar{u}_i$ (3)
$p_i = r_i/\beta_{i-1}$ (4)
$r_i = \bar{u}_i - p_i \alpha_i$ (5)
$u_i = M^{-1} r_i$ (6)
$\beta_i = \sqrt{u_i^T r_i}$ (7)
end

This algorithm produces a base for $\mathcal{K}_k$ that tridiagonalizes A as before but instead of being orthogonal, is M-orthogonal. That is

$$Q_k^T A Q_k = T_k \tag{43}$$
$$Q_k^T M Q_k = I_k, \tag{44}$$

effectively reducing the generalized eigenvalue problem $Ax = \lambda Mx$ again into a tridiagonal eigensystem $T_k s = \theta s$.

One of the major benefits of Lanczos methods is that the entering matrices does not have to be directly defined, instead they can be implicitly defined through operators that return how A, M and $M^{-1}A$ acts upon arbitrary vectors, corresponding to step (2) and (3) in algorithm (5.2). This makes this procedure especially well suited for the type of large sparse and structured matrices we deal with in this work.

VI. Efficient Optimization

A. Subgradient Optimization

First we present a method, similar to that used in [15] 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 (25) 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 [2], [15].

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

$$g(\sigma) \leq g(\sigma_0) + v^T(\sigma - \sigma_0), \quad \forall \sigma \in \mathbb{R}^{k+1}. \tag{45}$$

One can show that if a function is differentiable then the derivative is the only vector satisfying (45). 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

$$v = \frac{\hat{y}_0^T E_{\hat{C}} \hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0} \tag{46}$$

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: The statement follows from

$$\theta(t) = \min_{||\hat{y}||^2_{N_{\hat{C}}}=1} \frac{\hat{y}^T (L_{\hat{C}} + tE_{\hat{C}})\hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}} \leq \frac{\hat{y}_0^T (L_{\hat{C}} + tE_{\hat{C}})\hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0} =$$
$$\frac{\hat{y}_0^T (L_{\hat{C}} + t_0 E_{\hat{C}})\hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0} + \frac{\hat{y}_0^T E_{\hat{C}} \hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0}(t - t_0) =$$
$$= \theta(t_0) + v^T (t - t_0). \tag{47}$$

■

1) 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

$$\theta_i(t) = F(\hat{y}_i, t_i) + v^T (t - t_i). \tag{48}$$

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

$$\bar{\theta}(t) = \inf_{i \in I} F(\hat{y}_i, t_i) + v^T (t - t_i) \tag{49}$$

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

$$\begin{aligned} &\sup_{t,\alpha} \alpha \\ &\alpha \leq F(\hat{y}_i, t_i) + v^T (t - t_i), \ \forall i \in I, \ t_{min} \leq t \leq t_{max} \end{aligned} \tag{50}$$

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 (50). 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 (50) 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 $\sup_t \theta(t)$.

2) Initialization: In order for the problem (50) to have a meaningful (finite) solution the set I needs to have at least size two. Further more, since the function is concave, there must be $i \in I$ and $j \in I$ $(i \neq j)$ such that $t_i \leq t^* \leq t_j$, where t^* is the optimal solution. In order to achieve this we will start the algorithm by using the asymptotic behavior of $\theta(t)$.

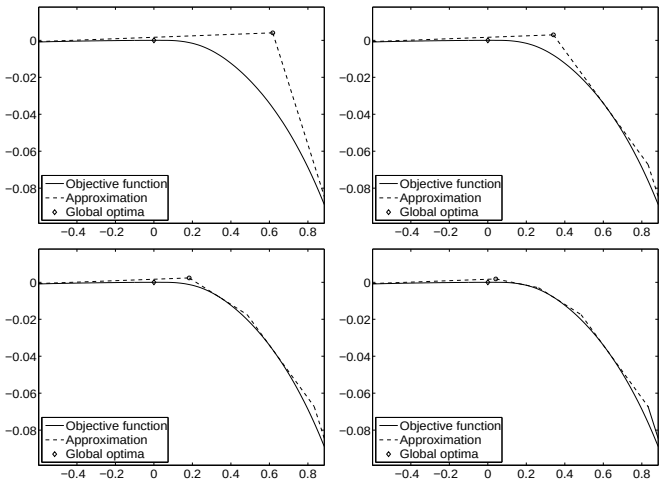

Fig. 1. Approximations of a randomly generated objective function after the first four iterations of the algorithm.

