Optimization Methods for Large Scale Combinatorial Problems and Bijectivity Constrained Image Deformations

Anders Eriksson

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden

http://www.maths.lth.se/

Doctoral Theses in Mathematical Sciences 2008:2
ISSN 1404-0034

ISBN 978-91-628-7417-9
LUTFMA-1029-2008

Printed in Sweden by MEDIA-TRYCK, Lund 2008

Preface

The thesis is based on the following work

1. Eriksson, A., Olsson, C. and Kahl F., Efficiently Solving the Fractional Trust Region Problem. In *Asian Conference on Computer Vision, ACCV, Tokyo*, Nov. 2007.
2. Eriksson, A., Olsson, C. and Kahl F., Normalized Cuts Revisited: A Reformulation for Segmentation with Linear Grouping Constraints. In *International Conference on Computer Vision, ICCV, Brazil*, Oct. 2007.
3. Olsson, C., Eriksson, A. and Kahl F., Improved Spectral Relaxation Methods for Binary Quadratic Optimization Problems Submitted to *Journal of Computer Vision and Image Understanding*, 2007.
4. Olsson, C., Eriksson, A. and Kahl F., Solving Large Scale Binary Quadratic Problems: Spectral Methods vs. Semidefinite Programming. *Proc. CVPR, Minneapolis, USA*, 2007.
5. Eriksson, A., Olsson, C. and Kahl F., Segmenting with Context. *Proc. SCIA, Aalborg, Denmark*, 2007.
6. Eriksson, A. and Åström, K., Image Registration using Thin-Plate Splines. In *Proc. ICPR, Hong Kong, China*, 2006.
7. Eriksson, A., Barr, O. and Åström, K. Image Segmentation Using Minimal Graph Cuts. To appear *Proc. SSBA Symposium on Image Analysis, Umeå, Sweden*, 2006.
8. Eriksson, A. and Åström, K., On the Bijectivity of Thin-plate Splines. In *Proc. SSBA Symposium on Image Analysis, Malmö, Sweden*, 2005.
9. Eriksson, A., Licentiate thesis, *Lund University*, 2005.

Subsidiary Papers

10. Eriksson, A. and Åström, K., Robustness and specificity in object detection. In *Proc. ICPR, Cambridge, UK*, 2004.
11. Balkenius, C., Åström, K. and Eriksson, A. Learning in visual attention, In *Proc. of International Workshop on Learning for Adaptable Visual Systems, ICPR, Cambridge, UK*, 2004.

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

I want to thank my department for giving me opportunity to commence my studies in the first place.

I am deeply indebted to my supervisor Kalle Åström. Without his unyielding support and seemingly infinite patience this work would have never been completed. During my time here he has always provided encouragement, sound advice, lots of good ideas and very good company.

All the members of the Mathematical Imaging Group has supported me tremendously in my work. I would especially like to thank Carl Olsson and Fredrik Kahl who provided invaluable assistance and advice as well as many interesting discussions. It has been a pleasure working with you both.

Finally, I would also like to thank all my colleagues at the department for their help, support, interest and valuable hints and for making my time here so much more enjoyable.

Introduction

1 Background

The field of computer vision has, ever since it emerged in the 60's, been full of optimization problems. Looking back, the subfield where optimization perhaps first made an impact was in projective geometry. Structure and motion problems were early on solved either by local methods or some algebraic approach, or a combination of both. Typical examples of this are the 8-point algorithm [11] and bundle adjustment [9]. For a long time this was the predominant approach, either the problem was reformulated in such a way that it could be easily solved or one employed some local refinement method directly to the problem at hand. This usually either resulted in a formulation that lacked a meaningful connection to the original problem or in the latter case in a solution which one could not be certain was the desired one. It is only in the last decade that optimization theory has received the attention it perhaps rightly deserves within this field. Recently several publications have appeared that address the issue of how to appropriately formulate a wide variety of optimization problems in computer vision. Formulations with objective functions that has a definite relevance to the problem at hand. And stated in such a way that they, not only can be efficiently solved, but where one can also say something about the quality of the solution obtained. Examples of such work includes the L_∞-norm formulations of [8] and [10], the Gröbner base approach of [15] and the branch-and-bound technique of [1] Here terms such as convexity and globally optimal solutions are central.

Discrete optimization techniques are possibly not as commonly occurring in computer vision as their continuous counterpart, but whose relevance has still proven to be considerable. Vision problems as varying as image segmentation and grouping, image enhancement and denoising, motion segmentation, tracking and object recognition can be formulated as discrete, or combinatorial, optimization problems. The work that perhaps popularized discrete techniques and first reached a wider audience within the field was the normalized cuts approach of [14]. This method, based on spectral relaxation techniques, has successfully been used in a wide variety of applications. However, the discrete optimization technique that lately has received the most attention is the graph cut algorithm of [4]. This is a method that can exactly minimize certain objective functionals in polynomial time. Its attraction can be attributed to a thorough understanding of both its underlying theory as well as the properties of the set of problems to which it can be applied. Several implementations of this method with exceptional computational complexity are also widely available.

Even though continuous and discrete optimization techniques for vision problems

has largely evolved separately they are closely related. It can also be argued on a purely theoretical basis that they share a common underlying framework. In addition, with the above exception, most combinatorial problems that arise in computer vision are known to be NP-hard and can not be solved optimally in reasonable time. A common approach is to instead look for an approximate solution to the combinatorial problem by dropping the discrete constraints, a process called relaxation, thus turning the problem into a continuous one that can be solved efficiently. The first half of this thesis deals with discrete optimization problems of this category.

Next we will present a brief overview of the two applications treated in this thesis. Followed by a an introduction to some of the basic concepts in optimization theory. In the final section of this introduction a more detailed summary of the separate papers is given.

2 Overview

This thesis treats two separate but connected themes. Namely, **image segmentation** and **image deformation**. This affiliation originates in optimization being the common choice of method for solving most of the occurring challenges.
The thesis consists of the following six segments of work:

I Eriksson, A., Olsson, C. and Kahl F., *Normalized Cuts Revisited: A Reformulation for Segmentation with Linear Grouping Constraints*, In Proc. International Conference on Computer Vision, ICCV, Brazil, Oct. 2007.

II Eriksson, A., Olsson, C. and Kahl F., *Efficiently Solving the Fractional Trust Region Problem*, In Proc. Asian Conference on Computer Vision, ACCV, Tokyo, Nov. 2007.

III Olsson, C., Eriksson, A. and Kahl F., *Improved Spectral Relaxation Methods for Binary Quadratic Optimization Problems*, Submitted to Journal of Computer Vision and Image Understanding, 2007.

IV Eriksson, A., *Bijective Thin-Plate Splines*, In Licentiate thesis, chapter 2, Lund University, 2005.

V Eriksson, A. and Kalle Åström, *Image Registration using Thin-Plate Splines*, In Proc. International Conference on Pattern Recognition, ICPR, Hong Kong, China, 2006.

VI Eriksson, A., *Groupwise Image Registration and Automatic Active Appearance Model Generation*, In Licentiate thesis, chapter 4, Lund University, 2005.

The first theme of the thesis is image segmentation. This is usually defined as the task of distinguishing objects from background in unseen images. This visual grouping process is typically based on low-level cues such as intensity, homogeneity or image

contours. Popular approaches include thresholding techniques, edge based methods and region-based methods. Regardless of the method, the difficulty lies in formulating and describing the perception of what constitutes foreground and background in an arbitrary image. Furthermore, such a grouping is also highly contextually driven, certain image regions may be labeled differently depending on the task at hand - are we looking for people, buildings or trees? If one also allows for more labels than only foreground and background, the problem becomes increasingly harder and requires a much higher level of scene understanding.

Once a formulation of the problem has been established and properly stated the question of how to efficiently solve it still remains. The complexity of this task and the size of most natural images typically leads to very large and difficult optimization problems. It is these issues we make an attempt at addressing in this thesis. We are interested in how to efficiently find visually relevant image partitions as well as how prior information can be included into the segmentation process.

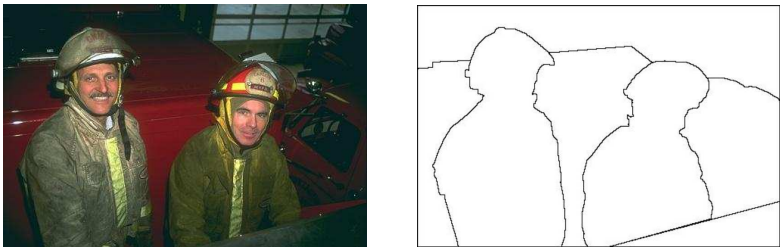


Figure 1.1: An example of an image segmentation.

The methods investigated in this work is based on techniques from combinatorial optimization, a choice that was mainly motivated by the proven success of these approaches. Such problems can be stated as large-scale quadratic 0-1 optimization programs with linear constraints. Usually, such proven NP-complete problems are solved by relaxing, or simply dropping, some constraints and rewriting the problem as one that can be solved efficiently, In paper I and II we study large scale fractional quadratic 0-1 programs. We present a reformulation of this class of optimization problems that in a unified way can handle any type of linear equality constraints as well as proposing efficient algorithms for solving them. Paper III introduces two new methods for solving binary quadratic problems that greatly improves on existing and established methods for finding solutions in the large scale case. Both these proposed methods have been applied to several vision problems with promising results.

The second theme of this thesis concerns non-linear deformations of images and its applications. Functions that map $\mathbb{R}^2$ onto itself are widely used in computer vision, medical imaging and computer graphics. What is common to all three is that mappings

are used to model deformation occurring in natural images. As such deformations are highly complex they are near impossible to characterize. A reasonable and widely accepted assumption, or approximation, is that as the overall structure of the objects depicted will remain intact after deformation, hence folding or tearing of the images should never occur, see fig 1.2. Under these premises there must exist a dense mapping that is both one-to-one and onto. The deformations must be bijective. This is not entirely correct as for instance self-occlusion can not be described by bijective mappings.

Figure 1.2: Examples of bijective and non-bijective deformations.

There exist an abundance of methods for parameterizing non-linear deformations. This part of the thesis concerns conditions for bijectivity of, perhaps the most commonly used method of describing non-linear deformations, the thin-plate spline mapping and its applications in computer vision. Paper IV discusses the thin-plate spline and In paper V and VI we apply the results of paper IV to the task of pair-wise and group-wise registering of images.

3 Optimization

This section provides a brief review of some of the basic concepts of optimization, used in this thesis.

Optimization refers to the task of minimizing (or maximizing) a real-valued function $f : S \mapsto \mathbb{R}$, (the objective function) over a given set S. The term mathematical programming is also commonly occurring. [1]

[1] The name bears no reference to computer programming but instead originates from the research in optimization conducted by the United States army in the 1940's applied to logistical planning and personnel scheduling. "Program" here refers to its military meaning of sequence of operations. The inclusion of this word supposedly increased chances for receiving governmental funding at the time.

A typical notation is

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & x \in S. \end{aligned} \tag{1.1}$$

The function f is said to have a **global minima** at x^* if

$$f(x) \geq f(x^*), \qquad \forall x \in S. \tag{1.2}$$

The function f is said to have a **local minima** at x^* if there is a neighborhood $||x - x^*|| < \delta$ such that

$$f(x) \geq f(x^*), \qquad \forall x \in S,\ ||x - x^*|| < \delta. \tag{1.3}$$

The set S is typically defined by a number of **equality constraints**, $h_i(x) = 0,\ i = 1...m$ and **inequality constraints**, $g_i(x) \leq 0,\ i = 1...l$. The notation becomes

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = 0,\ i = 1...m \\ & g_i(x) \leq 0,\ i = 1...l \\ & x \in X, \end{aligned} \tag{1.4}$$

where X is usually some subspace of $\mathbb{R}^n$ and h_i and g_i as well as f are real-valued functions defined on $\mathbb{R}^n$.

3.1 Unconstrained Optimization

In the absence of any constraints h_i and g_i, the problem (1.4) is called unconstrained, i.e. simply the task of finding an x^* that ideally fulfills (1.2), but at least (1.3) over the whole of X, a subspace of $\mathbb{R}^n$.

Despite their apparent limitations, these problems are central to the field of optimization. Also many algorithms for solving constrained problems are extensions of methods for unconstrained problems. There are numerous algorithms for unconstrained optimization. In the one-dimensional case there are the sequential methods, such as Dichotomous, Golden-section and Fibonacci search [2]. These very simple yet powerful algorithms produce a sequence of decreasing intervals that converges to a local. They do demand an starting interval and can only find an optima within this interval. They do not make use of derivatives, a desirable property if f' is not readily available.

In multi-dimensional problems, gradient methods, such as steepest descent, use first order derivatives to find directions for which the objective function decrease and then perform one-dimensional line searches in these directions. The Newton method and the conjugate direction exploit the Hessian of the objective function to find search directions.

In some instances the step length can also be determined directly, eliminating the need for a line-search.

If f is differentiable, a necessary condition for optimality for unconstrained problems is the well known condition

$$\nabla f = 0. \tag{1.5}$$

3.2 Constrained Optimization

In this section we describe two important topics in constrained optimization, the concept of duality and dual problems and the Karush-Kuhn-Tucker conditions.

3.2.1 Karush-Kuhn-Tucker conditions

The Karush-Kuhn-Tucker (KKT) conditions provide necessary conditions for a local optima for a constrained optimization problem. They can be interpreted as the constrained equivalence to (1.5). The Lagrangian function associated with the constrained problem (1.4) is defined as

$$L(x;\lambda,\nu) = f(x) + \sum_{i=1}^{l} \lambda_i g_i(x) + \sum_{j=1}^{m} \nu_j h_j(x) \tag{1.6}$$

where $x \in X$ and $(\lambda,\nu) \in D = \{\nu \in \mathbb{R}^l,\ \lambda \in \mathbb{R}^m,\ \lambda \geq 0\}$. Obviously, for feasible x, λ and ν

$$L(x;\lambda,\nu) \leq f(x), \tag{1.7}$$

since $h_j(x) = 0$, $g_i(x) \leq 0$ and λ_i are non-negative. If x^* is an optimizer of (1.4) it can be shown that there must exist a $\lambda^* \geq 0$ such that $\sum_{i=1}^{l} \lambda_i^* g_i(x^*) = 0$. Thus

$$f(x^*) = L(x^*;\lambda^*,\nu^*) = f(x) + \sum_{i=1}^{l} \lambda_i g_i(x) + \sum_{j=1}^{m} \nu_j h_j(x) \leq f(x^*), \tag{1.8}$$

which implies that the gradient of $L(x;\lambda,\nu)$ with respect to x must be zero. Combining this with the feasibility constraints we get the **Karush-Kuhn-Tucker conditions**,

$$\nabla L(x^*;\lambda^*,\nu^*) = \nabla f(x) + \sum_{i=1}^{l} \lambda_i \nabla g_i(x) + \sum_{j=1}^{m} \nu_j \nabla h_j(x) = 0 \tag{1.9}$$

$$h_j(x^*) = 0,\ j = 1...m \tag{1.10}$$

$$g_i(x^*) \leq 0,\ i = 1...l \tag{1.11}$$

$$x^* \in X \tag{1.12}$$

$$\lambda_i \geq 0 \tag{1.13}$$

$$\sum_{i=1}^{l} \lambda_i^* g_i(x^*) = 0. \tag{1.14}$$

Here (1.11)-(1.14) are called the feasibility conditions and (1.14) the complementary slackness condition. One additional requirement for the KKT-conditions to hold is that the gradients of the constraints ∇h_j and ∇g_i are linearly independent at x^*, i.e.

$$\left(\sum_{i=1}^{l} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{m} \nu_j \nabla h_j(x^*) = 0 \Rightarrow \lambda, \nu = 0 \right), \tag{1.15}$$

also known as the constraint qualification.

The conditions that this theorem of Karush-Kuhn-Tucker provide are fundamental to optimization theory. Many of the existing algorithms for constrained optimization are based on these KKT-conditions. Such methods can interpreted as trying to identify points (x, λ, ν) that satisfy (1.10)-(1.14), instead of, for instance carrying out line-searches along descent directions,

3.2.2 Duality

Given a constrained optimization problem, on the form (1.4), there is another closely related optimization problem called the **Lagrangian dual problem**. The original problem is accordingly called the primal problem. One of the many remarkable properties of the dual problem is that it is an underestimator of its primal. Thus one can obtain a lower bound, and in certain instances even the exact minima to the original problem by solving its dual. This field of Lagrangian duality matured in the 1950's and led to a flood of new results that produced countless new optimization algorithms, including some of the most successful ones in use today.

The Lagrangian dual function of (1.4) is defined as

$$\theta(\lambda,\nu) = \inf_{x \in S} L(x;\lambda,\nu). \tag{1.16}$$

Clearly,

$$\theta(\lambda, \nu) \leq L(x; \lambda, \nu) \leq f(x), \tag{1.17}$$

for any $x \in S$ and $\lambda \geq 0$.

The Lagrangian dual is defined as

$$\begin{aligned} \max \quad & \theta(\lambda, \nu) \\ \text{s.t.} \quad & \lambda \geq 0 \end{aligned} \tag{1.18}$$

If λ^*, ν^* maximize (1.18) the inequality (1.17) still holds and we have that

$$\theta(\lambda^*, \nu^*) \leq f(x^*). \tag{1.19}$$

The difference between the maximum of the dual problem and the minima of the primal problem is called the **duality gap**, if the duality gap is zero we speak of **strong duality**. In general, there is a duality gap between the dual and primal problem, but there are instances when strong duality holds.

Another remarkable property of the Lagrangian dual function is that it is concave, regardless of what the primal problem looks like. Furthermore, the feasible set of the dual problem is the intersection of a number of halfspaces, $\lambda_i \geq 0$. Thus solving the dual problem implies maximizing a concave function over a convex set. This is equivalent to a convex optimization problem, a topic we will discuss in the next section.

3.2.3 Convex Optimization Problems

Convex optimization problems are a class of problems of special interest to the optimization community, it is defined as the minimization of a convex function over a convex set. It has long been known that for such problems any local minima is a global minima, the set of global minima is a convex set and that if the objective function is strictly convex the minima is unique. In optimization convexity it is actually a more important property than linearity or non-linearity. Despite this it is only recently that it has become a central tool in engineering, this can be attributed to recent breakthroughs in convex optimization algorithms, most notably the development of the interior point methods [13].

A function f is convex if, for any x and y and $0 \leq t \leq 1$

$$f(tx + (1-t)y) \geq tf(x) + (1-t)f(y), \tag{1.20}$$

see figure 1.3. The function is called concave if $-f$ is convex.

A set C is called convex if the line segment between any two points in C is contained in C. That is, if for any $x, y \in C$ and $0 \leq t \leq 1$

$$tx + (1-t)y \in C, \tag{1.21}$$

examples of convex and non-convex sets can be seen in figure 1.4.

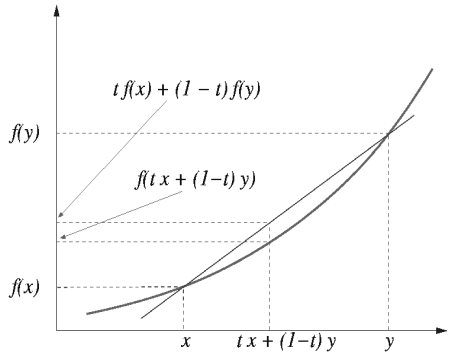


Figure 1.3: A convex function.

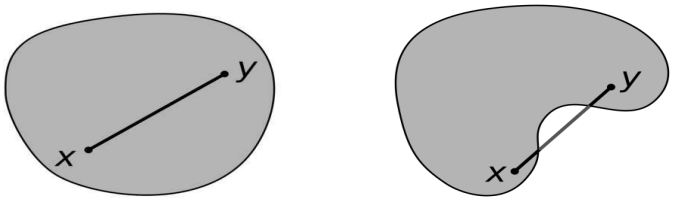


Figure 1.4: A convex set (left) and a non-convex set (right).