Let $h_+ = at + b$ and $h_- = ct + d$ be the asymptote as $t \to \pm\infty$ respectively. To find a we need to compute the limit value of $\frac{\theta(t)}{t}$ as $t \to \infty$.

$$a = \lim_{t\to\infty} \frac{\theta(t)}{t} = \lim_{t\to\infty} \frac{1}{t} \left(\min_{\hat{y}} \frac{\hat{y}^T (L_{\hat{C}} + tE_{\hat{C}})\hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}} \right) = \tag{51}$$
$$= \lim_{t\to\infty} \min_{\hat{y}} \left(\frac{1}{t} \frac{\hat{y}^T L_{\hat{C}} \hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}} + \frac{\hat{y}^T E_{\hat{C}} \hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}} \right) = \tag{52}$$
$$= \min_{\hat{y}} \frac{\hat{y}^T E_{\hat{C}} \hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}} = \lambda_1(E_{\hat{C}}, M_{\hat{C}}) = \frac{\hat{y}_1^T E_{\hat{C}} \hat{y}_1}{\hat{y}_1^T M_{\hat{C}} \hat{y}_1} \tag{53}$$

Similarly for the asymptote at $-\infty$ we get

$$c = \lim_{t\to-\infty} \frac{\theta(t)}{t} = \lambda_1(-E_{\hat{C}}, M_{\hat{C}}) = \tag{54}$$
$$= \lambda_n(E_{\hat{C}}, M_{\hat{C}}) = \frac{\hat{y}_n^T E_{\hat{C}} \hat{y}_n}{\hat{y}_n^T M_{\hat{C}} \hat{y}_n} \tag{55}$$

where λ_1 and λ_n are the smallest and largest generalized eigenvalues of $(E_{\hat{C}}, M_{\hat{C}})$, the corresponding eigenvectors are denoted $\hat{y}_1$ and $\hat{y}_n$.

Finding b requires us to compute $\lim_{t\to\infty} \theta(t) - at$.

$$b = \lim_{t\to\infty} \theta(t) - at = \lim_{t\to\infty} \left(\min_{\hat{y}} \frac{\hat{y}^T (L_{\hat{C}} + tE_{\hat{C}})\hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}} \right) \tag{56}$$
$$- \frac{\hat{y}_1^T E_{\hat{C}} \hat{y}_1}{\hat{y}_1^T M_{\hat{C}} \hat{y}_1} t = \frac{\hat{y}_1^T L_{\hat{C}} \hat{y}_1}{\hat{y}_1^T M_{\hat{C}} \hat{y}_1} \tag{57}$$

And d becomes

$$d = \lim_{t\to-\infty} \theta(t) - ct = \frac{\hat{y}_n^T L_{\hat{C}} \hat{y}_n}{\hat{y}_n^T M_{\hat{C}} \hat{y}_n} \tag{58}$$

Thus initializing the algorithm only requires finding the extremal eigenvalues for the pencil $(E_{\hat{C}}, M_{\hat{C}})$. As this does not involve the Laplacian matrix $L_{\hat{C}}$ this eigenproblem can be solved very little computational effort.

B. 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 (25). 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 fulfill the following equations,

$$((L_{\hat{C}} + tE_{\hat{C}}) - \lambda(t)M_{\hat{C}})\hat{y}(t) = 0 \tag{59}$$

$$||\hat{y}(t)||^2_{N_{\hat{C}}} = n + 1. \tag{60}$$

To emphasize the dependence on t we write $\lambda(t)$ for the eigenvalue and $\hat{y}(t)$ for the eigenvector. By differentiating (59) one obtains

$$(E_{\hat{C}} - \lambda'(t)M_{\hat{C}})\hat{y}(t) + ((L_{\hat{C}} + tE_{\hat{C}}) - \lambda(t)M)\hat{y}'(t) = 0. \tag{61}$$

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 (60), obtaining

$$\hat{y}^T(t){N_{\hat{C}}}^T N_{\hat{C}}\hat{y}'(t) = 0. \tag{62}$$

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

$$\begin{bmatrix} (L_{\hat{C}}+tE_{\hat{C}})-\lambda(t)M_{\hat{C}} \\ \hat{y}^T(t){N_{\hat{C}}}^T N_{\hat{C}} \end{bmatrix} \hat{y}'(t) = \begin{bmatrix} (-E_{\hat{C}}+\lambda'(t)M_{\hat{C}})\hat{y}(t) \\ 0 \end{bmatrix}. \tag{63}$$

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 (46),

$$\theta''(t) = \frac{d}{dt}\theta'(t) = \frac{d}{dt}\frac{\hat{y}(t)^T E_{\hat{C}}\hat{y}(t)}{\hat{y}(t)^T M_{\hat{C}}\hat{y}(t)} \tag{64}$$

$$= \frac{2}{\hat{y}(t)^T M_{\hat{C}}\hat{y}(t)}\hat{y}^T(t)\left(E_{\hat{C}} - \theta'(t)M_{\hat{C}}\right)\hat{y}'(t). \tag{65}$$

1) 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 optimum t^*, Newton's method for unconstrained optimization can be applied. It follows from these properties of $\theta(t)$ that Newton's method [2] 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^*, denoted by $t_{\min}$, $t_{\max}$, by looking at the derivative of the objective function for these values of $t = t_i$,

$$t_{\max} = \min_{i;\theta'(t_i)\leq 0} t_i, \text{ and } t_{\min} = \max_{i;\theta'(t_i)\geq 0} t_i. \tag{66}$$

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

$$\theta(t) \approx \theta(t_i) + \theta'(t_i)(t - t_i) + \frac{\theta''(t_i)}{2}(t - t_i)^2 \tag{67}$$

and finding its maximum. By differentiating (67) it is easily shown that its optimum, as well as the next point in the Newton sequence, is given by

$$t_{i+1} = -\frac{\theta'(t_i)}{\theta''(t_i)} + t_i. \tag{68}$$

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.

C. Approximating Derivatives of Eigenvalues and Eigenvectors

The use of second order derivatives for maximizing (25), as discussed in the previous section, should significantly reduce the number of required iterations. The algebraic expression for $\theta''(t)$ in (65) does have a significant disadvantage. It requires solution of a very large linear system (63), this task can be as demanding as determining the smallest generalized eigenvalue of $(L_{\hat{C}} + tE_{\hat{C}}, M_{\hat{C}})$.

This means that we reduce the number of iterations but also increase the computational effort needed at each step. In this section we discuss how one can compute an approximation of the second derivative of the smallest eigenvalue.

The underlying idea is best explained by, instead of (63), looking at the unconstrained optimization problem

$$\min_x \quad x^T(L_{\hat{C}} + tE_{\hat{C}} - \lambda M_{\hat{C}})x - 2b^T x. \tag{69}$$

Since $(L_{\hat{C}} + tE_{\hat{C}} - \lambda M_{\hat{C}}) \succeq 0$ and $(L_{\hat{C}} + tE_{\hat{C}} - \lambda M_{\hat{C}})v = 0$, a solution to this problem is given by $x^* = (L_{\hat{C}} + tE_{\hat{C}} - \lambda M_{\hat{C}})^+ b$, for which $v^T x^* = 0$, thus minimizing (69) is equivalent to solving (63).

If we now constrain the above program to some m-dimensional linear subspace P of $\mathbb{R}^n$ we get

$$\min_{x \in P \subseteq \mathbb{R}^n} \quad x^T(L_{\hat{C}} + tE_{\hat{C}} - \lambda M_{\hat{C}})x - 2b^T x. \tag{70}$$

Letting U be a base for P we can write (70) as

$$\min_{y \in \mathbb{R}^m} \quad y^T U^T(L_{\hat{C}} + tE_{\hat{C}} - \lambda M_{\hat{C}})Uy - 2b^T Uy. \tag{71}$$

The optima of this problem y^* will most likely not be equal to x^* and will hence only be an approximate solution to (70). From the equivalence to problem (63), y^* can consequently also be regarded as an approximate solution to that linear system. We

have that $(x^*)^T(L_{\hat{C}}+tE_{\hat{C}}-\lambda M_{\hat{C}})x^*-2b^Tx^* \leq (y^*)^TU^T(L_{\hat{C}}+tE_{\hat{C}}-\lambda M_{\hat{C}})Uy^*-2b^TUy^*$, obviously with equality if $m=n$.