Under some mild assumptions, strong duality always hold for convex problems. This means that one can solve convex optimization problem implicitly by maximizing its corresponding dual problem. It also implies that any point that satisfies the KKT conditions will give us the minimizer.

Certain commonly occurring convex optimization problems are

- **Linear program** - minimizing a linear function over the polytope

$$\left\{a_i^T x \leq 0,\ i = 1...n\right\}.$$

- **(Convex) Quadratic program** - minimizing a convex quadratic function f over over a polytope

$$f(x) = x^T H x,\ H \succeq 0.$$

- **Second-order cone program** - minimizing a linear function over the second-order cone
$$\left\{||A_i x - b_i||_2 \leq c_i^T x + d_i,\ i = 1...m\right\}.$$
- **Semidefinite program** - minimizing a linear function over the intersection of the cone of positive semidefinite matrices and an affine subspace
$$\{X \succeq 0\} \bigcap \{Tr(A_i X) = b_i,\ i = 1...m\}.$$

For a more detailed description of convex optimization we refer to [16].

4 Summary of the papers

PAPER I — Normalized Cuts Revisited: A Reformulation for Segmentation with Linear Grouping Constraints.

Indisputably Normalized Cuts is one of the most popular segmentation algorithms in 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, 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 suggested methods is that they are noticeably limited 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 all types of 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 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 with convincing results.

Author contribution: This paper was a result of a collaboration between myself, Carl Olsson and Fredrik Kahl. I acted as the primary author with the solid support of my coauthors. I carried out the experiments and devised the reformulation of the normalized cut problem. Carl Olsson also contributed with the proof in section 3.

PAPER II — Efficiently Solving the Fractional Trust Region Problem

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.

Author contribution: This paper was a continuation of paper I, with a similar work distribution. I again acted as the primary author and also carried out the experiments. Carl Olsson contributed with the experimental validation on the artificial problems as well as choosing the formulation for the second-order derivatives.

PAPER III — Improved Spectral Relaxation Methods for Binary Quadratic Optimization Problems

In this paper we introduce two new methods for solving binary quadratic problems. While spectral relaxation methods have been the workhorse subroutine for a wide variety of computer vision problems - segmentation, clustering, subgraph matching to name a few - it has recently been challenged by semidefinite programming (SDP) relaxations. In fact, it can be shown that SDP relaxations produce better lower bounds than spectral relaxations on binary problems with a quadratic objective function. On the other hand, the computational complexity for SDP increases rapidly as the number of decision variables grows making them inapplicable to large scale problems.

Our methods combine the merits of both spectral and SDP relaxations - better (lower) bounds than traditional spectral methods and considerably faster execution times than SDP. The first method is based on spectral subgradients and can be applied to large scale SDPs with binary decision variables and the second one is based on the trust region problem. Both algorithms have been applied to several large scale vision problems with good performance.

Author contribution: This paper was the result of merging papers [7] and [6]. Carl Olsson acted as primary author of the former and I on the latter, with the constant support of Fredrik Kahl.

PAPER IV — Bijective Thin-Plate Splines

Thin-plate splines are a class of widely used non-rigid spline mapping functions. It is a natural choice of interpolating function in two dimensions and has been a commonly used tool for over a decade. Introduced and developed by Duchon [5] and Meinguet [12] and popularized by Bookstein [3], its attractions include an elegant mathematical formulation along with a very natural and intuitive physical interpretation.

Consider a thin metal plate extending to infinity in all directions. At a finite number of discrete positions $\mathbf{t}_i \in \mathbb{R}^2$, $i = 1...n$, the plate is at fixed heights z_i. The metal plate will take then the form that minimizes its *bending energy*. In two dimensions the bending energy of a plate described by a function $g(x, y)$ is proportional to

$$J(g) = \int\int_{\mathbb{R}^2} \left(\left(\frac{\partial^2 g}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 g}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 g}{\partial y^2} \right)^2 \right) dxdy. \tag{1.22}$$

Consequently, the metal plate will be described by the function that minimizes (1.22) under the point constraints $g(\mathbf{t}_i) = z_i$. It was proven Duchon [5] that if such a function exists it is unique.

The thin-plate spline framework can also be employed in a deformation setting, that is mappings from $\mathbb{R}^m$ to $\mathbb{R}^m$. This is accomplished by the combination of several thin-plate spline interpolants. Here we restrict ourselves to $m = 2$. If instead of understanding the displacement of the thin metal plate as occurring orthogonally to the (x_1, x_2)-plane view them as displacements of the x_1- or x_2- position of the point constraints. With this interpretation, a new function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ can be constructed from two thin-plate splines, each describing the x_1- and x_2-displacements respectively.

In spite of its appealing algebraic formulation, thin-plate spline mappings do have drawbacks and, disregarding computational and numerical issues, one in particular. Namely, bijectivity is never assured. In computer vision, non-linear mappings in $\mathbb{R}^2$ of this sort are frequently used to model deformations in images. The basic assumption is that all the images contain similar structures and therefore there should exist mappings between pairs of images that are both one-to-one and onto. Hence bijective mappings are of interest.

This work is an attempt at characterizing the set of bijective thin-plate spline mappings. It contains a formulation of how to describe this set, as well as proofs of many of its properties. It also includes a discussion of some experimentally derived indications of other attributes of this set, such as boundedness and convexity, as well as methods for finding sufficient conditions for bijectivity.

Author contribution: Papers IV-VI were all the result of work carried out by myself and my advisor Karl Åström. I acted as first author and carried out the experiments in all three instances.

PAPER V — Image Registration using Thin-Plate Splines

Image registration is the process of geometrically aligning two or more images. In this paper we describe a method for registering pairs of images based on thin-plate spline mappings. The proposed algorithm minimizes the difference in gray-level intensity over bijective deformations. By using quadratic sufficient constraints for bijectivity and a least squares formulation this optimization problem can be addressed using quadratic programming and a modified Gauss-Newton method. This approach also results in a very

computationally efficient algorithm. Example results from the algorithm on three different types of images are also presented.

PAPER VI — Groupwise Image Registration and Automatic Active Appearance Model Generation

This section is concerned with groupwise image registration, the simultaneous alignment of a large number of images. As opposed to pairwise registration the choice of reference image is not equally obvious, therefore an alternate approach must be taken.

Groupwise registration has received equivalent amounts of attention from the research community as pairwise registration. It has been especially addressed in shape analysis under the name Procrustes analysis.The areas of application are still remote sensing, medical imaging and computer vision, but now the aggregation of images allows for a greater understanding of their underlying distribution.

The focus here is towards a specific task, the use of image registration to automatically construct deformable models for image analysis.

Bibliography

[1] S. Agarwal, M.K. Chandraker, F. Kahl, D.J. Kriegman, and S. Belongie. Practical global optimization for multiview geometry. In *Proc. 4th European Conf. on Computer Vision, Graz, Austria*, pages I: 592–605, 2006.

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

[3] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11, 1989.

[4] 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.

[5] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. *Constructive Theory of Functions of Several Variables*, 1987.

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

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

[8] R. I. Hartley and F. Schaffalitzky. L-∞ minimization in geometric reconstruction problems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC*, June 2004.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[10] F. Kahl. Multiple view geometry and the l_∞-norm. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1002–1009, Washington, DC, USA, 2005.

[11] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. pages 61–62, 1987.

[12] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics*, 30, 1979.

[13] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

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

[15] H. Stewenius, F. Schaffalitzky, and D. Nister. How hard is 3-view triangulation really? In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 686–693, Washington, DC, USA, 2005.

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

PAPER I

In *Proceedings International Conference on Computer Vision,*
Rio de Janeiro, Brazil 2007.

Main Entry: nor · mal · ize
Pronunciation: \'nȯr-mə-,līz \
Function: transitive verb
Inflected Form(s): nor · mal · ized; nor · mal · iz · ing
Origin: 1520-1530; from Latin *normalis*, "in conformity with rule, normal"; from from *norma*, "rule, pattern" literally "carpenter's square" (see norm).

1 : to make conform to or reduce to a norm or standard
2 : to make normal (as by a transformation of variables)
3 : to bring or restore (as relations between countries) to a normal condition

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

Anders P. Eriksson, Carl Olsson and Fredrik Kahl

Centre for Mathematical Sciences
Lund University, Sweden

Abstract

Indisputably Normalized Cuts is one of the most popular segmentation algorithms in 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, 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 suggested methods is that they are noticeably limited 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 all types of 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 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 with convincing results.

1 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. 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, 2].

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 based were made popular first through the Normalized Cut formulation of [9] and more recently by the energy minimization method of [3]. This algorithm for optimizing objective functions that are submodular has the property of solving many discrete problems exactly. However, not all segmentation problems can be formulated with submodular objective functions, nor is it possible to incorporate all types of linear 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. It is not the aim of this paper to argue the merits of one method, or cut metric, over another, nor do we here concern ourselves with how the actual grouping constraints are obtained. Instead we will show how through Lagrangian relaxation one in a unified can handle such linear constrains and also in what way they influence the resulting segmentation.

1.1 Problem Formulation

Consider an undirected graph $\mathbf{G}$, with nodes $\mathbf{V}$ and edges $\mathbf{E}$ 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

$$cut(A, V) = \sum_{\substack{i \in A \\ j \in V \setminus A}} w_{ij} \tag{1.1}$$

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 [9]. 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

$$N_{cut} = \frac{cut(A, V)}{assoc(A, V)} + \frac{cut(B, V)}{assoc(B, V)}, \tag{1.2}$$

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 [9] that by relaxing (1.2) a continuous underestimator of the Normalized Cut can be efficiently computed. These techniques are then extended in [11] beyond graph bipartitioning to include multiple segments, and even further in [12] to handle certain types of linear equality constraints.

One can argue that the drawbacks of this, the classical formulation, for solving 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. Furthermore, the set of grouping constraints is also very limited, only homogeneous linear equality constraints can be included in the existing theory. We will show that this excludes many visually relevant constraints. In [4] an attempt is made at solving a similar problem with general linear constraints. This approach does however effectively involve dropping any discrete constraint all together, leaving one to question the quality of the obtained solution.

2 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 (1.2), the definition of Normalized Cuts, the cost of partitioning an image with affinity matrix W into two disjoint sets, A and B, can be written as

$$N_{cut} = \frac{\sum_{\substack{i \in A \\ j \in B}} w_{ij}}{\sum_{\substack{i \in A \\ j \in V}} w_{ij}} + \frac{\sum_{\substack{i \in B \\ j \in A}} w_{ij}}{\sum_{\substack{i \in B \\ j \in V}} w_{ij}}. \tag{1.3}$$

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 (1.3) as

$$\begin{aligned} N_{cut} &= \frac{\sum_{i,j} w_{ij}(z_i - z_j)^2}{2\sum_i (z_i+1)d_i} + \frac{\sum_{i,j} w_{ij}(z_i - z_j)^2}{2\sum_i (z_i-1)d_i} = \\ &= \frac{z^T(D-W)z}{2d^T(z+1)} + \frac{z^T(D-W)z}{2d^T(z-1)} = \\ &= \frac{\big(z^T(D-W)z\big)d^T 1}{1^T d d^T 1 - z^T d^T d^T z} = \frac{\big(z^T(D-W)z\big)d^T 1}{z^T((1^T d)D - dd^T)z}. \end{aligned} \tag{1.4}$$

In the last inequality we used the fact that $1^T d = z^T D z$. 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

$$\begin{aligned} \inf_{z} \quad & \tfrac{z^T(D-W)z}{z^T((1^Td)D-dd^T)z} \\ \text{s.t.} \quad & z \in \{-1,1\}^n \\ & Cz = b. \end{aligned} \tag{1.5}$$

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

$$\begin{aligned} \inf_{z} \quad & \tfrac{z^T(D-W)z}{z^T((1^Td)D-dd^T)z} \\ \text{s.t.} \quad & z^Tz = n \\ & Cz = b. \end{aligned} \tag{1.6}$$

This is also a non-convex problem, however we shall see in section 3 that we are able to solve this problem exactly. Next we will write problem (1.6) 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

$$L = \begin{bmatrix} (D-W) & 0 \\ 0 & 0 \end{bmatrix}, \quad M = \begin{bmatrix} ((1^Td)D-dd^T) & 0 \\ 0 & 0 \end{bmatrix}, \tag{1.7}$$

and

$$\hat{C} = [C \quad -b] \tag{1.8}$$

the homogenized constraint matrix. The relaxed problem (1.6) can now be written

$$\begin{aligned} \inf_{z} \quad & \tfrac{[z^T\ 1]L\left[\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right]}{[z^T\ 1]M\left[\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right]} \\ \text{s.t.} \quad & z^Tz = n \\ & \hat{C}\left[\begin{smallmatrix} z \\ 1 \end{smallmatrix}\right] = 0. \end{aligned} \tag{1.9}$$

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 (1.6) in its homogenized form is

$$\begin{aligned} \inf_{\hat{z}} \quad & \tfrac{\hat{z}^TL\hat{z}}{\hat{z}^TM\hat{z}} \\ \text{s.t.} \quad & \hat{z}_{n+1}^2 - 1 = 0 \\ & \hat{z}^T\hat{z} = n+1 \\ & \hat{C}\hat{z} = 0. \end{aligned} \tag{1.10}$$

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 that basis such that $\hat{y}_{k+1} = \hat{z}_{n+1} = 1$. We put $\hat{L} = N_{\hat{C}}^T L N_{\hat{C}}$, $\hat{M} = N_{\hat{C}}^T M N_{\hat{C}}$. In the new space we get the following formulation

$$\begin{aligned} \inf_{\hat{y}} \quad & f(\hat{y}) = \tfrac{\hat{y}^T \hat{L} \hat{y}}{\hat{y}^T \hat{M} \hat{y}} \\ \text{s.t.} \quad & \hat{y}_{k+1}^2 - 1 = 0 \\ & \hat{y}^T N_{\hat{C}}^T N_{\hat{C}} \hat{y} = ||\hat{y}||_{N_{\hat{C}}}^2 = n+1. \end{aligned} \tag{1.11}$$

A common approach to 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.

3 Lagrangian Relaxation and Strong Duality

In this section we will show how to solve (1.6) using Lagrange duality. To do this we start by generalizing a lemma from [7] for trust region problems

Lemma 1. *If there exists a y with $y^T A_3 y + 2b_3^T y + c_3 < 0$, then, assuming the existence of a minima, the primal problem*

$$\inf_y \frac{y^T A_1 y + 2b_1^T y + c_1}{y^T A_2 y + 2b_2^T y + c_2}, \; s.t \, y^T A_3 y + 2b_3^T y + c_3 \leq 0 \tag{1.12}$$

and the dual problem

$$\sup_{\lambda \geq 0} \inf_y \frac{y^T (A_1 + \lambda A_3) y + (b_1 + \lambda b_3)^T y + c_1 + \lambda c_3}{y^T A_2 y + 2b_2^T y + c_2} \tag{1.13}$$

has no duality gap.

Proof. The primal problem can be written as

$$\begin{aligned} \inf \quad & \gamma_1 \\ \text{s.t} \quad & y^T (A_1 - \gamma_1 A_2) y + 2(b_1 - \gamma_1 b_2)^T y + c_1 - \gamma_1 c_2 \leq 0 \\ & y^T A_3 y + 2b_3^T y + c_3 \leq 0 \end{aligned} \tag{1.14}$$

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

$$\mathcal{M}(\lambda,\gamma)=\left[\begin{array}{cc} A_1+\lambda A_3-\gamma A_2 & b_1+\lambda b_3-\gamma b_2 \\ (b_1+\lambda b_3-\gamma b_2)^T & c_1+\lambda c_3-\gamma c_2 \end{array}\right]. \tag{1.15}$$

The dual problem can be written

$$\begin{array}{ll} \sup_{\lambda\geq 0}\inf_{\gamma_2,y} & \gamma_2 \\ \text{s.t} & \left[\begin{array}{c} y \\ 1 \end{array}\right]^T \mathcal{M}(\lambda,\gamma_2)\left[\begin{array}{c} y \\ 1 \end{array}\right]\leq 0. \end{array} \tag{1.16}$$

Since (1.16) is dual to (1.14) we have that for their optimal values, $\gamma_2^*\leq\gamma_1^*$ 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

$$\begin{array}{ll} \sup_{\gamma_3,\lambda\geq 0} & \gamma_3 \\ \text{s.t} & \mathcal{M}(\lambda,\gamma_3)\succeq 0. \end{array} \tag{1.17}$$

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

$$\left[\begin{array}{c} y \\ 1 \end{array}\right]^T \mathcal{M}(\lambda,\gamma_3)\left[\begin{array}{c} y \\ 1 \end{array}\right]+\epsilon\leq 0 \tag{1.18}$$

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

$$\begin{array}{c} y^TA_3y+2b_3^Ty+c_3\leq 0\Rightarrow \\ y^T(A_1-\gamma A_2)y+2(b_1-\gamma b_2)^Ty+c_1-\gamma c_2\geq 0. \end{array} \tag{1.19}$$

However according to the S-procedure [1] this is true if and only if there exists $\lambda\geq 0$ such that $\mathcal{M}(\lambda,\gamma)\succeq 0$. Therefore (γ,λ) is feasible for problem (1.17) and thus $\gamma_3=\gamma_1$. □

We note that for a fixed γ the problem

$$\begin{array}{ll} \inf_y & y^T(A_1-\gamma A_2)y+2(b_1-\gamma b_2)^Ty+c_1-\gamma c_2 \\ \text{s.t.} & y^TA_3y+2b_3^Ty+c_3\leq 0 \end{array} \tag{1.20}$$

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

Corollary 1. *Let A_3 be positive semidefinite. If there exists a y with $y^TA_3y+2b_3^Ty+c_3<0$, then the primal problem*

$$\inf_y \frac{y^TA_1y+2b_1^Ty+c_1}{y^TA_2y+2b_2^Ty+c_2},\ \textit{s.t.}\ y^TA_3y+2b_3^Ty+c_3=0 \tag{1.21}$$

and the dual problem

$$\sup_{\lambda}\inf_{y}\frac{y^T(A_1+\lambda A_3)y+(b_1+\lambda b_3)^Ty+c_1+\lambda c_3}{y^TA_2y+2b_2^Ty+c_2} \tag{1.22}$$

has no duality gap, (once again assuming that a minima exists for the primal problem).

Next we will show how to solve a problem on a form related to (1.11). Let

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

Theorem 3.1. *Assuming the existence of a minima, if $\mathbf{A}_3$ is positive definite, then the primal problem*

$$\begin{aligned}\inf_{y^TA_3y+2b_3^Ty+c_3=n+1}\frac{y^TA_1y+2b_1^Ty+c_1}{y^TA_2y+2b_2^Ty+c_2}&=\\ =\inf_{\substack{\hat{y}^T\mathbf{A}_3\hat{y}=n+1\\ y_{n+1}^2=1}}\frac{\hat{y}^T\mathbf{A}_1\hat{y}}{\hat{y}^T\mathbf{A}_2\hat{y}}&\end{aligned} \tag{1.23}$$

and its dual

$$\sup_{t}\inf_{\hat{y}^T\mathbf{A}_3\hat{y}=n+1}\frac{\hat{y}^T\mathbf{A}_1\hat{y}+ty_{n+1}^2-t}{\hat{y}^T\mathbf{A}_2\hat{y}} \tag{1.24}$$

has no duality gap.