How well the solution to (71) approximates (63) will clearly depend on the subspace P, so a great deal of care is needed when choosing U. Ideally, the resulting system should also one that can be solved with relative ease. It turns out that the base for the Krylov space Q_k associated with the matrices $L_{\hat{C}}+tE_{\hat{C}}$ and $M_{\hat{C}}$ is a good choice. As this base is has already been computed when determining the generalized eigenvalues of $(L_{\hat{C}}+tE_{\hat{C}}, M_{\hat{C}})$ no additional work is needed. Recalling that Q_k simultaneously tridiagonalizes both $L_{\hat{C}}+tE_{\hat{C}}$ and $M_{\hat{C}}$, that is $Q_k^T(L_{\hat{C}}+tE_{\hat{C}})Q_k=T_k$ and $Q_k^TM_{\hat{C}}Q_k=I$, inserting Q_k into (71) gives

$$\min_{y\in\mathbb{R}^m} \quad y^TQ_k^T(L_{\hat{C}}+tE_{\hat{C}}-\lambda M_{\hat{C}})Q_ky-2b^TQ_ky= \tag{72}$$

$$=y^T(T_k-\lambda I)y-2b^TQ_ky. \tag{73}$$

A solution to this problem is given by $y^*=(T_k-\lambda I)^+Q_k^Tb$, with $\tilde{x}=Q_ky$ an approximate solution to (70) will then be

$$\tilde{x}=Q_k^T(T_k-\lambda I)^+Q_k^Tb. \tag{74}$$

Since typically $k<<n$, we now have not only a much smaller problem but also one that is tridiagonal, such systems can be solved extremely efficiently.

Combining (74) with (65) we can give a formulation for an approximation of the second derivative of our objective function (25).

$$\tilde{\theta}_k''(t)=-b^T\tilde{x}= \tag{75}$$

$$=\frac{-2}{v^TMv}v^T(\lambda'M-E)^TQ_k(T_k-\lambda I)^+Q_k^T(\lambda'M-E)v \tag{76}$$

Since $v=Q_ks$ and $(T_k-\lambda I)s=0$ we can simplify this expression to

$$\tilde{\lambda}_k''(t)=\frac{-2}{v^TMv}v^TE^TQ_k(T_k-\lambda I)^+Q_kEv \tag{77}$$

We can now use the approximation of $\theta''(t)$ in the Newton-like method of section VI-B.1 in order to maximize the concave Lagrangian dual function (25). However since we now only have an overestimating approximation of $\theta''(t)$ we can not be certain of how this method will now behave. In the following section we will show experimentally that the approximation of $\theta''(t)$ still results in an efficient algorithm.

VII. Experimental Validation

The experiments are divided into two separate parts. The first one evaluates the proposed reformulation of Normalized Cuts and linear grouping constraints. The second part evaluates the different numerical methods, discussed previously, for efficiently solving the resulting optimization problem.

A. Normalized Cuts Reformulation and Linear Grouping Constraints

A number of experiments were conducted to evaluate our proposed formulation but also to illustrate how relevant visual information can be incorporated into the segmentation process through non-homogeneous, linear constraints and how this can influence the partitioning.

All images were gray-scale of approximately 100-by-100 pixels in size. The affinity matrix was calculated based on edge information, as described in [14]. The one-dimensional maximization over t was carried out using a golden section search, typically requiring $15-20$ eigenvalue calculations. The relaxed solution z was discretized by simply thresholding at 0.

Firstly, we compared our approach with the standard Normalized Cut method, Fig. 2. Both approaches produce similar

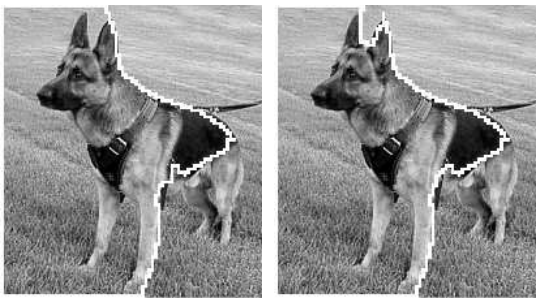

Fig. 2. Image segmentation using the standard Normalized Cut algorithm (left) and the reformulated Normalized Cut algorithm with no constraints (right).

results, suggesting that in the absence of constraints the two formulations are equivalent. However, where our approach has the added advantage of being able to handle linear constraints.