Proof. Let γ^* be the optimal value of problem (1.11). Then

$$\begin{aligned}\gamma^*&=\inf_{\substack{\hat{y}^T\mathbf{A}_3\hat{y}=n+1\\ y_{n+1}^2=1}}\frac{\hat{y}^T\mathbf{A}_1\hat{y}}{\hat{y}^T\mathbf{A}_2\hat{y}}\\ &=\sup_t\inf_{\substack{\hat{y}^T\mathbf{A}_3\hat{y}=n+1\\ y_{n+1}^2=1}}\frac{\hat{y}^T\mathbf{A}_1\hat{y}+ty_{n+1}^2-t}{\hat{y}^T\mathbf{A}_2\hat{y}}\\ &\geq\sup_t\inf_{\hat{y}^T\mathbf{A}_3\hat{y}=n+1}\frac{\hat{y}^T\mathbf{A}_1\hat{y}+ty_{n+1}^2-t}{\hat{y}^T\mathbf{A}_2\hat{y}}\\ &\geq\sup_{t,\lambda}\inf_{\hat{y}}\frac{\hat{y}^T\mathbf{A}_1\hat{y}+ty_{n+1}^2-t+\lambda(\hat{y}^T\mathbf{A}_3\hat{y}-(n+1))}{\hat{y}^T\mathbf{A}_2\hat{y}}\\ &=\sup_{s,\lambda}\inf_{\hat{y}}\\ &\frac{\hat{y}^T\mathbf{A}_1\hat{y}+sy_{n+1}^2-s+\lambda(y^TA_3y+y_{n+1}2b_3^Ty+c_3-(n+1))}{\hat{y}^T\mathbf{A}_2\hat{y}}=\\ &=\sup_\lambda\inf_{y_{n+1}^2=1}\frac{\hat{y}^T\mathbf{A}_1\hat{y}+\lambda(y^TA_3y+2b_3^Ty+c_3-(n+1))}{\hat{y}^T\mathbf{A}_2\hat{y}}\\ &=\sup_\lambda\inf_y\frac{y^TA_1y+2b_1^Ty+c_1+\lambda(y^TA_3y+2b_3^Ty+c_3-(n+1))}{y^TA_2y+2b_2^Ty+c_2}\\ &=\gamma^*.\end{aligned} \tag{1.25}$$

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

$$\begin{aligned}\mu^* &= \sup_t \inf_{\hat{y}^T\mathbf{A}_3\hat{y}=n+1} \frac{\hat{y}^T\mathbf{A}_1\hat{y} + ty_{n+1}^2 - t}{\hat{y}^T\mathbf{A}_2\hat{y}} = \\ &= \sup_t \inf_{\hat{y}^T\mathbf{A}_3\hat{y}=n+1} \frac{\hat{y}^T\mathbf{A}_1\hat{y} + ty_{n+1}^2 - t\frac{\hat{y}^T\mathbf{A}_3\hat{y}}{n+1}}{\hat{y}^T\mathbf{A}_2\hat{y}} = \\ &= \sup_t \inf_{\hat{y}^T\mathbf{A}_3\hat{y}=n+1} \frac{\hat{y}^T\left(\mathbf{A}_1 + t\left(\left[\begin{smallmatrix}0 & 0\\ 0 & 1\end{smallmatrix}\right] - \frac{\mathbf{A}_3}{n+1}\right)\right)\hat{y}}{\hat{y}^T\mathbf{A}_2\hat{y}}\end{aligned} \tag{1.26}$$

□

Finally, since strong duality holds, we can state the following corollary. [1].

Corollary 2. *If t^* and $\hat{y}^*$ solves* (1.26)*, 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* (1.12)*.*

4 The Dual Problem and Constrained Normalized Cuts

Returning to our relaxed problem (1.11) we start off by introducing the following lemma.

Lemma 2. *L and M are both $(n+1)\times(n+1)$ positive semidefinite matrices of rank $n-1$, both their nullspaces are spanned by $n_1 = \left[\begin{smallmatrix}1 & \dots & 1 & 0\end{smallmatrix}\right]^T$ and $n_2 = \left[\begin{smallmatrix}0 & \dots & 0 & 1\end{smallmatrix}\right]^T$. Consequently, $L_{\hat{C}}$ and $M_{\hat{C}}$ 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 $(1^Td)D - dd^T$ is p.s.d.

$$\begin{aligned}v^T((1^Td)D - dd^T)v &= \textstyle\sum_i d_i \sum_j d_j v_j^2 - \left(\sum_i d_i v_i\right)^2 \\ &= \textstyle\sum_{i,j} d_i d_j v_j(v_j - v_i) = \sum_i d_i d_i v_i(v_i - v_i) + \\ &+ \textstyle\sum_{i,j<i} d_i d_j v_j(v_j - v_i) + d_j d_i v_i(v_i - v_j) = \\ &\textstyle\sum_{i,j<i} d_i d_j (v_j - v_i)^2 \geq 0,\ \forall v \in \mathbb{R}^n\end{aligned} \tag{1.27}$$

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

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

Next, since

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

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

Assuming that the original problem is feasible then we have that, as $f(\hat{y})$ of problem (1.23) is the quotient of two positive semidefinite quadratic forms and is therefore $f(\hat{y})$ non-negative, a minima for the relaxed Normalized Cut problem will exist. Theorem 3.1 states that strong duality holds for a program on the form (1.23), if a minima exists. Consequently, we can apply the theory from the previous section directly and solve (1.11) through its dual formulation. Let

$$E_{\hat{C}} = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right] - \frac{N_{\hat{C}}^T N_{\hat{C}}}{n+1} = N_{\hat{C}}^T \left[\begin{smallmatrix} -\frac{I}{n+1} & 0 \\ 0 & 1 \end{smallmatrix}\right] N_{\hat{C}} \tag{1.28}$$

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

$$\sup_t \inf_{||\hat{y}||^2_{N_{\hat{C}}}=n+1} \theta(\hat{y}, t) = \frac{\hat{y}^T(\hat{L} + tE_{\hat{C}})\hat{y}}{\hat{y}^T \hat{M} \hat{y}}. \tag{1.29}$$

The inner minimization is the well known generalized Rayleigh quotient, for which the minima is given by the algebraically smallest generalized eigenvalue[1] of $(L_{\hat{C}} + tE_{\hat{C}})$ and $M_{\hat{C}}$. Letting $\lambda^G_{\min}(t)$ and $v^G_{\min}(t)$, denote the smallest generalized eigenvalue and corresponding generalized eigenvector of $(L_{\hat{C}} + tE_{\hat{C}})$ and $\hat{M}$ we can write problem (1.29) as

$$\sup_t \lambda^G_{\min}(\hat{L} + tE_{\hat{C}}, \hat{M}). \tag{1.30}$$

It can easily be shown that the minimizer of the inner problem of (1.29) for some t, is given by a scaling of the generalized eigenvector, $\hat{y}(t) = \frac{\sqrt{n+1}}{||v^G_{\min}(t)||^2_{N_{\hat{C}}})} v^G_{\min}(t)$. The relaxed Normalized Cut problem can thus be solved by finding the maxima of (1.30). 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 (1.30) 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 in this case. This can,

[1] 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.

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

$$\begin{aligned}
\hat{M} = \left[\begin{smallmatrix} c & c_0 \\ I & \end{smallmatrix}\right]^T \left(\begin{bmatrix} ((1^T d)D - dd^T) & 0 \\ 0 & 0 \end{bmatrix} \right) \left[\begin{smallmatrix} c & c_0 \\ I & \end{smallmatrix}\right] = \\
= \left\{ \begin{smallmatrix} D := \left[\begin{smallmatrix} D_1 & 0 \\ 0 & D_2 \end{smallmatrix}\right] \\ d := \left[\begin{smallmatrix} d_1 \\ d_2 \end{smallmatrix}\right] \end{smallmatrix} \right\} = \underbrace{\begin{bmatrix} D_2 & 0 \\ 0 & c_0^T D_1 c_0 + 1 \end{bmatrix}}_{\tilde{D}} + \\
+ \underbrace{\begin{bmatrix} c^T & cd_1 + d_2 & 0 \\ c_0^T & c_0^T d_1 & 1 \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} D_1 & & \\ & 1 & \\ & & -1 \end{bmatrix}}_{S} \begin{bmatrix} c & c_0 \\ d_1^T c^T + d_2^T & d_1^T c_0 \\ 0 & 1 \end{bmatrix} = \\
= \tilde{D} + VSV^T. \qquad (1.31)
\end{aligned}$$

Hence, $\hat{M}$ 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 [5] we can express the inverse of $\hat{M}$ as

$$\begin{aligned}
\hat{M}^{-1} = (\tilde{D} + VSV^T)^{-1} = \\
= \tilde{D}^{-1} \left(I - V(S^{-1} + V^T \tilde{D}^{-1} V)^{-1} V \tilde{D}^{-1} \right). \qquad (1.32)
\end{aligned}$$

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

$$\begin{aligned}
\sup_t \quad \lambda_{\min} \big((\tilde{D}^{-1}(I - V(S^{-1} + V^T \quad \tilde{D}^{-1} V)^{-1} V \tilde{D}^{-1}) \\
{N_{\hat{C}}}^T (\hat{L} + tE_{\hat{C}}) N_{\hat{C}} \big). \qquad (1.33)
\end{aligned}$$

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 $\hat{M}^{-1}(\hat{L} + tE_{\hat{C}})$ is still dense, it is the product and sum of diagonal ($\tilde{D}^{-1}$, $E_{\hat{C}}$), sparse ($\hat{L}$, $N_{\hat{C}}$) and low rank matrices (V, S^{-1}). It is a very structured matrix to which iterative eigensolvers can successfully be applied.

In certain cases it might however occur that the quadratic form in the denominator is only positive semidefinite and thus singular. These instances are easily detected and must be treated specially. As we then can not invert $\hat{M}$ and rewrite the problem as a standard eigenvalue problem we must instead work with generalized eigenvalues, as defined in (1.30). 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 $\hat{L}+tE$ and $\hat{M}$, do however exist [10]. Note, that the absence of linear constraints is such a special instance. However, in that case homogenization is completely unnecessary, and (1.6) 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 $\hat{y}^*$ are the optimizers of (1.29), (and consequently also (1.30)), 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 $\hat{z}_{n+1} = \hat{y}_{k+1}$, we have that z^* prior to rounding is the minimizer of (1.6). 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 (1.5). This is typically carried out by applying some rounding scheme to the solution.

4.1 Multi-Class Constrained Normalized Cuts

Multi-class Normalized Cuts is a generalization of (1.2) for an arbitrary number of partitions.

$$N_{cut}^k = \sum_{l=1}^{k} \frac{cut(A_l, V)}{assoc(A_l, V)}. \tag{1.34}$$

If one minimizes (1.34) 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, [3]. The associated subproblem at each iteration then becomes

$$\begin{aligned} \tilde{N}_{cut}^k &= \frac{cut(A_i, V)}{assoc(A_i, V)} + \\ &+ \frac{cut(A_j, V)}{assoc(A_j, V)} + \sum_{l \neq i,j} \frac{cut(A_l, V)}{assoc(A_l, V)} = \\ &\frac{cut(A_i, V)}{assoc(A_i, V)} + \frac{cut(A_j, V)}{assoc(A_j, V)} + c, \end{aligned} \tag{1.35}$$

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

$$\begin{aligned} \inf_z \quad & f(z) = \frac{z^T(D-W)z}{-z^T dd^T z + (1^T d)^2} \\ \text{s.t.} \quad & z \in \{-1, 1\}^n \\ & Cz = b, \end{aligned} \tag{1.36}$$

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 fulfills all constraints. There is however no guarantee that this method will avoid getting trapped in local minima and producing a sub-optimal solution, but during the experimental validation this procedure always produced satisfactory results.

5 Experimental Validation

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-homogenous, 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 [6]. 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, figure 1.1. Both approaches produce similar results, suggesting that in the absence of constraints

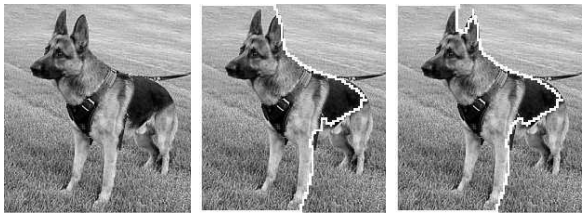


Figure 1.1: Original image (left), standard Normalized Cut algorithm (middle) and the reformulated Normalized Cut algorithm with no constraints (right).

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 hardcoding of some pixels, i.e. pixel i should belong to a certain class. This can be expressed as the linear constraints $z_i = \pm 1$, $i = 1...m$. In figure 1.2 it can be seen how a number of such hard constraints influences the segmentation of the image in figure 1.1.

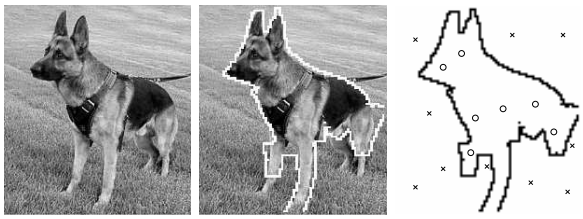


Figure 1.2: Original image (left), segmentation with constraints (middle) and constraints applied (right).

Another visually significant prior is the size or area of the resulting segments, that is constraints such as $\sum_i z_i = 1^T z = a$. The impact of enforcing limitations on the size of the partitions is shown in figure 1.3.

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 figure 1.4.

Finally, we also performed a multi-class segmentation with linear constraints, figure 1.5.

We argue that these results, not only indicate a satisfactory performance of the suggested method, but also illustrates the relevance of linear grouping constraints in image segmentation and the impact that they can have on the resulting partitioning. These experiments also seem to indicate that even a simple rounding scheme as the one used here can often suffice. As we threshold at zero, hard, including and excluding constraints are all ensured to hold after discretizing. Only the area constraints are not guaranteed to hold, however probably since the relaxed solution has the correct area, thresholding it typically produces a discrete solution with roughly the correct area.

6 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

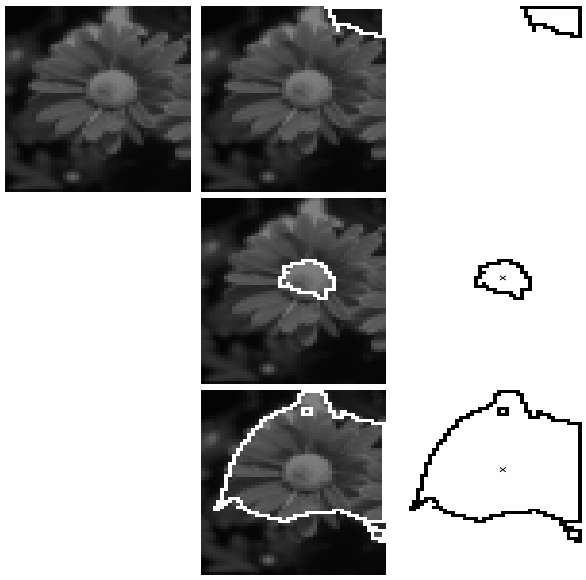


Figure 1.3: Original image (top left), segmentation without constraints (top middle) and segmentation boundary and constraints applied (top right). Segmentation with area constraints, (area=100 pixels) (middle left), segmentation boundary and constraints applied (middle right). Segmentation with area constraints, (area=2000 pixels) (bottom left), segmentation boundary and constraints applied (bottom right).

for very large scale problems, has also been offered. A number of experiments as well as 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^*. Another issue that was left open in this work is regarding the rounding scheme. The relaxed solution z is currently discretized 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 ensures that the linear constraints hold can easily be devised. We felt that an in-depth discussion on different procedures for discretization was outside the scope of this 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 and the final result.

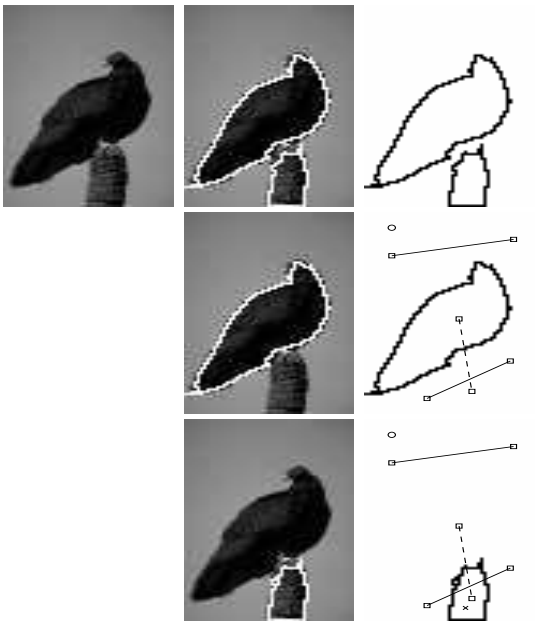


Figure 1.4: Original image (top left), segmentation without constraints (top middle), segmentation boundary and constraints applied (top right). Segmentation with hard, including and excluding, as well as area constraints, (area=25% of the entire image) (middle left), segmentation boundary and constraints 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 indicate an including constraint, and a dashed line an excluding.

Acknowledgments

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 programme Vision in Cognitive Systems II (VISCOS II), Swedish Research Council through grants no. 2004-4579 'Image-Based Localisation and Recognition of Scenes' and no. 2005-3230 'Geometry of multi-camera systems'.

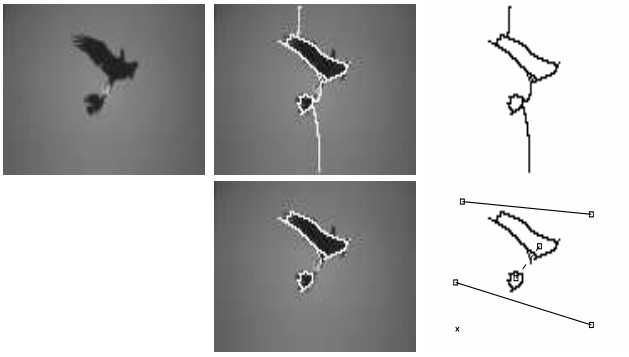


Figure 1.5: Original image (top left), three-class segmentation without constraints (top middle), segmentation boundary (top right). Three-class segmentation with hard, including and excluding constraints (bottom left), segmentation boundary and constraints applied (bottom right).

Bibliography

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

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

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

[10] 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.

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

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

PAPER II

In *Proceedings Asian Conference on Computer Vision,*
Tokyo, Japan 2007.

Main Entry: ef · fi · cient
Pronunciation: \i-'fi-shənt\
Function: adjective
Origin: 1350-1400; from Latin *efficientem* "making"; present participle of *efficere* "work out, accomplish" (see effect). Meaning "productive, skilled" is from 1787.

1: being or involving the immediate agent in producing an effect <*the efficient action of heat in changing water to steam*>
2: productive of desired effects; especially : productive without waste <*an efficient worker*>

Efficiently Solving the Fractional Trust Region Problem

Anders 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, piecewise 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 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, [9, 3].

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 [10] and more recently by the energy minimization method of [2]. This algorithm for optimizing objective functions that are submodular has the property of that it solves many discrete problems exactly. However, not all segmentation problems can 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 $\mathbf{G}$, with nodes $\mathbf{V}$ and edges $E_{\hat{C}}$ 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

$$cut(A, V) = \sum_{i \in A,\ j \in V \setminus A} w_{ij}. \tag{1.1}$$

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 [10]. 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

$$N_{cut} = \frac{cut(A, V)}{assoc(A, V)} + \frac{cut(B, V)}{assoc(B, V)}, \tag{1.2}$$

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 [10] that by relaxing (1.2) 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 $d = W\mathbf{1}$ and $D = \text{diag}(d)$, Normalized Cut cost can be written as

$$\inf_z \frac{z^T(D-W)z}{-z^T dd^T z + (1^T d)^2}, \text{ s.t. } z \in \{-1,1\}^n, \; Cz = b. \tag{1.3}$$

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

$$\inf_z \frac{z^T(D-W)z}{-z^T dd^T z + (1^T d)^2}, \text{ s.t. } z^T z = n, \; Cz = b. \tag{1.4}$$

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 (1.4). 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 $L_{\hat{C}} = {N_{\hat{C}}}^T \begin{bmatrix} (D-W) & 0 \\ 0 & 0 \end{bmatrix} N_{\hat{C}}$ and $M_{\hat{C}} = {N_{\hat{C}}}^T \begin{bmatrix} ((1^T d)D - dd^T) & 0 \\ 0 & 0 \end{bmatrix} N_{\hat{C}}$, both positive semidefinite, (see [5]). In the new space we get the following formulation

$$\inf_{\hat{y}} \quad \frac{\hat{y}^T L_{\hat{C}} \hat{y}}{\hat{y}^T M_{\hat{C}} \hat{y}}, \text{ s.t.} \quad \hat{y}_{k+1} = 1, \; ||\hat{y}||^2_{N_{\hat{C}}} = n+1, \tag{1.5}$$

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 [11]. 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 [4] 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 2.1. *If a minima of* (1.5) *exists its dual problem*

$$\sup_t \inf_{||\hat{y}||^2_{N_{\hat{C}}}=n+1} \frac{\hat{y}^T(L_{\hat{C}}+tE_{\hat{C}})\hat{y}}{\hat{y}^T M_{\hat{C}}\hat{y}} \tag{1.6}$$
$$\textit{where } E_{\hat{C}} = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right] - \frac{N_{\hat{C}}^T N_{\hat{C}}}{n+1} = N_{\hat{C}}^T \left[\begin{smallmatrix} -\frac{1}{n+1} I & 0 \\ 0 & 1 \end{smallmatrix}\right] N_{\hat{C}},$$

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 (1.5) through its dual formulation. We will use $F(t, \hat{y})$ to denote the objective function of (1.6), the Lagrangian of problem (1.5). 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 (1.6) is the well known generalized Rayleigh quotient, for which the minima is given by the algebraically smallest generalized eigenvalue [1] of $(L_{\hat{C}} + tE_{\hat{C}})$ and $M_{\hat{C}}$. Letting $\lambda_{\min}(\cdot, \cdot)$ denote the smallest generalized eigenvalue of two entering matrices, we can also write problem (1.6) as

$$\sup_t \ \lambda_{\min}(L_{\hat{C}} + tE_{\hat{C}}, M_{\hat{C}}). \tag{1.7}$$

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 (1.6) 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 $\hat{y}^*$.

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

It is well known that the eigenvalues are analytic (and thereby differentiable) functions as long as they are distinct [6]. Thus, to be able to use a steepest ascent method we need to consider subgradients. Recall the definition of a subgradient [1, 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*

$$g(\sigma) \leq g(\sigma_0) + v^T(\sigma - \sigma_0), \quad \forall \sigma \in \mathbb{R}^{k+1}. \tag{1.8}$$

One can show that if a function is differentiable then the derivative is the only vector satisfying (1.8). 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 3.1. *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*

$$v = \frac{\hat{y}_0^T E_{\hat{C}} \hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0} \tag{1.9}$$

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. From (1.6), we get

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

$$= \frac{\hat{y}_0^T(L_{\hat{C}} + t_0E_{\hat{C}})\hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0} + \frac{\hat{y}_0^T E_{\hat{C}} \hat{y}_0}{\hat{y}_0^T M_{\hat{C}} \hat{y}_0}(t - t_0) = \theta(t_0) + v^T(t - t_0). \tag{1.10}$$

□

3.1.1 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 $\theta_i(t) \geq \theta(t)$. 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

$$\theta_i(t) = F(\hat{y}_i, t_i) + v^T(t - t_i). \tag{1.11}$$

Note that if $\hat{y}_i$ is not unique θ_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

$$\bar{\theta}(t) = \inf_{i \in I} F(\hat{y}_i, t_i) + v^T(t - t_i) \tag{1.12}$$

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 figure 1.1. We then take the solution to the problem $\sup_t \bar{\theta}(t)$, given by

$$\begin{aligned} &\sup_{t,\alpha} \alpha \\ &\alpha \leq F(\hat{y}_i, t_i) + v^T(t - t_i),\ \forall i \in I,\ t_{min} \leq t \leq t_{max}. \end{aligned} \tag{1.13}$$

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 (1.13). 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 (1.13) 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 $\sup_t \theta(t)$.

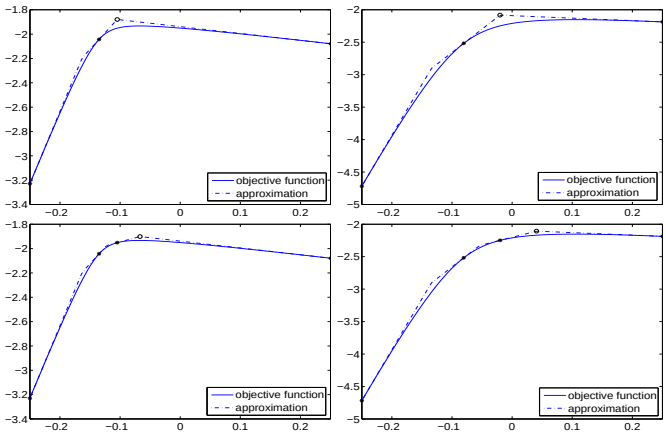


Figure 1.1: Approximations of two randomly generated objective functions. Top: Approximation after 1 step of the algorithm. Bottom: Approximation after 2 steps of the algorithm.

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 (1.6). 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

$$((L_{\hat{C}} + tE_{\hat{C}}) - \lambda(t)M_{\hat{C}})\hat{y}(t) = 0 \tag{1.14}$$

$$||\hat{y}(t)||^2_{N_{\hat{C}}} = n + 1. \tag{1.15}$$

To emphasize the dependence on t we write $\lambda(t)$ for the eigenvalue and $\hat{y}(t)$ for the eigenvector. By differentiating (1.14) we obtain

$$(E_{\hat{C}} - \lambda'(t)M_{\hat{C}})\hat{y}(t) + ((L_{\hat{C}} + tE_{\hat{C}}) - \lambda(t)M)\hat{y}'(t) = 0. \tag{1.16}$$

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 (1.15), obtaining

$$\hat{y}^T(t){N_{\hat{C}}}^T N_{\hat{C}}\hat{y}'(t) = 0. \tag{1.17}$$

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

$$\begin{bmatrix} (L_{\hat{C}}+tE_{\hat{C}})-\lambda(t)M_{\hat{C}} \\ \hat{y}^T(t){N_{\hat{C}}}^T N_{\hat{C}} \end{bmatrix} \hat{y}'(t) = \begin{bmatrix} (-E_{\hat{C}}+\lambda'(t)M_{\hat{C}})\hat{y}(t) \\ 0 \end{bmatrix} \tag{1.18}$$

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 (1.9).

$$\theta''(t) = \frac{d}{dt}\theta'(t) = \frac{d}{dt}\frac{\hat{y}(t)^T E_{\hat{C}}\hat{y}(t)}{\hat{y}(t)^T M_{\hat{C}}\hat{y}(t)} = \frac{2}{\hat{y}(t)^T M_{\hat{C}}\hat{y}(t)}\hat{y}^T(t)\left(E_{\hat{C}} - \theta'(t)M_{\hat{C}}\right)\hat{y}'(t) \tag{1.19}$$

3.2.1 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 overestimator 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 [1] 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 $t = t_i$.

$$t_{\max} = \min_{i;\theta'(t_i)\leq 0} t_i, \tag{1.20}$$

$$t_{\min} = \max_{i;\theta'(t_i)\geq 0} t_i. \tag{1.21}$$

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

$$\theta(t) \approx \theta(t_i) + \theta'(t_i)(t - t_i) + \frac{\theta''(t_i)}{2}(t - t_i)^2. \tag{1.22}$$

and finding its maxima. By differentiating (1.22) it is easily shown that its optima, as well as the next point in the Newton sequence, is given by

$$t_{i+1} = -\frac{\theta'(t_i)}{\theta''(t_i)} + t_i. \tag{1.23}$$

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 [1], 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 1.2 shows a histogram of the number of eigenvalue evaluations for the subgradient-modified Newton method as well as the baseline golden section search.

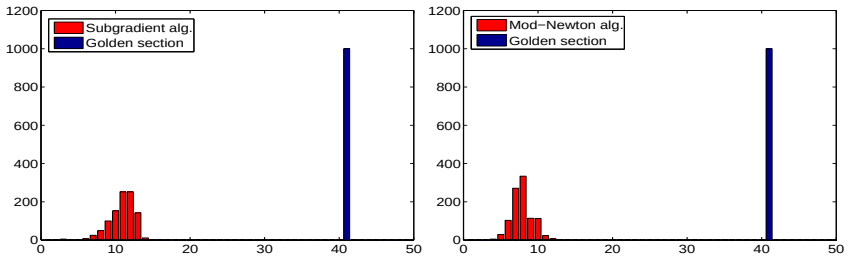


Figure 1.2: Histogram of the number of function evaluations required for 1000-synthetically generated experiments using a golden section method (blue) and the subgradient algorithm (red).

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 figure 1.3, 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 [7]. 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 figure 1.4.

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

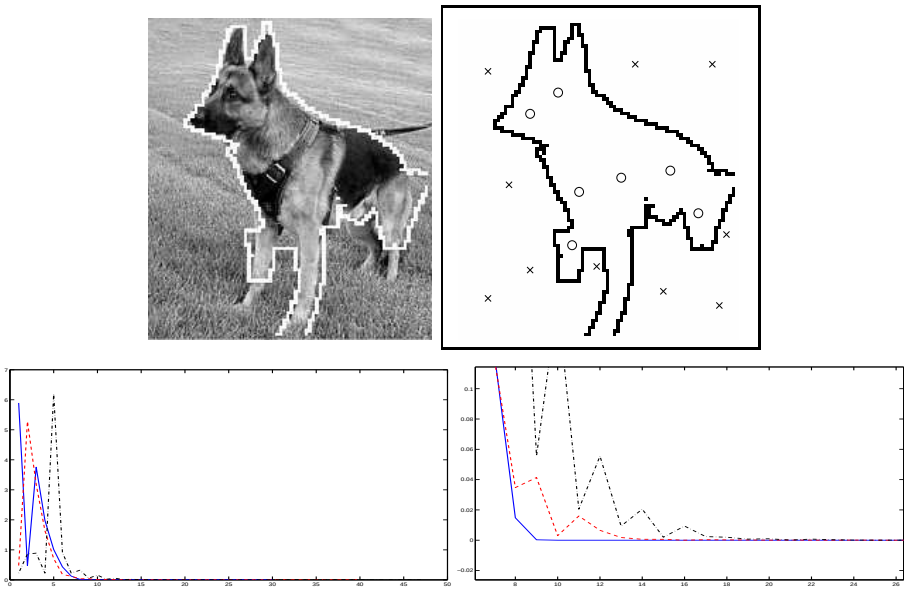


Figure 1.3: Top: Resulting segmentation (left) and constraints applied (right). Here an X means that this pixel belongs to the foreground and an O to the background. Bottom: Convergence of the modified Newton (solid), subgradient (dashed) and the golden section (dash-dotted) algorithms. The algorithms converged after 9, 14 and 23 iteration respectively.

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 figure 1.3 and 1.4 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.

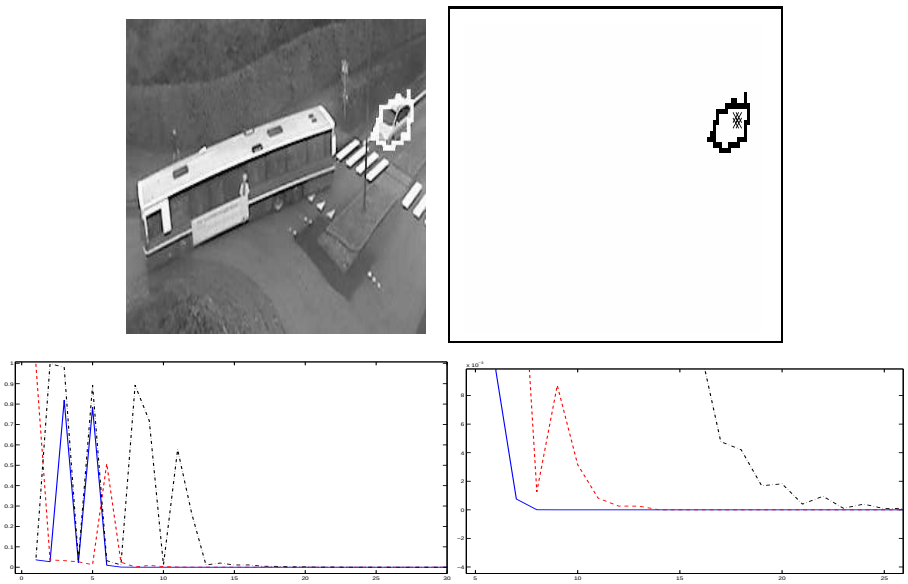


Figure 1.4: Top: Resulting segmentation (left) and constraints applied, in addition to the area requirement used (area = 50 pixels) (right). Here the X in the top right part of the corner means that this pixel belongs to the foreground. Bottom: Convergence of the modified Newton (solid), subgradient (dashed) and the golden section (dash-dotted) algorithms. The algorithms converged after 9, 15 and 23 iteration respectively.

Bibliography

[1] Bazaraa, Sherali, and Shetty. *Nonlinear Programming, Theory and Algorithms.* Wiley, 2006.

[2] 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.

[3] Yuri Boykov and Marie-Pierre 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.

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

[5] A.P. 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.

[6] J. Magnus. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, 1.

[7] Jitendra Malik, Serge Belongie, Thomas K. Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[8] C. Olsson, A.P. 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.

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

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

[11] D.C. Sorensen. Newton's method with a model trust region modification. *SIAM Journal on Nummerical Analysis*, 19(2):409–426, 1982.

PAPER III

Submitted to *Computer Vision and Image Understanding*, 2007.

Main Entry: prob · lem
Pronunciation:\'prä-blәm\
Function: noun
Origin: 1350-1400; from Greek *problema* "a question" literally "thing put forward"; from *proballein* "propose"; from *pro* "forward" + *ballein* "to throw" (see ballistics).

1 a: a question raised for inquiry, consideration, or solution
b: a proposition in mathematics or physics stating something to be done
2 a: an intricate unsettled question
b: a source of perplexity, distress, or vexation
c: difficulty in understanding or accepting <*I have a problem with your saying that*>

Improved Spectral Relaxation Methods for Binary Quadratic Optimization Problems

Carl Olsson, Anders P. Eriksson and Fredrik Kahl

Abstract

In this paper we introduce two new methods for solving binary quadratic problems. While spectral relaxation methods have been the workhorse subroutine for a wide variety of computer vision problems - segmentation, clustering, subgraph matching to name a few - it has recently been challenged by semidefinite programming (SDP) relaxations. In fact, it can be shown that SDP relaxations produce better lower bounds than spectral relaxations on binary problems with a quadratic objective function. On the other hand, the computational complexity for SDP increases rapidly as the number of decision variables grows making them inapplicable to large scale problems.

Our methods combine the merits of both spectral and SDP relaxations - better (lower) bounds than traditional spectral methods and considerably faster execution times than SDP. The first method is based on spectral subgradients and can be applied to large scale SDPs with binary decision variables and the second one is based on the trust region problem. Both algorithms have been applied to several large scale vision problems with good performance.

1 Introduction

Spectral relaxation methods can be applied to a wide variety of problems in computer vision. They have been developed to provide solutions to, e.g., motion segmentation, figure-ground segmentation, clustering, subgraph matching and digital matting [9, 14, 21, 26, 12]. In particular, large scale problems that can be formulated with a binary quadratic objective function are handled efficiently with several thousands of decision variables.

More recently, semidefinite programming (SDP) relaxations have also been applied to the same type of computer vision problems, e.g., [10, 25, 20]. It can be shown that

such relaxations produce better estimates than spectral methods. However, as the number of variables grows, the execution times of the semidefinite programs increase rapidly. In practice, one is limited to a few hundred decision variables.

Spectral and SDP relaxation methods can be regarded as two points on an axis of increasing relaxation performance. We introduce two alternative methods that lie somewhere in between these two relaxations. Unlike standard SDP solvers that suffer from poor time complexity, they can still handle large scale problems. The two methods are based on a subgradient optimization scheme. We show good performance on a number of problems. Experimental results are given on the following problems: segmentation with prior information, binary restoration, partitioning and subgraph matching. Our main contributions are:

- An efficient algorithm for solving binary SDP problems with quadratic objective function based on subgradient optimization is developed. In addition, we show how to incorporate linear constraints in the same program.

- The trust region subproblem is introduced and we modify it to in order to be applicable to binary quadratic problems with a linear term in the objective function.

Many of the application problems mentioned above are known to be NP-hard, so in practice they cannot be solved optimally. Thus one is forced to rely on approximate methods which results in sub-optimal solutions. Certain energy (or objective) functionals may be solved in polynomial time, for example, submodular functionals using graph cuts [11], but this is not the topic of the present paper.

In [8], an alternative (and independent) method is derived which is also based on subgradients, called the *spectral bundle method*. Our subgradient method differs from [8] in that it is simpler (just look for an ascent direction) and we have found empirically on the experimental problems (see Section 5) that our method performs equally well (or better). An in-depth comparison of the two alternatives is, however, beyond the scope of this paper.

1.1 Outline

The outline of this paper is as follows: In the next section we present the problem and some existing approximation techniques for obtaining approximate solutions.

In section 3 we present our algorithm. We develop theory for improving the spectral relaxation, by using the notion of subgradients. Subgradients is a generalization of gradients that is used when a function is not differentiable. We show that for our problem the subgradients can be calculated analytically, to determine ascent directions, to be used in an ascending direction scheme.

In section 4 we study the Trust Region Subproblem, which is an interesting special case in which we only try to enforce the binary constraints on one of the variables. This

has been extensively studied in the optimization literature and we show that it is always possible to solve exactly.

Finally we test our algorithms and compare with existing methods on the following problems: segmentation with prior information, binary restoration, partitioning and subgraph matching. Preliminary results of this work was presented in [15] and [5].

2 Background

In this paper we study different ways to find approximate solutions of the following binary quadratic problem

$$z = \inf y^T A y + b^T y, \quad y \in \{-1, 1\}^n, \tag{1}$$

where A is an $n \times n$ (possibly indefinite) matrix. A common approach for approximating this highly nonconvex problem is to solve the relaxed problem:

$$z_{sp} = \inf_{||x||^2 = n+1} x^T L x, \tag{2}$$

where

$$x = \begin{pmatrix} y \\ y_{n+1} \end{pmatrix}, L = \begin{pmatrix} A & \frac{1}{2}b \\ \frac{1}{2}b^T & 0 \end{pmatrix}.$$

Solving (2) amounts to finding the eigenvector corresponding to the algebraically smallest eigenvalue of L. Therefore we will refer to this problem as the spectral relaxation of (1). The benefits of using this formulation is that eigenvalue problems of this type are well studied and there exist solvers that are able to efficiently exploit sparsity in the matrix L, resulting in fast execution times. A significant weakness of this formulation is that the constraints $y \in \{-1, 1\}^n$ and $y_{n+1} = 1$ are relaxed to $||x||^2 = n + 1$, which often results in poor approximations.