The simplest such constraint might be the hard coding of some pixels, i.e. pixel i should belong to a certain class. This can be expressed as the linear constraints $z_i=\pm 1$, $i=1,\ldots,m$. In Fig. 3 it can be seen how a number of such hard constraints influences the segmentation of the image in Fig. 2.

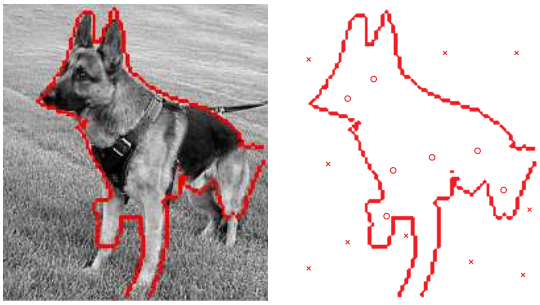

Fig. 3. Image segmentation with constraints (left) and constraints applied (right).

Another visually significant prior is the size or area of the resulting segments, that is, constraints such as $\sum_i z_i=1^Tz=a$. The impact of enforcing limitations on the size of the partitions is shown in Fig. 4.

Excluding and including constraints such as, pixel i and j should belong to the same or separate partitions, $z_i+z_j=0$ or $z_i-z_j=0$, is yet another meaningful constraint. The result of including a combination of all the above types of constraints can be seen in Fig. 5.

Finally, we also performed a multi-class segmentation with linear constraints, Fig. 6.

We argue that these results, not only indicate a satisfactory performance of the suggested method, but also illustrate the relevance of linear grouping constraints in image segmentation and the impact that they can have on the resulting partitioning. These experiments also indicate that even a simple rounding scheme as the one used here can often suffice. As we threshold

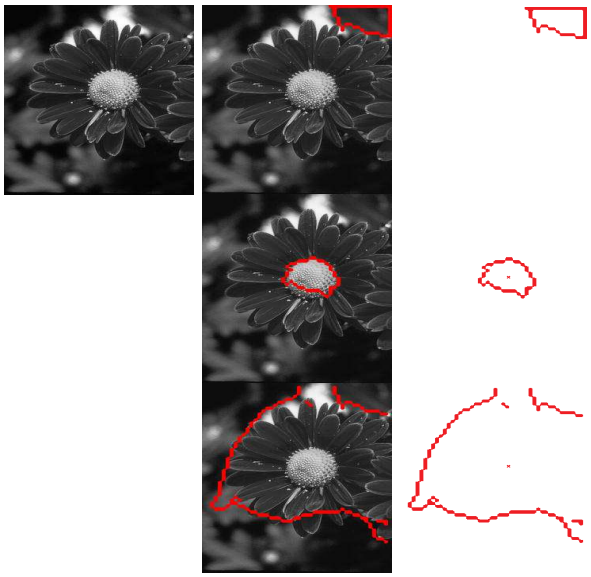

Fig. 4. Original image (top left), segmentation without constraints (top middle) and segmentation boundary and constraints applied (top right). Segmentation with area constraints, (area=100 pixels) (middle left), segmentation boundary and constraints applied (middle right). Segmentation with area constraints, (area=2000 pixels) (bottom left), segmentation boundary and constraints applied (bottom right).

at zero, hard, including and excluding constraints are all ensured to hold after discretizing. Only the area constraints are not guaranteed to hold, however, since the relaxed solution has the correct area, thresholding it typically produces a discrete solution with roughly the correct area.

B. Numerical Experiments

In this section a number of experiments were conducted in an attempt to evaluate the suggested numerical 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 [2], used in [9], 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 matrix pencil $(L_{\hat{C}} + tE_{\hat{C}}, M_{\hat{C}})$ we wish to reduce the number of such calculations. Fig. 7 shows the required number of eigenvalue evaluations for the subgradient method, the Newton method and its approximation, as well as the baseline golden section search.