Now let us turn our attention to bounds obtained through semidefinite programming. Using Lagrange multipliers $\sigma = [\,\sigma_1, \ldots, \sigma_{n+1}\,]^T$ for each binary constraint $x_i^2 - 1 = 0$, one obtains

$$\sup_{\sigma} \inf_{x} x^T (L + \operatorname{diag}(\sigma)) x - e^T \sigma, \tag{3}$$

as relaxation of (1). Here e is an $(n+1)$-vector of ones. The inner minimization is finite valued if and only if $(L + \operatorname{diag}(\sigma))$ is positive semidefinite, which we write, $L + \operatorname{diag}(\sigma) \succeq 0$. This gives the equivalent relaxation

$$z_d = \inf_{\sigma} e^T \sigma, \quad L + \operatorname{diag}(\sigma) \succeq 0. \tag{4}$$

We will denote this problem the dual semidefinite problem since it is dual to the problem

$$z_p = \inf_{X \succeq 0} \operatorname{tr}(LX), \quad \operatorname{diag}(X) = I, \tag{5}$$

where X denotes a $(n+1) \times (n+1)$ matrix, as was shown in [3, 10]. Consequently we will call this problem the primal semidefinite program. Since the dual problems (4) and (5) are convex, there is in general no duality gap. In [10], the proposed method is to solve (5) and use randomized hyperplanes (see [7]) to determine an approximate solution to (1). This method has a number of advantages. Most significantly, using a result from [7] one can derive bounds on the expected value of the relaxed solution. It is demonstrated that the approach works well on a number of computer vision problems. On the other hand, solving this relaxation is computationally expensive. Note that the number of variables is $O(n^2)$ for the primal problem (5) while the original problem (1) only has n variables.

3 A Spectral Subgradient Method

In this section we present a new method for solving the binary quadratic problem (1). Instead of using semidefinite programming we propose to solve the (relaxed) problem

$$z_{sg} = \sup_{\sigma} \inf_{||x||^2=n+1} x^T(L + \text{diag}(\sigma))x - e^T\sigma, \tag{6}$$

with steepest ascent. At a first glance it looks as though the optimum value of this problem is greater than that of (3) since we have restricted the set of feasible x. However it is shown in [16] that (3), (5) and (6) are in fact all equivalent. The reason for adding the norm condition to (6) is that for a fixed σ we can solve the inner minimization by finding the smallest eigenvalue.

3.1 Differentiating the objective function

Let

$$\begin{aligned} \mathcal{L}(x,\sigma) &= x^T(L + \text{diag}(\sigma))x - e^T\sigma & (7)\\ f(\sigma) &= \inf_{||x||^2=n+1} \mathcal{L}(x,\sigma). & (8) \end{aligned}$$

Since f is a pointwise infimum of functions linear in σ it is easy to see that f is a concave function. Hence our problem is a concave maximization problem. Equivalently, f can be written as

$$f(\sigma) = (n+1)\lambda_{min}(L + \text{diag}(\sigma)) - e^T\sigma. \tag{9}$$

Here $\lambda_{min}(\cdot)$ denotes the smallest eigenvalue of the entering matrix. It is widely known that the eigenvalues are analytic (and thereby differentiable) functions everywhere as long as they are distinct. To be able to use a steepest ascent method we need to consider subgradients as eigenvalues *will* cross during the optimization. Recall the definition of a subgradient [1].

Definition 3.1. If $f : \mathbb{R}^{n+1} \mapsto \mathbb{R}$ is concave, then $\xi \in \mathbb{R}^{n+1}$ is a subgradient to f at σ_0 if

$$f(\sigma) \leq f(\sigma_0) + \xi^T(\sigma - \sigma_0), \quad \forall \sigma \in \mathbb{R}^{n+1}. \tag{10}$$

Figure 1 shows a geometrical interpretation of (10). Note that if f is differentiable at σ_0, then letting ξ be the gradient of f turns the right hand side of (10) into the tangent plane. One can show that if a function is differentiable then the gradient is the only vector satisfying (10). If f is not differentiable at σ_0 then there are several subgradients satisfying (10).

We will denote the set of all subgradients at a point σ_0 by $\partial f(\sigma_0)$. From (10) it is easy to see that this set is convex and if $0 \in \partial f(\sigma_0)$ then σ_0 is a global maximum.

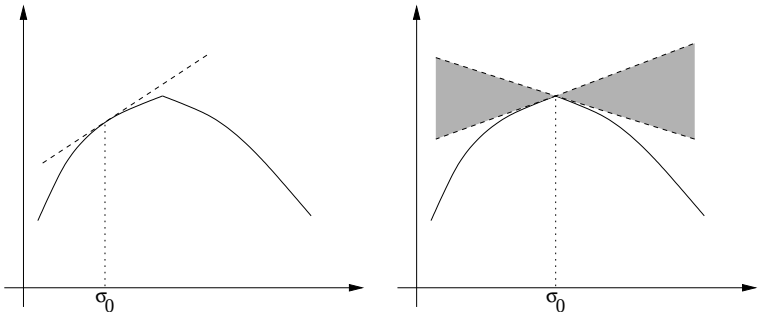


Figure 1: Geometric interpretation of the definition of subgradients. Left: When the function is differentiable in σ_0 the only possible right hand side in (10) is the tangent plane. Right: When the function is not differentiable there are several planes fullfilling (10), each one giving rise to a subgradient.

Next we show how to calculate the subgradients of our problem. Let x^2 be the vector containing the entries of x squared. Then we have:

Lemma 3.1. *If $\bar{x}$ is an eigenvector corresponding to the minimal eigenvalue of $L + diag(\bar{\sigma})$ with norm $||\bar{x}||^2 = n + 1$ then $\xi = \bar{x}^2 - e$ is a subgradient of f at $\bar{\sigma}$.*

Proof. If $\bar{x}$ is an eigenvector corresponding to the minimal eigenvalue of $L + \text{diag}(\bar{\sigma})$ then $\bar{x}$ solves

$$\inf_{||x||^2 = n+1} \mathcal{L}(x, \bar{\sigma}). \tag{11}$$

Assume that $\tilde{x}$ solves

$$\inf_{||x||^2 = n+1} \mathcal{L}(x, \tilde{\sigma}) \tag{12}$$

then

$$\begin{aligned} f(\tilde{\sigma}) &= \tilde{x}^T(L + \text{diag}(\tilde{\sigma}))\tilde{x} - e^T\tilde{\sigma} \\ &\leq \bar{x}^T(L + \text{diag}(\tilde{\sigma}))\bar{x} - e^T\tilde{\sigma} \\ &= f(\bar{\sigma}) + \bar{x}^T\text{diag}(\tilde{\sigma} - \bar{\sigma})\bar{x} - e^T(\tilde{\sigma} - \bar{\sigma}) \\ &= f(\bar{\sigma}) + \sum_i (\tilde{\sigma}_i - \bar{\sigma}_i)(\bar{x}_i^2 - 1) \\ &= f(\bar{\sigma}) + \xi^T(\tilde{\sigma} - \bar{\sigma}). \end{aligned}$$

The inequality comes from the fact that $\tilde{x}$ solves (12). □

The result above is actually a special case of a more general result given in [1] (Theorem 6.3.4). Next we state three corollaries obtained from [1] (Theorems 6.3.7, 6.3.6 and 6.3.11). The first one gives a characterization of all subgradients.

Corollary 3.2. *Let $\mathcal{E}(\bar{\sigma})$ be the set of all eigenvectors with norm $\sqrt{n+1}$ corresponding to the minimal eigenvalue of $L + diag(\bar{\sigma})$. Then the set of all subgradients of f at $\bar{\sigma}$ is given by*

$$\partial f(\bar{\sigma}) = \textit{convhull}(\{x^2 - e;\ x \in \mathcal{E}(\bar{\sigma})\}). \tag{13}$$

We do not give the proof here but note that the inclusion $\partial f(\bar{\sigma}) \supseteq$ convhull$(\{x^2 - e;\ x \in \mathcal{E}(\bar{\sigma})\})$ is obvious by Lemma 3.1 and the fact that $\partial f(\bar{\sigma})$ is a convex set.

Corollary 3.3. *Let $\mathcal{E}(\bar{\sigma})$ be the set of all eigenvectors with norm $\sqrt{n+1}$ corresponding to the minimal eigenvalue of $L + diag(\bar{\sigma})$. Then*

$$f'(\bar{\sigma}, d) = \inf_{\xi \in \partial f(\bar{\sigma})} d^T\xi = \inf_{x \in \mathcal{E}(\bar{\sigma})} d^T(x^2 - e). \tag{14}$$

Here $f'(\bar{\sigma}, d)$ is the directional derivative in the direction d or formally

$$f'(\bar{\sigma}, d) = \lim_{t \to 0^+} \frac{f(\bar{\sigma} + td) - f(\bar{\sigma})}{t}. \tag{15}$$

The first equality is proven in [1]. The second equality follows from Corollary 3.2 and the fact that the objective function $d^T\xi$ is linear in ξ. For a linear (concave) function the optimum is always attained in an extreme point. From [1] we also obtain the corollary

Corollary 3.4. *The direction d of steepest ascent at σ_0 is given by*

$$d = \begin{cases} 0 & if\ \ \xi = 0 \\ \frac{\xi}{||\xi||} & if\ \ \xi \neq 0 \end{cases} \tag{16}$$

where $\xi \in \partial f(\sigma_0)$ is the subgradient with smallest norm.

We will use subgradients in a similar way as gradients is used in a steepest ascent algorithm. Event though there may be many subgradients to choose between, corollary 3.4 finds the locally best one. Figure 2 shows the level sets of a function its subgradients at two points. To the left the function is differentiable at σ_0 and hence the only subgradient is the gradient which points in the direction of steepest ascent. To the right there are several subgradients and the one with the smallest norm points in the direction of steepest ascent.

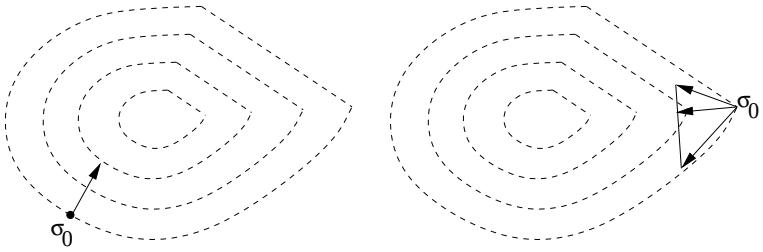


Figure 2: The levelsets of a function and its subgradients at two points. Left: f is differentiable at σ_0 and hence the gradients points in the direction of steepest ascent. Right: f is non differentiable at σ_0 and the direction of steepest ascent is given by the subgradient with the smallest norm.

3.2 Implementation

The basic idea is to find an ascending direction and then to solve an approximation of $f(\sigma)$ along this direction. This process is then repeated until a good solution is found.

3.2.1 Finding ascent directions

The first step is to find an ascending direction. We use Corollary 3.2 to find a good direction. A vector $x \in \mathcal{E}(\bar{\sigma})$ can be written

$$x = \sum_i \lambda_i x_i, \quad \sum_i \lambda_i^2 = 1, \tag{17}$$

where $\{x_i\}$ is an orthogonal base of the eigenspace corresponding to the smallest eigenvalue (with $||x_i||^2 = n+1$). For the full subgradient set we need to calculate $x^2 - e$ for all possible values of λ in (17). In practice, we are led to an approximation and empirically we have found that it is sufficient to pick the vectors $x_i^2 - e$ and use the convex envelope of these vectors as our approximation. Let $\mathcal{S}$ be our approximating set. To determine the best direction, the vector of minimum norm in $\mathcal{S}$ needs to be found. The search can be

written as

$$\inf_{\xi\in\mathcal{S}} ||\xi||^2 = \inf ||\sum_k \mu_k x_k^2 - e||^2, \quad \sum_k \mu_k = 1, \mu_k \geq 0, \tag{18}$$

which is a convex quadratic program in μ_k that can be solved efficiently. To test if an ascending direction d is actually obtained, we use Corollary 3.3 to calculate the directional derivative. In fact we can solve the optimization problem (14) efficiently by using the parameterization (17), which results in

$$\inf d^T \left((\sum_i \lambda_i x_i)^2 - e \right), \quad \sum_i \lambda_i^2 = 1. \tag{19}$$

This is a quadratic function in λ with a norm constraint which can be solved by calculating eigenvalues. If d is not an ascent direction then we add more vectors to the set $\mathcal{S}$ to improve the approximation. In this way we either find an ascending direction or we find that zero is a subgradient, meaning that we have reached the global maximum.

3.2.2 Approximating f along a direction

The next step is to find an approximation $\tilde{f}$ of the objective function along a given direction. We do this by restricting the set of feasible x to a set X consisting of a few of the eigenvectors corresponding to the lowest eigenvalues of $L + \text{diag}(\sigma)$. The intuition behind this choice for X is that if the eigenvalue λ_i is distinct then $x_i^2 - e$ is in fact the gradient of the function

$$(n+1)\lambda_i(L + \text{diag}(\sigma)) - e^T\sigma, \tag{20}$$

where $\lambda_i(\cdot)$ is the ith smallest eigenvalue as a function of a matrix. The expression

$$f_i(t) = x_i^T(L + \text{diag}(\bar{\sigma} + td))x_i - e^T(\bar{\sigma} + td) \tag{21}$$

is then a Taylor expansion around $\bar{\sigma}$ in the direction d. The function f_1 approximates f well in neighborhood around $t = 0$ if the smallest eigenvalue does not cross any other eigenvalue. If it does then one can expect that there is some i such that $\inf(f_1(\sigma), f_i(\sigma))$ is a good approximation.

This gives us a function $\bar{f}$ of the type

$$\bar{f}(\sigma) = \inf_{x_i\in X} x_i^T(L + \text{diag}(\bar{\sigma} + td))x_i - e^T(\bar{\sigma} + td). \tag{22}$$

To optimize this function we can solve the linear program

$$\begin{array}{l} \max_{t,f} f \\ f \leq x_i^T(L + \text{diag}(\bar{\sigma} + td))x_i - e^T(\bar{\sigma} + td) \\ \forall x_i \in X,\ t \leq t_{max}. \end{array} \tag{23}$$

The parameter t_{max} is used to express the interval for which the approximation is valid. The program gives a value for t and thereby a new $\tilde{\sigma} = \bar{\sigma} + td$. In general, $f(\tilde{\sigma})$ is greater than $f(\sigma)$, but if the approximation is not good enough, one needs to improve the approximating function. This can be accomplished by making a new Taylor expansion around the point $\tilde{\sigma}$ and incorporate these terms to our approximation and repeat the process. Figure 3 shows two examples of the objective function f and its approximating function $\tilde{f}$.

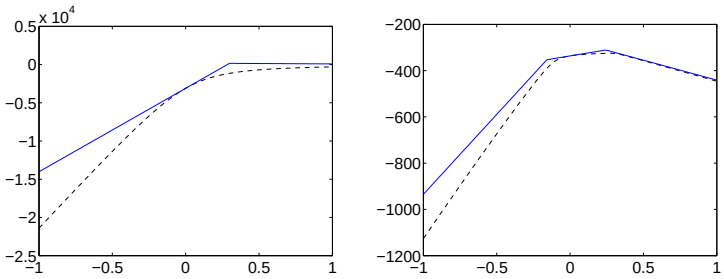


Figure 3: Two approximations of the objective function $f(\sigma + td)$ along an ascent direction d. The dashed line is the true objective f function and the solid line is the approximation $\tilde{f}$.

4 The Trust Region Problem

Another interesting relaxation of our original problem is obtained if we add the additional constraint $y_{n+1} = 1$ to (2). We then obtain a the following relaxation:

$$z_{tr} = \inf_{||y||^2=n} y^T A y + b^T y. \tag{24}$$

We propose to use this relaxation instead of the spectral relaxation (2). Since the objective function is the same as for the spectral relaxation with $y_{n+1} = 1$ it is obvious that

$$z_{sp} \leq z_{tr} \tag{25}$$

holds. Equality will only occur if the solution to z_{sp} happens to have ± 1 as its last component. This is generally not the case. In fact, empirically we have found that the last component is often farther away from ± 1 than the rest of the components. So by enforcing the constraint, that is, solving (24) often yields much better solutions.

Next we will show that it is possible to solve (24) exactly. A problem closely related to (24) is

$$\inf_{||y||^2 \leq n} y^T A y + b^T y. \tag{26}$$

This problem is usually referred to as the trust region subproblem. Solving the problem is one step in a general optimization scheme for descent minimization and it is known as the trust region method [6]. Instead of minimizing a general function, one approximates it with a second order polynomial $y^T Ay + b^T y + c$. A constraint of the type $||y||^2 \leq m$ then specifies the set in which the approximation is believed to be good (the trust region).

The trust region subproblem have been studied extensively in the optimization literature ([18, 19, 23, 22, 17]). A remarkable property of this problem is that, even though it is non convex, there is no duality gap (see [3]). In fact, this is always the case when we have quadratic objective function and only one quadratic constraint. The dual problem of (26) is

$$\sup_{\lambda \leq 0} \inf_{y} y^T Ay + b^T y + \lambda(n - y^T y). \tag{27}$$

In [22] is shown that y^* is the global optimum of (26) if and only if (y^*, λ^*) is feasible in (27) and fulfills the following system of equations:

$$(A - \lambda^* I)y^* = -\frac{1}{2}b, \tag{28}$$

$$\lambda^*(n - y^T y) = 0, \tag{29}$$

$$A - \lambda^* I \succeq 0. \tag{30}$$

The first two equations are the KKT conditions for a local minimum, while the third determines the global minimum. From equation (30) it is easy to see that if A is not positive semidefinite, then λ^* will not be zero. Equation (29) then tells us that $||y||^2 = n$. This shows that for an A that is not positive semidefinite problems (24) and (26) are equivalent. Note that we may always assume that A is not positive semidefinite in (24). This is because we may always subtract mI form A since we have the constant norm condition. Thus replacing A with $A - mI$ for sufficiently large m gives us an equivalent problem with A not positive definite.

A number of methods for solving this problem has been proposed. In [17] semidefinite programming is used to optimize the function $nk(\lambda_{\min}(H(t)) - t)$, where

$$H(t) = \begin{pmatrix} A & \frac{1}{2}b \\ \frac{1}{2}b^T & t \end{pmatrix}, \tag{31}$$

and $\lambda_{\min}$ is the algebraically smallest eigenvalue. In [13] the authors solve $\frac{1}{\psi(\lambda)} - \frac{1}{\sqrt{n}} = 0$ where $\psi(\lambda) = ||(A - \lambda I)^{-1}\frac{1}{2}b||$. This is a rational function with poles at the eigenvalues of A. To ensure that that $A - \lambda I$ is positive semidefinite a Cholesky factorization is computed. If one can afford this, Cholesky factorization is the preferred choice of method. However, the LSTRS-algorithm developed in [18] and [19] is more efficient for large scale problems. LSTRS works by solving a parameterized eigenvalue problem. It searches for a t such that the eigenvalue problem

$$\begin{pmatrix} A & \frac{1}{2}b \\ \frac{1}{2}b^T & t \end{pmatrix} \begin{pmatrix} y \\ 1 \end{pmatrix} = \lambda_{min} \begin{pmatrix} y \\ 1 \end{pmatrix} \tag{32}$$

or equivalently

$$\begin{aligned}(A - \lambda_{min} I)y &= -\frac{1}{2}b, \\ t - \lambda_{min} &= -\frac{1}{2}b^T y,\end{aligned} \tag{33}$$

has a solution. Finding this t is done by determining a λ such that $\phi'(\lambda) = n$, where ϕ is defined by

$$\phi(\lambda) = \frac{1}{4}b^T(A - \lambda I)^\dagger b = -\frac{1}{2}b^T y. \tag{34}$$

It can be shown that λ gives a solution to (33). Since ϕ is a rational function with poles at the eigenvalues of A, it can therefore be expensive to compute. Instead rational interpolation is used to efficiently determine λ. For further details see [18] and [19].