The two Newton methods clearly outperform the subgradient approach and golden section search. The difference between the standard Newton and the approximate Newton methods is not as discernible. It appears that the approximation of the first and second order derivatives of the smallest generalized eigenvalue produced by the base of the Krylov space is sufficiently accurate to ensure fast convergence.

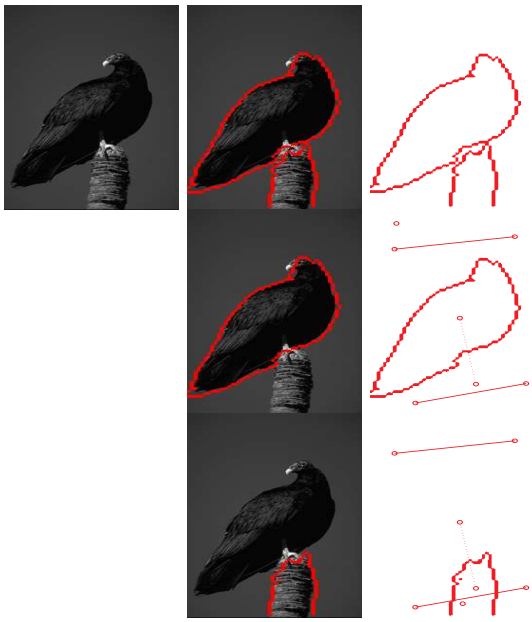

Fig. 5. Original image (top left), segmentation without constraints (top middle), segmentation boundary and constraints applied (top right). Segmentation with hard, including and excluding, as well as area constraints, (area=25% of the entire image) (middle left), segmentation boundary and constraints applied (middle right). Segmentation with constraints, (area=250 pixels) (bottom left), segmentation boundary and constraints applied (bottom right). Here a solid line between two pixels indicates an including constraint, and a dashed line an excluding.

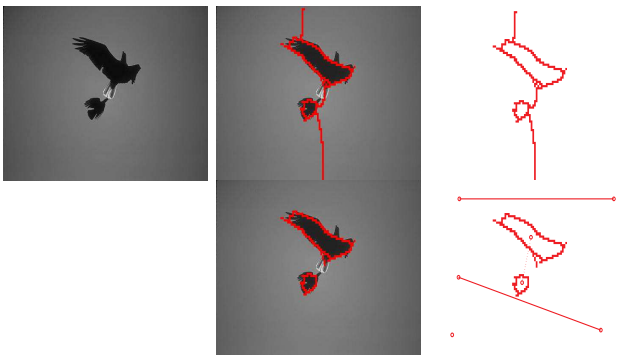

Fig. 6. Original image (top left), three-class segmentation without constraints (top middle), segmentation boundary (top right). Three-class segmentation with hard, including and excluding constraints (bottom left), segmentation boundary and constraints applied (bottom right).

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 used was the same as 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. The second image is of a 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 known and included as linear constraint into the segmentation process. The resulting partition can be seen in

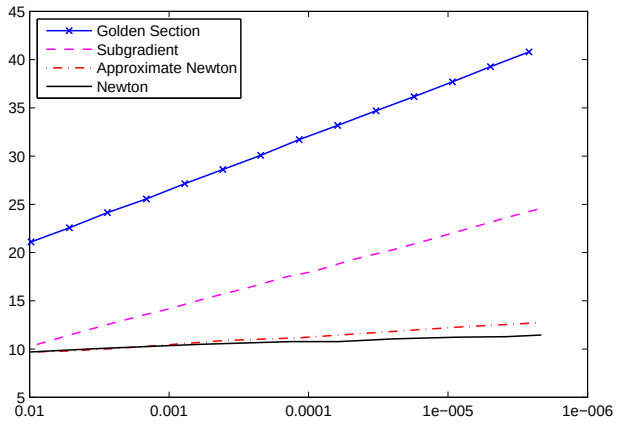

Fig. 7. The average number of eigenvalue evaluations required by the different algorithms, as a function of tolerance, for 100 synthetically generated experiments.

Fig. 9.