5 Applications

In this section we evaluate the performance of our methods for a few different applications that can be solved as binary quadratic problems. The algorithms are compared with spectral relaxations using Matlab's sparse eigenvalue solver, SDP relaxations using SeDuMi [24] and the spectral bundle algorithm developed by Helmberg [8]. Our spectral subgradient algorithm is implemented in Matlab and the trust region algorithm is based on LSTRS [18] (also Matlab). Note that our implementations consist of simple matlab scripts while the other software has implementations in C (and often highly optimized for speed).

5.1 Segmentation with Prior Information

In our first exampe we will compare the trust region method to the spectral relaxation. We will see that the spectral relaxation can result in poor segmentations when the extra variable is not ± 1. To evaluate the two methods we consider a simple multiclass segmentation problem with prior information.

5.1.1 Graph Representations of Images

The general approach of constructing an undirected graph from an image is shown in 5.1.1. Basically each pixel in the image is viewed as a node in a graph. Edges are formed between nodes with weights corresponding to how alike two pixels are, given some measure of similarity, as well as the distance between them. In an attempt to reduce the number of edges in the graph, only pixels within a small, predetermined neighborhood $\mathcal{N}$ of each other are considered. Cuts made in such a graph will then correspond to a segmentation of the underlying image.

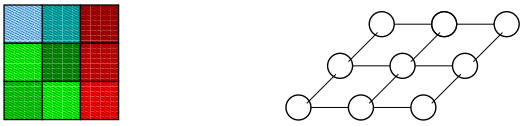
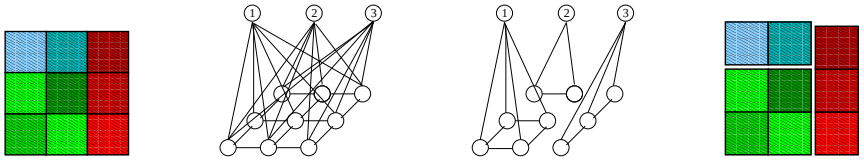


Figure 4: Graph representation of a 3×3 image.

5.1.2 Including Prior Information

To be able to include prior information into the visual grouping process we modify the construction of the graphs in the following way. To the graph G we add k artificial nodes. These nodes do not correspond to any pixels in the image, instead they are meant to represent the k different classes the image is to be partitioned into. The contextual information that we wish to incorporate is modeled by a simple statistical model. Edges between the class nodes and the images nodes are added, with weights proportional to how likely a particular pixel is to a certain class. With the labeling of the k class nodes fixed, a minimal cut on such a graph should group together pixels according to their class likelihood and still preserving the spatial structure, see 5.



(a) Original image (b) Corresponding graph (c) Multiclass min-cut (d) Resulting segmentation

Figure 5: A graph representation of an image and an example three-class segmentation. Unnumbered nodes corresponds to pixels and numbered ones to the artificial class nodes.

5.1.3 Combinatorial Optimization

Next we show how to approximate this problem using the spectral method and the trust region method. Let $Z = [\, z_1, \ldots, z_k \,] \in \{-1, 1\}^{n \times k}$ denote the $n \times k$ assignment matrix for all the n nodes. A 1 in row i of column j signifies that pixel i of the image belongs to class j, and of course -1 in the same position signifies the opposite. If we let W contain the inter-pixel affinities, the min-cut (without pixel class probabilities) can

then be written

$$C_{min} = \inf_{Z} \sum_{i=1}^{k} \sum_{\substack{u \in A_i \\ v \notin A_i}} w_{uv} = \inf_{Z} \sum_{i,j,l} w_{jl}(z_{ij} - z_{il})^2 = \inf_{Z} \sum_{i=1}^{k} z_i^T (D - W) z_i. \quad (35)$$

Here D denotes diag$(W\mathbf{1})$. The assignment matrix Z must satisfy $Z\mathbf{1} = (2-k)\mathbf{1}$. In addition, if the pixel/class-node affinities $P = [\, p_1, \ldots, p_k \,]$ (that is, the probabilities of a single pixel belonging to a certain class) are included and also the labels of the class-nodes are fixated, we get

$$C_{min} = \inf_{\substack{Z \in \{-1,1\}^{n \times k} \\ Z\mathbf{1} = (2-k)\mathbf{1}}} \sum_{i=1}^{k} z_i^T \underbrace{(D - W)}_{L} z_i - 2p_i^T z_i = \inf_{\substack{Z \in \{-1,1\}^{n \times k} \\ Z\mathbf{1} = (2-k)\mathbf{1}}} tr\left(Z^T L Z\right) +$$

$$+2 \underbrace{[\, -p_1^T, \ldots, -p_k^T \,]}_{b^T} \underbrace{\begin{bmatrix} z_1 \\ \vdots \\ z_k \end{bmatrix}}_{z} = \inf_{\substack{z \in \{-1,1\}^{nk} \\ Z\mathbf{1} = (2-k)\mathbf{1}}} z^T \underbrace{\begin{bmatrix} L & 0 & \ldots & 0 \\ 0 & L & \ldots & 0 \\ 0 & \vdots & \ddots & 0 \\ 0 & \vdots & 0 & L \end{bmatrix}}_{A} z + 2b^T z. \quad (36)$$

As $z \in \{-1, 1\}^{nk} \Leftrightarrow z_i^2 = 1$, we can write

$$\begin{aligned} \mu = \quad & \inf_z z^T A z + 2b^T z && (37) \\ \text{s.t.} \quad & z_i^2 = 1 && (38) \\ & Z\mathbf{1} = (2-k)\mathbf{1}. && (39) \end{aligned}$$

The linear subspace of solutions to $Z\mathbf{1} = (2-k)$ can be parametrized as $z = Qy + v$, where Q and v can be chosen so that $Q^T Q = I$ and $Q^T v = 0$ and $y \in \mathbb{R}^{n(k-1)}$. With this change in coordinates and by replacing the discrete constraint $z_i^2 = 1$ with $z^T z = nk$ we arrive at the following relaxed quadratically constrained quadratic program

$$\begin{aligned} \mu = \quad & \inf_y (Qy + v)^T A (Qy + v) + 2b^T (Qy + v), \\ \text{s.t.} \quad & z^T z = (Qy + v)^T (Qy + v) = nk. && (40) \end{aligned}$$

For efficiently solving this problem we here turn our attention to two relaxations that are tractable from a computational perspective. Simplifying (40), we obtain an equivalent trust region problem an the form

$$\mu_{tr} = \inf_{||y||^2 = nk - v^T v} y^T \tilde{A} y + 2\tilde{b}^T y. \quad (41)$$

By adding an extra variable $y_{n(k-1)+1}$ as in (2) we obtain the spectral relaxation.

5.1.4 Experimental Results

As mentioned in the previous section prior knowledge is incorporated into the graph cut framework through the k artificial nodes. For this purpose we need a way to describe each pixel as well as model the probability of that pixel belonging to a certain class.

The image descriptor in the current implementation is based on color alone. Each pixel is simply represented by their three RGB color channels. The probability distribution for these descriptors are modeled using a Gaussian Mixture Model (GMM).

$$p(v|\Sigma,\mu) = \sum_{i=1}^{k} \frac{1}{\sqrt{2\pi|\Sigma_i|}} e^{(-\frac{1}{2}(v-\mu_i)^T\Sigma_i^{-1}(v-\mu_i))}. \tag{42}$$

From a number of manually annotated training images the GMM parameters are then fitted through Expectation Maximization, [4]. This fitting is only carried out once and can be viewed as the learning phase of our proposed method.

The edge weight between pixel i and j and the weights between pixel i and the different class-nodes are given by

$$w_{ij} = e^{(-\frac{r(i,j)}{\sigma_R})} e^{(-\frac{||s(i)-s(j)||^2}{\sigma_W})} \tag{43}$$

$$p_{ki} = \alpha \frac{p(w(i)|i\in k)}{\sum_j p(w(i)|i\in j)}. \tag{44}$$

Here $||\cdot||$ denotes the euclidian norm, $r(i,j)$ the distance between pixel i and j. The tuning parameters λ, σ_R and σ_W weights the importance of the different features. Hence, w_{ij} contains the inter-pixel similarity, that ensures that the segmentation more coherent. p_i describes how likely a pixel is to belong to class k and α is a parameter weighting the importance of spatial structure vs. class probability.

Preliminary tests of the suggested approach were carried out on a limited number of images. We chose to segment the images into four simple classes, sky, grass, brick and background. Gaussian mixture models for each of these classes were firstly acquired from a handful of training images manually chosen as being representative of such image regions, see figure 6.

For an unseen image the pixel affinity matrix W and class probabilities were computed according to (43) and (44). The resulting optimization program was then solved using both the spectral relaxation and the trust region subproblem method. The outcome can be seen in fig. 7. Parameters used in these experiments were $\sigma_R = 1$, $\sigma_W = 1$, $\alpha = 10$ and $\mathcal{N}$ a 9×9 neighborhood structure.

Both relaxations produce visually relevant segmentations, based on very limited training data our proposed approach does appear to use the prior information in a meaningful way. Taking a closer look at the solutions supplied by the trust region method and the spectral relaxation, for these two examples, does however reveal one substantial difference. The spectral relaxation was reached by ignoring the constraint on the homogenized coordinate $y_{n(k-1)+1} = 1$. The solutions to the examples in fig. 7 produces an homogeneous

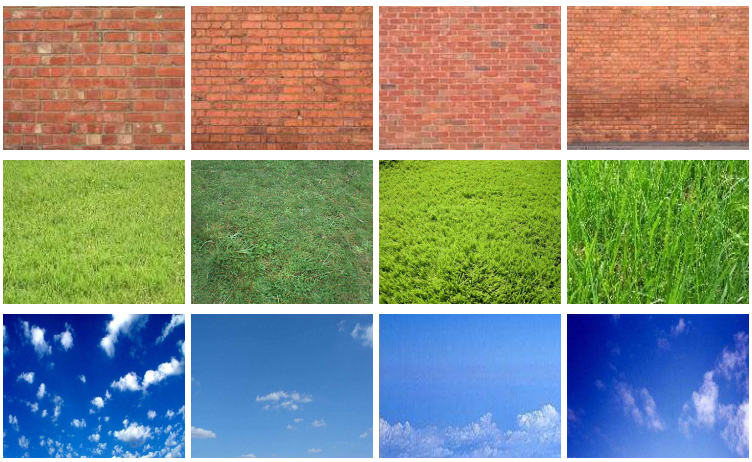


Figure 6: Sample training images.

coordinate value of $y_{n(k-1)+1} \approx 120$, in both cases. As the class probabilities of the pixels are represented by the linear part of eq. 37, the spectral relaxation, in these two cases, thus yields an image partition that that weights prior information much higher than spatial coherence. Any spatial structure of an image will thus not be preserved, the spectral relaxation is basically just a maximum-likelihood classification of each pixel individually.

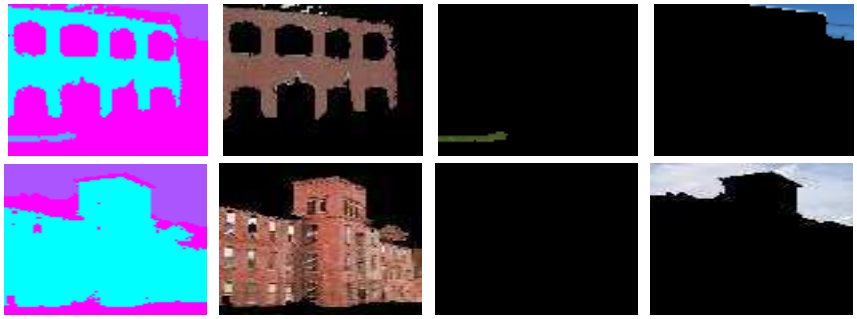
5.2 Binary Restoration

As a test problem (which can be solved exactly by other means), we first consider the problem of separating a signal from noise. The signal $\{x_i\},\ i = 1, ..., n$ is assumed to take the values ± 1. Normally distributed noise with mean 0 and variation 0.6 is then added to obtain a noisy signal $\{s_i\},\ i = 1, ..., n$. Figure 8 (a) and (b) graphs the original signal and the noisy signal respectively for $n = 400$. A strategy to recover the original signal is to minimize the following objective function:

$$\sum_i (x_i - s_i)^2 + \mu \sum_i \sum_{j \in N(i)} (x_i - x_j)^2, \quad x_i \in \{-1, 1\}. \tag{45}$$

Here $N(i)$ means a neighborhood of i, in this case $\{i - 1, i + 1\}$. By adding the (homogenization) variable x_{n+1}, the problem can be transformed to the same form as in (6). Table 1 shows the execution times and Table 2 displays the obtained estimates for different n. For the subgradient method, 10 iterations were run and in each iteration, the 15 smallest eigenvectors were computed for the approximation set $\mathcal{S}$ in (18). Note in

Original images.



(TSP) Resulting class labelling.

(SR) Resulting class labelling.

Figure 7: Example segmentation/classification of an image using both Trust Region Subproblem (TSP) formulation and Spectral Relaxation (SR).

n	*Spectral*	*Trust region*	*Subgradient*	*SDP*
100	0.33	0.60	4.21	3.81
200	0.30	0.62	6.25	13.4
400	0.32	0.68	6.70	180
600	0.33	0.80	10.7	637
800	0.49	1.40	10.1	2365
1000	0.37	1.85	15.2	4830

Table 1: Execution times in seconds for the signal problem.

n	*Spectral*	*Trust region*	*Subgradient*	*SDP*
100	24.3	31.6	40.6	53.1
200	27.4	40.5	53.5	76.1
400	74.9	88.4	139	174
600	134	164	240	309
800	169	207	282	373
1000	178	229	322	439

Table 2: Objective values of the relaxations. A higher value means a better lower bound for the (unknown) optimal value.

particular the growth rate of the execution times for the SDP. Figure 8 (b) - (d) shows the computed signals for the different methods when $n = 400$. The results for other values of n have similar appearance. The spectral relaxations behave (reasonably) well for this problem as the estimated value of x_{n+1} happens to be close to ± 1. Next we consider a similar problem as above, which was also a test problem in [10]. We want to restore the map of Iceland given in Figure 9. The objective function is the same as in (45), except that the neighborhood of a pixel is defined to be all its four neighboring pixels. The size of the image is 78×104, which yields a program with $78 \cdot 104 + 1 = 8113$ variables. Recall that the semidefinite primal program will contain $8113^2 = 65820769$ variables and therefore we have not been able to compute a solution with SeDuMi. In [10], a different SDP solver was used and the execution time was 64885s. Instead we compare with the spectral bundle algorithm [8]. Table 3 gives the execution times and the objective values of the estimations. Figure 10 shows the resulting restorations for the different methods. For the subgradient algorithm, the 4 smallest eigenvalues were used in (18). Even though the spectral relaxation results in a slightly lower objective value than the trust region, the restoration looks just as good. Here the last component of the eigenvector is 0.85 which explains the similarity of these two restorations. The subgradient method yields a solution with values closer to ± 1 as expected. Recall that there is a duality gap which means

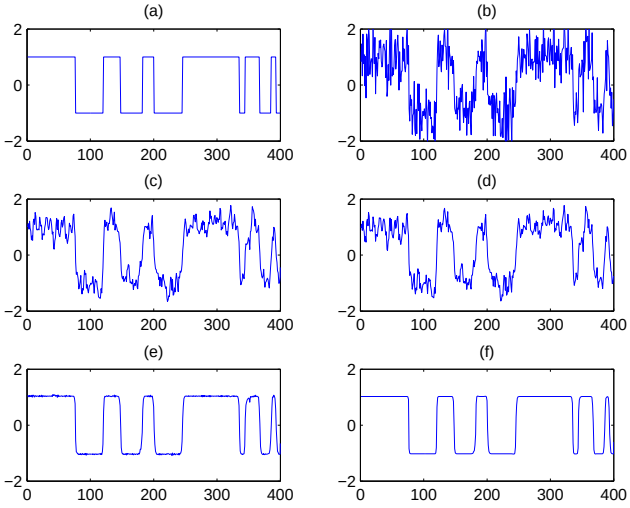


Figure 8: Computed solutions for the signal problem with $n = 400$. (a) Original signal, (b) signal + noise, (c) solution obtained using spectral relaxations, (d) trust region, (e) subgradient algorithm and (f) dual semidefinite program.

that the optimal solution will not attain $x_i = \pm 1$ for all i in general. The spectral bundle method provides a solution where some pixel values are much larger than 1. In order to make the difference between pixels with values -1 and 1 visible in Figure 10(d) we had to replace these pixel values with a smaller value. This results in the white areas in Figure 10(d) and the bar close to the value 2 in Figure 10(d).

5.3 Partitioning

In this section we consider the problem of partitioning an image into perceptionally different parts. Figure 11 (a) shows the image that is to be partitioned. Here we want to separate the buildings from the sky. To do this we use the following regularization term

$$\sum_{ij} w_{ij}(x_i - x_j)^2. \tag{46}$$

The weights w_{ij} are of the type

$$w_{ij} = e^{-\frac{(\mathrm{RGB}(i)-\mathrm{RGB}(j))^2}{\sigma_{\mathrm{RGB}}}} e^{\frac{-d(i,j)^2}{\sigma_d}}, \tag{47}$$

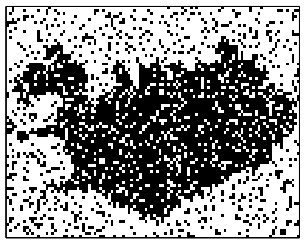


Figure 9: Map of Iceland corrupted by noise.

Method	Time (s)	Lower bound
Spectral	0.48	-1920
Trust region	2.69	-1760
Subgradient, 10 iter.	74.6	-453
Bundle, 5 iter.	150.4	-493

Table 3: Execution times and objective values of the computed lower bounds for the Iceland image.

where RGB(i) denotes the RGB value of pixel i and $d(i,j)$ denotes the distance between pixels i and j. To avoid solutions where all pixels are put in the same partition, and to favour balanced partitions, a term penalizing unbalanced solutions is added. If one adds the constraint $e^T x = 0$ (as in [10]) or equivalently $x^T e e^T x = 0$ we will get partitions of exactly equal size (at least for the subgradient method). Instead we add a penalty term to the objective function yielding a problem of the type

$$\inf x^T (L + \mu e e^T) x, \quad x_i \in \{-1, 1\}. \tag{48}$$

Method	Time (s)
Subgradient, 4 iter.	209
Subgradient, 7 iter.	288
Normalized Cuts	5.5

Table 4: Computing times for the skyline image.

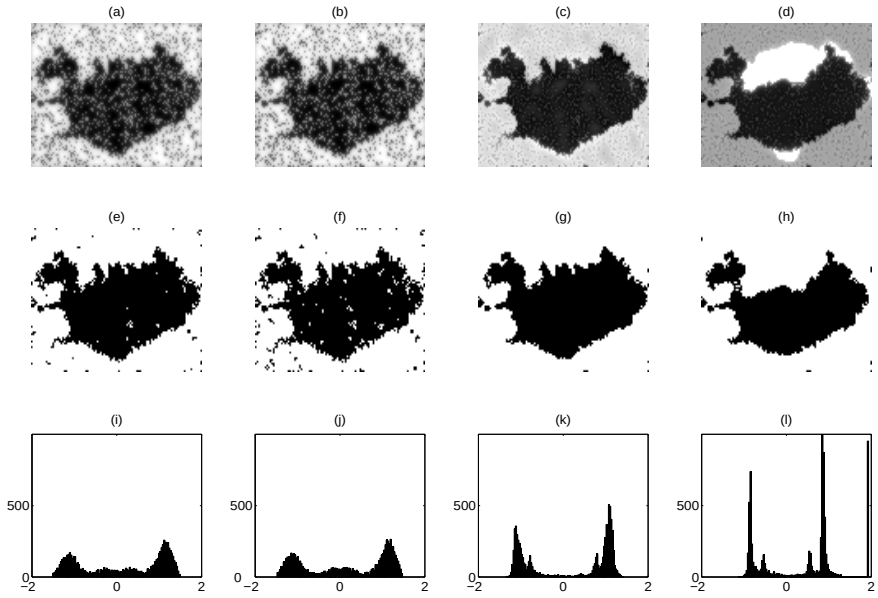


Figure 10: Top row: relaxed soutions. Middle: threshholded solutions. Bottom: histogram of the estimated pixel values. (a),(e),(i): spectral method, (b),(f),(j): trust region, (c),(g),(k): subgradient, 10 iterations, (d),(h),(l): Helmberg's bundle method, 5 iterations.