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 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. 8 and 9 how the subgradient methods converge quickly in the initial iterations only to slow down as it approaches the optimum. This is in support of the above discussion regarding the linear appearance of the function $\theta(t)$ far away from the optimum. We therefore expect the modified Newton method to be superior when higher accuracy is required.

In conclusion we have proposed three 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 methods are superior.

VIII. Conclusions

We have presented a reformulation of the classical Normalized Cut problem that allows for the inclusion of linear grouping constraints into the segmentation procedure, through a Lagrangian dual formulation. A method for how to efficiently find such a cut, even for very large scale problems, has also been given. A number of experiments as well as a theoretical proof were also supplied in support of these claims.

Improvements to the presented method include, firstly, the one-dimensional search over t. As the dual function is the point-wise infimum of the eigenvalues of a matrix, it is sub-differentiable and utilizing this information should greatly reduce the time required for finding t^*. Then, an issue that was left open in this work is the rounding scheme. The relaxed solution z is currently discretized

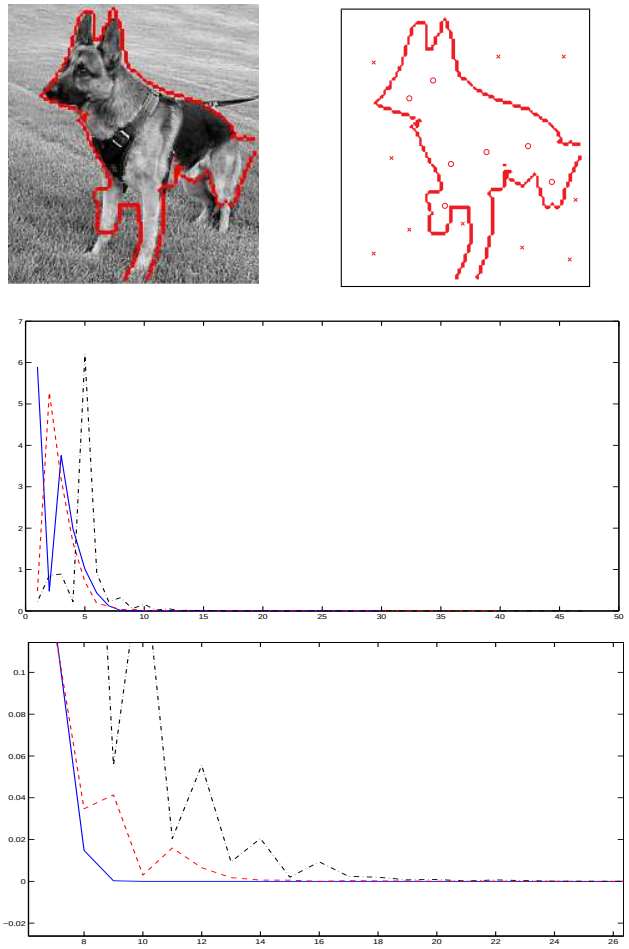

Fig. 8. Top: Resulting segmentation (left) and constraints applied (right). Here an X means that this pixel belongs to the foreground and an O to the background. Bottom: Convergence of the modified Newton (solid), subgradient (dashed) and the golden section (dash-dotted) algorithms. The algorithms converged after 9, 14 and 23 iterations, respectively.

by simple thresholding at 0. Even though we can guarantee that z prior to rounding fulfills the linear constraints, this is not necessarily true after thresholding and should be addressed. For simpler constraints, as the ones used here, rounding schemes that ensure that the linear constraints hold can easily be devised. An in-depth discussion on different procedures for discretization is outside the scope of the present paper.

Finally, the question of properly initializing the multi-class partitioning should also be investigated as it turns out that this choice can affect both the convergence speed and the final result.

Acknowledgments

This work has been funded by the Swedish Research Council (grant no. 2007-6476), by the Swedish Foundation for Strategic Research (SSF) through the programme Future Research Leaders, by the European Research Council (GlobalVision grant no. 209480), and under the Australian Research Council's Discovery Projects funding scheme (project DP0988439).

References

[1] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(1):1729, 1951.

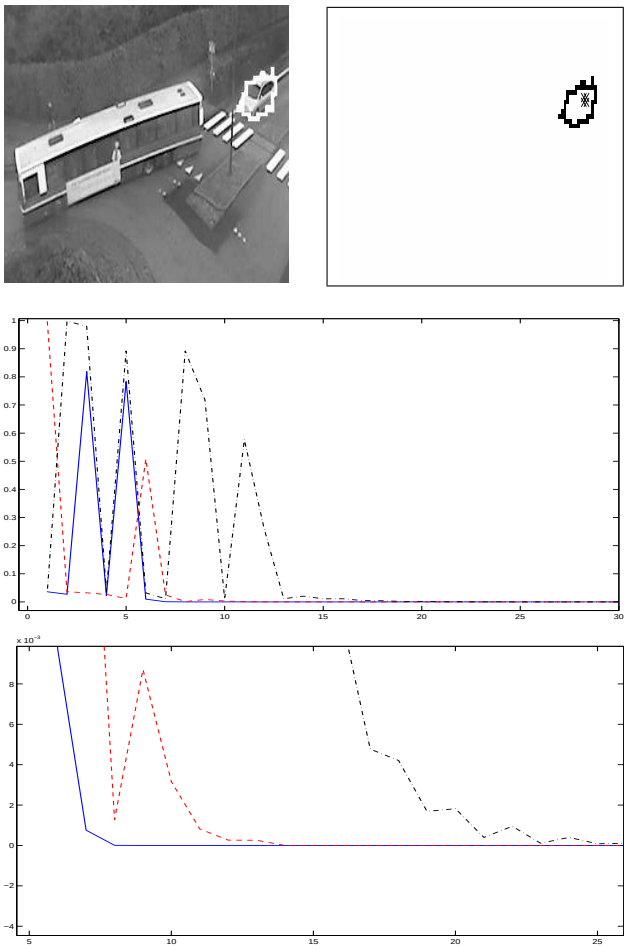

Fig. 9. Top: Resulting segmentation (left) and constraints applied, in addition to the area requirement used (area = 50 pixels) (right). Here the X in the top right part of the corner means that this pixel belongs to the foreground. Bottom: Convergence of the modified Newton (solid), subgradient (dashed)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

[2] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming, Theory and Algorithms*. Wiley, 2006.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *International Conference on Computer Vision*, pages 05–112. Vancouver, Canada, 2001.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[7] T. Cour and J. Shi. Solving markov random fields with spectral relaxation. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 11, 2007.

[8] A. Eriksson, C. Olsson, and F. Kahl. Image segmentation with context. In *Proc. Conf. Scandinavian Conference on Image Analysis*, Ahlborg, Denmark, 2007.

[9] A. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. In *International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[10] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *Int. Journal Computer Vision*, 59(2):167–181, 2004.

[11] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences, 1996.

[12] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, Nov. 2006.

[13] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the Natural Bureau of Standards*, 45(4):225–281, 1950.

[14] J. Malik, S. Belongie, T. K. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[15] C. Olsson, A. Eriksson, and F. Kahl. Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. In *Proc. Conf. Computer Vision and Pattern Recognition*, Mineapolis, USA, 2007.

[16] B. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, 1980.

[17] F. Rendl and H. Wolkowicz. A Semidefinite Framework for Trust Region Subproblems with Applications to Large Scale Minimization. Technical Report CORR 94-32, Departement of Combinatorics and Optimization, December 1994.

[18] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics*, pages 309–314, 2004.

[19] C. Shen, H. Li, and M. Brooks. A convex programming approach to the trace quotient problem. In *Asian Conf. Computer Vision*, Tokyo, Japan, 2007.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[21] D. C. Sorensen and C. Yang. Truncated QZ methods for large scale generalized eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 19(4):1045–1073, 1998.

[22] D. Tolliver and G. Miller. Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1053–1060, New York City, USA, 2006.

[23] T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for map-mrf. In *Proc. Conf. Computer Vision and Pattern Recognition*, Anchorage, USA, 2008.

[24] L. Xu, W. Li, and D. Schuurmans. Fast normalized cut with linear constraints. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 2866–2873, 2009.

[25] S. Yu and J. Shi. Multiclass spectral clustering. In *International Conference on Computer Vision*. Nice, France, 2003.

[26] S. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(26):173–183, 2004.