Observe that this problem is not submodular [11]. Since the size of the skyline image (Figure 11(a)) is 35×55 we obtain a dense matrix of size 1925×1925. However, because of the structure of the matrix it is easy to calculate $(L + \mu ee^T)x$ which is all that is needed to employ power iteration type procedures to calculate eigensystems. This type of matrices are not supported in the spectral bundle software, so we cannot compare with this method. Also, the problem is too large for SeDuMi and there is no point in running the trust region method on this problem since the matrix L has not been homogenized. Figure 11 (b) shows the resulting partition. Figures 11 (e),(f) give the relaxed solutions after 4 and 7 iterations, respectively, of the subgradient algorithm. Both relaxed solutions yield the same result when thresholded at zero. As a comparison, we have included the partitionings obtained from Normalized Cuts [21] which is a frequently applied method for segmentation. The reason for the strange partitioning in Figures 11(c),(d) is that the Fiedler vector in Normalized Cuts essentially contains values close to -0.3 and 3.3 and the median is also close to -0.3. Table 4 shows the computing times of the different methods. Note that the convergence of the subgradient method here is slower than previously, this is because the eigenvalue calculations is more demanding for $(L+\mu ee^T)$.

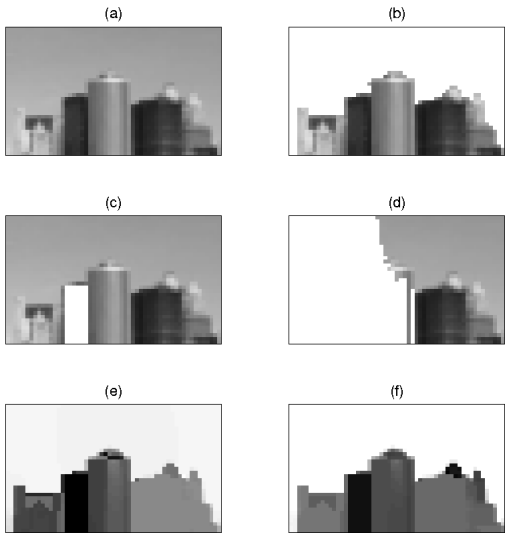


Figure 11: (a) Original image, (b) thresholded segmentation with 7 iterations of the subgradient algorithm (white pixels correspond to one class, remaining pixels are in the other class) (c) Fiedlervector thresholded at the median, (d) Fiedlervector thresholded at the mean, (e),(f) relaxed (untruncated) solutions obtained with 4 and 7 iterations, respectively, of the subgradient algorithm.

5.4 Registration

In our final experiments we consider the registration problem. It appears as a subproblem in many vision applications and similar formulations as the one we propose here have appeared in [2, 20, 25].

Suppose we are given a set of m source points that should be registered to a set of n target points, where $m < n$. Let x_{ij} denote a binary $(0, 1)$-variable which is 1 when source point i is matched to target point j, otherwise 0. As objective function, we choose the quadratic function

$$\sum w_{ijkl} x_{ij} x_{kl}, \tag{49}$$

and set $w_{ijkl} = -1$ if the coordinates of the source points s_i, s_k are *consistent* with the coordinates of the target points t_j, t_l, otherwise $w_{ijkl} = 0$. Two correspondence pairs are considered to be consistent if the distances are approximately the same between source and target pairs, that is,

$$\text{abs}(||s_i - s_k|| - ||t_j - t_l||) < \theta, \tag{50}$$

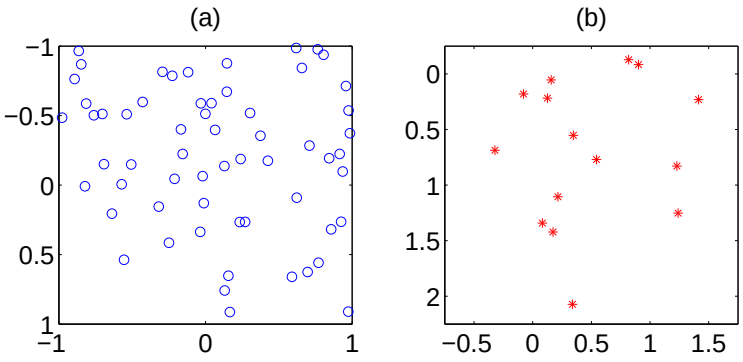


Figure 12: One random example for the registration problem: (a) Target points $n = 60$ and (b) source points $m = 15$.

for some threshold θ. Each source point is a priori equally likely to be matched to any of the target points and hence there is no linear term in the objective function. In addition, each source point should be mapped to one of the target points and hence $\sum_j x_{ij} = 1$ for all i. Also, two source points cannot be mapped to the same target point. This can be specified by introducing $(0, 1)$-slack variables $x_{m+1,j}$ for $j = 1, \ldots, n$ and the constraints $\sum_j x_{m+1,j} = n - m$ as well as $\sum_{i=1}^{m+1} x_{ij} = 1$ for all j.

By substituting $x_{ij} = \frac{z_{ij}+1}{2}$, the problem is turned into a standard $(-1, 1)$-problem, but now with linear equality constraints. In the case of the trust region method we may penalize deviations from the linear constraints by adding penalties of the type $\mu(\sum_j x_{ij} - 1)^2$ to the objective function. One could do the same in the case of the subgradient algorithm, however, in this case the penalties have to be homogenized and may therefore not be as effective as for the trust region method. Instead Lagrange multipliers of the type $\sigma_k(\sum_j x_{ij})^2 - \sigma_k$ are introduced. These multipliers can then be handled in exactly the same way as the constraints $x_{ij}^2 - 1 = 0$. Each constraint gives a new entry in the subgradient vector which is updated in the same way as before.

Method	Time (s)
Trust region	1.9
Subgradient, 7 iter.	43.5
Subgradient, 15 iter.	193
SDP	6867

Table 5: The registration problem with $m = 15, n = 60$.

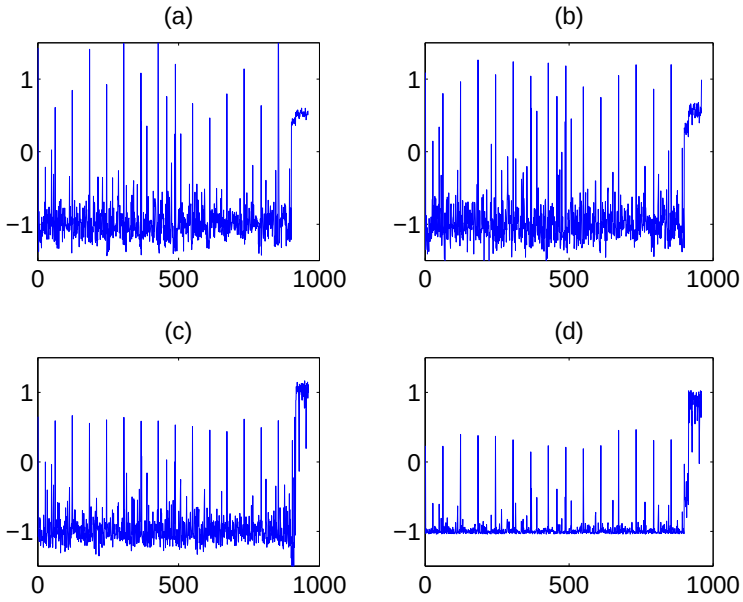


Figure 13: Computed solutions $z = [\, z_{11}, z_{12}, \ldots, z_{m+1,n} \,]$ for the registration problem using (a) the trust region method, (b) the subgradient method, 7 iterations, (c) the subgradient method, 15 iterations, and (d) SDP with SeDuMi, cf. Figure 12.

We have tested the formulation on random data of various sizes. First, coordinates for the n target points are randomly generated with a uniform distribution, then we randomly selected m source points out of the target points, added noise and applied a random Euclidean motion. Figures 12 (a),(b) show the target and source points for one example with $m = 15$ and $n = 60$. The threshold θ is set to 0.1. The untruncated (vectorized) solutions for z_{ij} are plotted in Figure 13 and the resulting registration for the subgradient method is shown in Figure 14. The standard spectral relaxation for this problem works rather poorly as the last entry z_{n+1} is in general far from one. The computing times are given in Table 5. Note that this example has approximately four times as many decision variables as the largest problems dealt with in [20, 25]. For more information on the quality of SDP relaxations for this problem, the reader is also referred to the same papers.

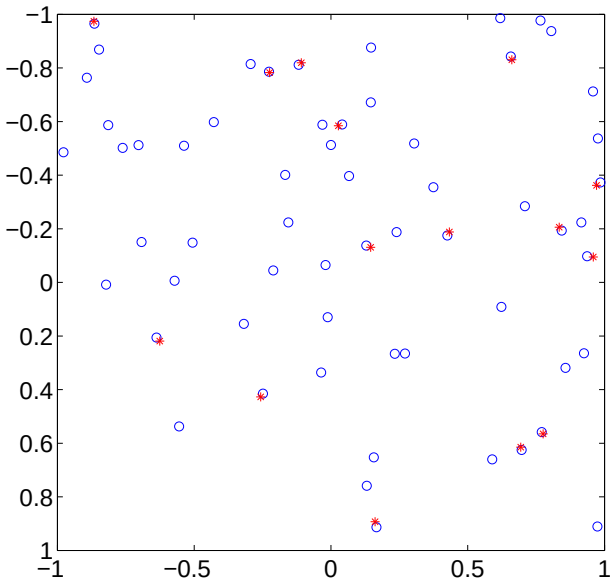


Figure 14: Registration of the source points to their corresponding target points, cf. Figure 12.

6 Conclusions

We have shown how large scale binary problems with quadratic objectives can be solved by taking advantage of the spectral properties of such problems. The approximation gap compared to traditional spectral relaxations is considerably smaller, especially, for the subgradient method. Compared to standard SDP relaxations, the computational effort is less demanding, in particular, for the trust region method. Future work includes to apply the two methods to more problems that can be formulated within the same framework and to make an in-depth experimental comparisons. It would also be interesting to see how the proposed methods behave in a branch-and-bound algorithm for obtaining more accurate estimates.

Bibliography

[1] Bazaraa, Sherali, and Shetty. *Nonlinear Programming, Theory and Algorithms.* Wiley, 1993.

[2] A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Conf. Computer Vision and Pattern Recognition*, pages 26–33, San Diego, USA, 2005.

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

[4] A. Dempster, M. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc.*, 1977.

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

[6] R. Fletcher. *Practical Methods of Optimization*. John WIley & Sons, 1987.

[7] M.X Goemans and D.P Wiliamson. Improved approximation algorithms for maximum cut and satisfiability problem using semidefinite programming. *J.ACM*, 42(6):1115–1145, 1995.

[8] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[9] H. Zha J. Park and R. Kasturi. Spectral clustering for robust motion segmentation. In *European Conf. Computer Vision*, Prague, Czech Republic, 2004.

[10] J. Keuchel, C. Schnörr, C. Schellewald, and D Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(11):1364–1379, 2006.

[11] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[12] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. In *Proc. Conf. Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007.

[13] J.J. Moré and D.C. Sorensen. Computing a trust region step. *SIAM J. Sci. Stat. Comput.*, 4(3):553–572, 1983.

[14] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

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

[16] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 7:51–73, 1995.

[17] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Math. Prog.*, 77(2 Ser.B):273–299, 1997.

[18] M. Rojas, S.A. Santos, and D.C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM Journal on optimization*, 11(3):611–646, 2000.

[19] M. Rojas, S.A. Santos, and D.C. Sorensen. Lstrs: Matlab software for large-scale trust-region subproblems and regularization. Technical Report 2003-4, Department of Mathematics, Wake Forest University, 2003.

[20] C. Schellewald and C. Schnörr. Probabilistic subgraph matching based on convex relaxation. In *Proc. Int. Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 171–186, 2005.

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

[22] D.C. Sorensen. Newton's method with a model trust region modification. *SIAM Journal on Nomerical Analysis*, 19(2):409–426, 1982.

[23] D.C Sorensen. Minimization of a large-scale quadratic fuction subject to a spherical constraint. *SIAM J. Optim.*, 7(1):141–161, 1997.

[24] J.F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[25] P.H.S. Torr. Solving markov random fields using semi definite programming. In *Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[26] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):695–703, 1988.

PAPER IV

Chapter 2 in Licentitate Thesis
Lund University, 2005.

Main Entry: spline
Pronunciation: \'splīne\
Function: noun
Origin: 1750-1760; originally East Anglian dialect; perhaps akin to "splint"; cf. old English *splin* "spindle".

1: a thin wood or metal strip used in building construction
2: a key that is fixed to one of two connected mechanical parts and fits into a keyway in the other; also : a keyway for such a key
3: a function that is defined on an interval, is used to approximate a given function, and is composed of pieces of simple functions defined on subintervals and joined at their endpoints with a suitable degree of smoothness

Bijective Thin-Plate Splines

Anders Eriksson

Centre for Mathematical Sciences
Lund University, Sweden

1.1 Thin-Plate Splines

Thin-plate splines are a class of widely used non-rigid spline mapping functions. It is a natural choice of interpolating function in two dimensions and has been a commonly used tool for over a decade. Introduced and developed by Duchon [2] and Meinguet [6] and popularized by Bookstein [1], its attractions include an elegant mathematical formulation along with a very natural and intuitive physical interpretation.

Consider a thin metal plate extending to infinity in all directions. At a finite number of discrete positions $\mathbf{t}_i \in \mathbb{R}^2$, $i = 1...n$, the plate is at fixed heights z_i, see figure 1.1.

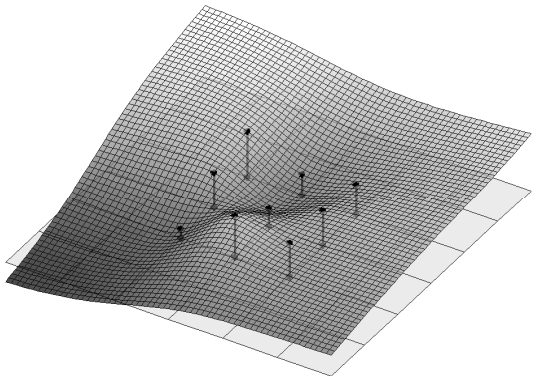


Figure 1.1: The shape of a thin metal plate constrained to lie at some distances above a ground plane at nine different locations.

The metal plate will take the form that minimizes its *bending energy*. In two dimensions the bending energy of a plate described by a function $g(x, y)$ is proportional to

$$J(g) = \int\int_{\mathbb{R}^2} \left(\left(\frac{\partial^2 g}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 g}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 g}{\partial y^2} \right)^2 \right) dxdy. \tag{1.1}$$

Consequently, the metal plate will be described by the function that minimizes (1.1) under the point constraints $g(\mathbf{t}_i) = z_i$. It was proven by Duchon [2] that if such a function exists it is unique.

Given n point constraints $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2 ... \mathbf{t}_n)$, along with the corresponding displacements $\hat{z} = (z_1, z_2, ..., z_n),\ z_i \in \mathbb{R}$. Define

$$\sigma(h) = \begin{cases} ||h||^2 \log(||h||), & ||h|| > 0, \\ 0, & ||h|| > 0, \end{cases} \tag{1.2}$$

where $|| \cdot ||$ is the Euclidian vector norm.

Definition 1.1.1. A thin-plate spline function $g_z : \mathbb{R}^2 \Rightarrow \mathbb{R}$ is a minimizer of (1.1) iff it can be written on the following form

$$\begin{aligned} g_{\mathbf{T},\hat{z}}(\mathbf{x}) &= \sum_{i=1}^{n} \delta_i \sigma(\mathbf{x} - \mathbf{t}_i) + a_1 + a_2 x_1 + a_3 x_2 = \\ &= \underbrace{\begin{bmatrix} \delta_1 & \delta_2 & ... & \delta_n \end{bmatrix}}_{\delta^T} \underbrace{\begin{bmatrix} \sigma(\mathbf{x} - \mathbf{t}_1) \\ \sigma(\mathbf{x} - \mathbf{t}_2) \\ \vdots \\ \sigma(\mathbf{x} - \mathbf{t}_n) \end{bmatrix}}_{\mathbf{s}(\mathbf{x})} + \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = \\ &= \begin{bmatrix} \delta^T & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix}, \end{aligned} \tag{1.3}$$

where $g_{\mathbf{T},\hat{z}}(\mathbf{x})$ δ_i, a_i satisfy

$$g_{\mathbf{T},\hat{z}}(\mathbf{t}_i) = z_i, \tag{1.4}$$

$$\sum_{i=1}^{n} \delta_i = \sum_{i=1}^{n} \delta_i t_{ix} = \sum_{i=1}^{n} \delta_i t_{iy} = 0. \tag{1.5}$$

Combining (1.3), (1.4) and (1.5) the thin-plate spline can be found by solving the equations

$$\begin{bmatrix} \mathbf{s}(\mathbf{t_1}) & 1 & t_{11} & t_{12} \end{bmatrix} \begin{bmatrix} \delta^T \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = z_1,$$

$$\vdots$$

$$\begin{bmatrix} \mathbf{s}(\mathbf{t_n}) & 1 & t_{n1} & t_{n2} \end{bmatrix} \begin{bmatrix} \delta^T \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = z_n, \tag{1.6}$$

$$\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \delta \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = 0, \tag{1.7}$$

$$\begin{bmatrix} \mathbf{T}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = 0. \tag{1.8}$$

With the symmetric $n \times n$ matrix S defined by $S_{ij} = \sigma(\mathbf{t}_i - \mathbf{t}_j)$ we can write (1.6)-(1.8)

$$\underbrace{\begin{bmatrix} S & \mathbf{1}_n & \mathbf{T} \\ \mathbf{1}_n^T & 0 & 0 \\ \mathbf{T}^T & 0 & 0 \end{bmatrix}}_{\Gamma} \begin{bmatrix} \delta^T \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ 0 \\ 0 \end{bmatrix}. \tag{1.9}$$

If $\mathbf{t}_1, ..., \mathbf{t}_n$ are not collinear the symmetric matrix Γ is of full rank (see [3]) and equation (1.9) has the unique solution

$$\begin{bmatrix} \delta^T \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \Gamma^{-1} \begin{bmatrix} \mathbf{z} \\ 0 \\ 0 \end{bmatrix}. \tag{1.10}$$

Consequently, with the following partition of Γ^{-1}

$$\Gamma^{-1} = \begin{bmatrix} \Gamma^{11} & \Gamma^{12} \\ \Gamma^{21} & \Gamma^{22} \end{bmatrix},$$
$$\Gamma^{11},\ n \times n$$
$$\Gamma^{12} = (\Gamma^{21})^T,\ n \times 3$$
$$\Gamma^{22},\ 3 \times 3$$

the thin-plate spline can be defined.

Definition 1.1.2. A thin-plate spline under point constraints $\mathbf{T}$ and $\hat{z}$ can be written

$$\begin{aligned} g_{\mathbf{T},\hat{z}}(\mathbf{x}) &= \begin{bmatrix} \delta^T & a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix} = \\ &= \left(\Gamma^{-1} \begin{bmatrix} \mathbf{z} \\ 0 \\ 0 \end{bmatrix} \right)^T \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{z^T} & 0 & 0 \end{bmatrix} \Gamma^{-1} \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix} = \\ &= \begin{bmatrix} \mathbf{s}(\mathbf{x})^\mathbf{T} & 1 & \mathbf{x} \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \hat{z}. \end{aligned} \tag{1.11}$$

Thin-plate splines of this form has a number of desirable properties. They are both continous and smooth interpolants. Equivariance under similarity transformations also holds.

Lemma 1.1.1. *The thin-plate spline are equivariant under similarity transformations* $\Pi : \mathbb{R}^2 \to \mathbb{R}^2$ *of* $\mathbf{T}$.

$$g_{\Pi(\mathbf{T}),\hat{z}}(\mathbf{x}) = g_{\mathbf{T},\hat{z}}(\Pi^{-1}(\mathbf{x})), \tag{1.12}$$

where

$$\begin{aligned} \Pi(\mathbf{T}) = \alpha(\mathbf{T} + \begin{bmatrix} \mathbf{1}_n\psi_1 & \mathbf{1}_n\psi_2 \end{bmatrix})R, \\ R \in O(2), \\ \alpha \in \mathbb{R}, \\ \psi \in \mathbb{R}^2. \end{aligned}$$

Proof. If the transformed spline and its associated variables is denoted by $\tilde{}$, from 1.1.2 and the preceding discussion we get

$$g_{\tilde{\mathbf{T}},\hat{z}} = \begin{bmatrix} \tilde{\mathbf{s}}(\mathbf{x}) & 1 & \mathbf{x} \end{bmatrix} \begin{bmatrix} \tilde{\Gamma}^{11} \\ \tilde{\Gamma}^{21} \end{bmatrix} \hat{z},$$

and

$$(\tilde{\mathbf{s}}(\mathbf{x}))_i = \sigma(\mathbf{x} - \tilde{\mathbf{t}}_i), \tag{1.13}$$

$$(\tilde{S})_{ij} = \sigma(\tilde{\mathbf{t}}_i - \tilde{\mathbf{t}}_j), \tag{1.14}$$

$$I = \begin{bmatrix} \tilde{\Gamma}^{11} & \tilde{\Gamma}^{12} \\ \tilde{\Gamma}^{21} & \tilde{\Gamma}^{22} \end{bmatrix} \begin{bmatrix} \tilde{S} & \mathbf{1}_n & \tilde{\mathbf{T}} \\ \mathbf{1}_n^T & 0 & 0 \\ \tilde{\mathbf{T}}^T & 0 & 0 \end{bmatrix}. \tag{1.15}$$

First, proving the lemma for rotations only, $\tilde{\mathbf{T}} = \mathbf{T}R$.
It is readily verified that

$$\begin{aligned} (\tilde{\mathbf{s}}(\mathbf{x}))_i &= (\mathbf{s}(\mathbf{x}R^T))_i, \\ (\tilde{S})_{ij} &= (S)_{ij}, \\ \tilde{\Gamma}^{11} &= \Gamma^{11}, \end{aligned}$$

and

$$\tilde{\Gamma}^{21} = \begin{bmatrix} 1 & 0 \\ 0 & R^T \end{bmatrix} \Gamma^{21}.$$

Which gives

$$g_{\mathbf{T}R^T,\hat{z}}(\mathbf{x}) = \begin{bmatrix} \mathbf{s}(\mathbf{x}R^T) & 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \begin{bmatrix} 1 & 0 \\ 0 & R^T \end{bmatrix} \Gamma^{21} \end{bmatrix} \hat{z} =$$

$$\begin{bmatrix} \mathbf{s}(\mathbf{x}R^T) & 1 & \mathbf{x}R^T \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} = g_{\mathbf{T},\hat{z}}(\mathbf{x}R^T).$$

Similarly for translation, $\tilde{\mathbf{T}} = \mathbf{T} + \begin{bmatrix} \mathbf{1}_n\psi_1 & \mathbf{1}_n\psi_2 \end{bmatrix}$.
Here the following holds

$$\begin{aligned} (\tilde{\mathbf{s}}(\mathbf{x}))_i &= (\mathbf{s}(\mathbf{x} - \psi))_i, \\ (\tilde{S})_{ij} &= (S)_{ij}, \\ \tilde{\Gamma}^{11} &= \Gamma^{11}, \\ \tilde{\Gamma}^{21} &= \begin{bmatrix} 1 & -\psi \\ 0 & I \end{bmatrix} \Gamma^{21}. \end{aligned}$$

Which then gives

$$g_{\tilde{\mathbf{T}},\hat{z}}(\mathbf{x}) = \begin{bmatrix} \mathbf{s}(\mathbf{x}-\psi) & 1 & \mathbf{x} \end{bmatrix} \begin{bmatrix} \begin{matrix} \Gamma^{11} \end{matrix} \\ \begin{bmatrix} 1 & -\psi \\ 0 & I \end{bmatrix} \Gamma^{21} \end{bmatrix} \hat{z} = \begin{bmatrix} \mathbf{s}(\mathbf{x}-\psi) & 1 & (\mathbf{x}-\psi) \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} = g_{\mathbf{T},\hat{z}}(\mathbf{x}-\psi).$$

Finally, scaling, $\tilde{\mathbf{T}} = \alpha\mathbf{T}$, gives

$$\begin{aligned}
&(\tilde{\mathbf{s}}(\mathbf{x}))_i = \sigma(||\mathbf{x} - \alpha\mathbf{t}_i||) = \\
&= \frac{1}{2}\Big((x_1 - \alpha t_{i,1})^2 + (x_2 - \alpha t_{i,1})^2\Big) \log\Big((x_1 - \alpha t_{i,1})^2 + (x_2 - \alpha t_{i,1})^2\Big) = \\
&= \frac{\alpha^2}{2}\Big((x_1 - t_{i,1})^2 + (x_2 - t_{i,1})^2\Big) \log\Big(\alpha^2((x_1 - t_{i,1})^2 + (x_2 - t_{i,1})^2)\Big) = \\
&= \alpha^2(s(\frac{\mathbf{x}}{\alpha}))_i + \alpha^2 \log(\alpha)||\frac{\mathbf{x}}{\alpha} - \mathbf{t}_i||^2,
\end{aligned}$$

and similarly

$$(\tilde{S})_{ij} = \alpha^2(S)_{ij} + \alpha^2 \log(\alpha)||\mathbf{t}_i - \mathbf{t}_j||^2,$$

It can be verified that the matrices that satisfy (1.15) are

$$\begin{aligned}
\tilde{\Gamma}^{11} &= \frac{1}{\alpha^2}\Gamma^{11}, \\
\tilde{\Gamma}_0^{21} &= \Gamma_0^{21} - \log(\alpha) \begin{bmatrix} \mathbf{T}_{11}^2 + \mathbf{T}_{12}^2 \\ \vdots \\ \mathbf{T}_{n1}^2 + \mathbf{T}_{n2}^2 \end{bmatrix} \Gamma^{11}, \\
\tilde{\Gamma}_1^{21} &= \frac{1}{\alpha}\Gamma_1^{21}, \\
\tilde{\Gamma}_2^{21} &= \frac{1}{\alpha}\Gamma_2^{21}.
\end{aligned}$$

$$g_{\tilde{\mathbf{T}},\hat{z}}(\mathbf{x}) = g_{\alpha\mathbf{T},\hat{z}}(\mathbf{x}) = \begin{bmatrix} \tilde{\mathbf{s}}(\mathbf{x}) & 1 & \mathbf{x} \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha^2}\Gamma^{11} \\ \tilde{\Gamma}_0^{21} \\ \frac{1}{\alpha}\Gamma_1^{21} \\ \frac{1}{\alpha}\Gamma_2^{21} \end{bmatrix} \hat{z} =$$

$$= \Big(\mathbf{s}(\frac{\mathbf{x}}{\alpha})\Gamma^{11} + \log(\alpha) \begin{bmatrix} ||\frac{\mathbf{x}}{\alpha} - \mathbf{t}_1||^2 \\ \vdots \\ ||\frac{\mathbf{x}}{\alpha} - \mathbf{t}_n||^2 \end{bmatrix}^T \Gamma^{11} + \tilde{\Gamma}_0^{21} + \frac{x_1}{\alpha}\Gamma_1^{21} + \frac{x_2}{\alpha}\Gamma_2^{21}\Big)\hat{z} =$$

$$= \Big(\mathbf{s}(\frac{\mathbf{x}}{\alpha})\Gamma^{11} + \log(\alpha) \begin{bmatrix} \mathbf{T}_{11}^2 + \mathbf{T}_{12}^2 \\ \vdots \\ \mathbf{T}_{n1}^2 + \mathbf{T}_{n2}^2 \end{bmatrix}^T \Gamma^{11} + \Gamma_0^{21} -$$

$$- \log(\alpha) \begin{bmatrix} \mathbf{T}_{11}^2 + \mathbf{T}_{12}^2 \\ \vdots \\ \mathbf{T}_{n1}^2 + \mathbf{T}_{n2}^2 \end{bmatrix} \Gamma^{11} + \frac{x_1}{\alpha}\Gamma_1^{21} + \frac{x_2}{\alpha}\Gamma_2^{21}\Big)\hat{z} =$$

$$\Big(\mathbf{s}(\frac{\mathbf{x}}{\alpha})\Gamma^{11} + \Gamma_0^{21} + \frac{x_1}{\alpha}\Gamma_1^{21} + \frac{x_2}{\alpha}\Gamma_2^{21}\Big)\hat{z} =$$

$$\begin{bmatrix} \mathbf{s}(\frac{\mathbf{x}}{\alpha}) & 1 & \frac{\mathbf{x}}{\alpha} \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha^2}\Gamma^{11} \\ \Gamma_0^{21} \\ \Gamma_1^{21} \\ \Gamma_2^{21} \end{bmatrix} \hat{z} = g_{\mathbf{T},\hat{z}}(\frac{\mathbf{x}}{\alpha}).$$

Combining these three parts completes the proof. □

Lemma 1.1.1 fits nicely in with the metal plate analogy. Rotation, scaling and translation of the location of the point constraints should not affect the bending of the plate but solely result in a corresponding alteration of the plate. From our intuitive understanding of this approach it is expected that the interpolation by such a transformed spline should be equal to a transformation of the original interpolation spline, which is exactly what this lemma confirms.

Finally, for the matrix Γ^{12}, the following also holds

Lemma 1.1.2. *If Γ is the matrix associated with a thin-plate spline mapping with point-constraints $\mathbf{T}$ then with*

$$\Gamma^{12} = \begin{bmatrix} \Gamma_0^{12} & \Gamma_1^{12} & \Gamma_2^{12} \end{bmatrix}.$$

It holds that

$$\begin{aligned} (\Gamma_0^{12})^T\mathbf{1}_n &= 1,\ (\Gamma_1^{12})^T\mathbf{1}_n = 0,\ (\Gamma_2^{12})^T\mathbf{1}_n = 0, \\ (\Gamma_1^{12})^T\mathbf{T}_1 &= 1,\ (\Gamma_0^{12})^T\mathbf{T}_1 = 0,\ (\Gamma_2^{12})^T\mathbf{T}_1 = 0, \\ (\Gamma_2^{12})^T\mathbf{T}_2 &= 1,\ (\Gamma_0^{12})^T\mathbf{T}_1 = 0,\ (\Gamma_1^{12})^T\mathbf{T}_1 = 0. \end{aligned} \tag{1.16}$$

Proof.

$$I = \Gamma^{-1}\Gamma = \begin{bmatrix} \Gamma^{11} & \Gamma^{12} \\ \Gamma^{21} & \Gamma^{22} \end{bmatrix} \begin{bmatrix} S & \mathbf{1}_n & \mathbf{T} \\ \mathbf{1}_n^T & 0 & 0 \\ \mathbf{T}^T & 0 & 0 \end{bmatrix} =$$

$$= \begin{bmatrix} \Gamma^{11}S + \Gamma^{12}\begin{bmatrix} \mathbf{1}_n^T \\ \mathbf{T}^T \end{bmatrix} & \Gamma^{11}\begin{bmatrix} \mathbf{1}_n & \mathbf{T} \end{bmatrix} \\ \\ \Gamma^{21}S + \Gamma^{22}\begin{bmatrix} \mathbf{1}_n^T \\ \mathbf{T}^T \end{bmatrix} & \Gamma^{21}\begin{bmatrix} \mathbf{1}_n & \mathbf{T} \end{bmatrix} \end{bmatrix}$$

$\Rightarrow$

$$\Gamma^{21}\begin{bmatrix} \mathbf{1}_n & \mathbf{T} \end{bmatrix} = \begin{bmatrix} \Gamma_0^{12^T} \\ \Gamma_1^{12^T} \\ \Gamma_2^{12^T} \end{bmatrix} \begin{bmatrix} \mathbf{1}_n & \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix} =$$

$$= \begin{bmatrix} \Gamma_0^{12^T}\mathbf{1}_n & \Gamma_0^{12^T}\mathbf{T}_1 & \Gamma_0^{12^T}\mathbf{T}_2 \\ \Gamma_1^{12^T}\mathbf{1}_n & \Gamma_1^{12^T}\mathbf{T}_1 & \Gamma_1^{12^T}\mathbf{T}_2 \\ \Gamma_2^{12^T}\mathbf{1}_n & \Gamma_2^{12^T}\mathbf{T}_1 & \Gamma_2^{12^T}\mathbf{T}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

□

The thin-plate spline formulation can easily be generalized into higher dimension interpolants. With a different bending energy function, and its associated fundamental solution (eq. (1.2)), the above lemmas can be extended under this generalisation. For more details see [9].

1.1.1 Pair of Thin-Plate Spline Mappings

The thin-plate spline framework can also be employed in a deformation setting, that is mappings from $\mathbb{R}^m$ to $\mathbb{R}^m$. This is accomplished by the combination of several thin-plate spline interpolants. In this section we do however restrict ourselves to $m = 2$.

If instead of understanding the displacement of the thin metal plate as occuring orthogonally to the (x_1, x_2)-plane view them as displacements of the x_1- or x_2- position of the point constraints. With this interpretation, a new function $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ can be constructed from two thin-plate splines, each describing the x_1- and x_2-displacements respectively.

Definition 1.1.3. Given a set of target points $\mathbf{T} = [\mathbf{T}_1\ \mathbf{T}_2] = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_n \end{bmatrix}$, $\mathbf{t}_i \in \mathbb{R}^2$ and a

set of destination points $\mathbf{Y} = [\mathbf{Y}_1\ \mathbf{Y}_2] = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_n \end{bmatrix}$, $\mathbf{y}_i \in \mathbb{R}^2$ A pair of thin-plate splines mapping $\phi_{\mathbf{T},\mathbf{Y}} : \mathbb{R}^2 \to \mathbb{R}^2$ is the bivariate function $\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}))$, where $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are two thin-plate spline interpolants ensuring the point constraints $\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{T}) = \mathbf{Y}$.

The two thin-plate splines satisfying these constraints are

$$g_1(\mathbf{x}) = g_{\mathbf{T},\mathbf{Y}_1}(\mathbf{x}) = \begin{bmatrix} \mathbf{Y}_1^T & 0 & 0 \end{bmatrix} \Gamma^{-1} \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix} \tag{1.17}$$

and

$$g_2(\mathbf{x}) = g_{\mathbf{T},\mathbf{Y}_2}(\mathbf{x}) = \begin{bmatrix} \mathbf{Y}_2^T & 0 & 0 \end{bmatrix} \Gamma^{-1} \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix}. \tag{1.18}$$

Since we know that $g_1(\mathbf{T}) = \mathbf{Y}_1$ and $\phi_2(\mathbf{T}) = \mathbf{Y}_2$ it follows that
$\phi_{\mathbf{T},\mathbf{Y}}(T) = (g_1(\mathbf{T}), g_2(\mathbf{T})) = (\mathbf{Y}_1, \mathbf{Y}_2) = Y$.

Using (1.11), such a pair of thin-plate splines mapping under point constraints $\mathbf{T}$ and $\mathbf{Y}$ is given by

$$\begin{aligned} \phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x})) = [g_1(\mathbf{x})\ \ g_2(\mathbf{x})]) = \\ = \left[\begin{bmatrix} \mathbf{Y}_1^T & 0 & 0 \end{bmatrix} \Gamma^{-1} \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix} \quad \begin{bmatrix} \mathbf{Y}_2^T & 0 & 0 \end{bmatrix} \Gamma^{-1} \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix} \right] = \\ = \left(\begin{bmatrix} \mathbf{Y}_1^T & 0 & 0 \\ \mathbf{Y}_2^T & 0 & 0 \end{bmatrix} \Gamma^{-1} \begin{bmatrix} \mathbf{s}(\mathbf{x}) \\ 1 \\ x_1 \\ x_2 \end{bmatrix} \right)^T = \begin{bmatrix} \mathbf{s}(\mathbf{x})^{\mathbf{T}} & 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \mathbf{Y}. \end{aligned} \tag{1.19}$$

Deformations of this type inherits many of the properties of the underlying thin-plate spline interpolants. Firstly, pair of thin-plate spline mappings are continous, smooth and surjective interpolants. The domain of these mappings is all of $\mathbb{R}^2$ and at infinity $\phi_{\mathbf{T},\mathbf{Y}}$ is purely affine. Equivariance holds, not only on $\mathbf{T}$ of lemma 1.1.1 but also on $\mathbf{Y}$

Lemma 1.1.3. *Thin-plate spline mappings are equivariant under affine transformations* $\Xi : \mathbb{R}^2 \to \mathbb{R}^2$ *of* $\mathbf{Y}$*, i.e.*

$$\phi_{\mathbf{T},\Xi(\mathbf{Y})}(\mathbf{x}) = \Xi(\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x})), \tag{1.20}$$

where

$$\Xi(\mathbf{Y}) = \mathbf{Y}\Psi + \begin{bmatrix} \mathbf{1}_n\psi_1 & \mathbf{1}_n\psi_2 \end{bmatrix}, \quad \Psi \in \mathbb{R}^{2\times 2}, \quad \psi \in \mathbb{R}^2.$$

Proof.

$$\phi_{\mathbf{T},\Xi(\mathbf{Y})}(\mathbf{x}) = \begin{bmatrix} \mathbf{s}(\mathbf{x})^\mathbf{T} & 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \left(\mathbf{Y}\Psi + \begin{bmatrix} \mathbf{1}_n & 0 \\ 0 & \mathbf{1}_n \end{bmatrix} \psi \right) =$$

$$\left(\begin{bmatrix} \mathbf{s}(\mathbf{x})^\mathbf{T} & 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \mathbf{Y} \right) \Psi +$$

$$+ \begin{bmatrix} \mathbf{s}(\mathbf{x})^\mathbf{T} & 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \begin{bmatrix} \mathbf{1}_n & 0 \\ 0 & \mathbf{1}_n \end{bmatrix} \psi = [\text{using lemma 1.1.2}] =$$

$$= \left(\begin{bmatrix} \mathbf{s}(\mathbf{x})^\mathbf{T} & 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \mathbf{Y} \right) \Psi + \psi = \Xi(\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x})).$$

□

1.2 Bijectivity Constraints on Thin-plate Spline Mappings

In spite of its appealing algebraic formulation presented in the previous section, thin-plate spline mappings do have drawbacks and, disregarding computational and numerical issues, one in particular. Namely, bijectivity is never assured. In computer vision, non-linear mappings in $\mathbb{R}^2$ of this sort are frequently used to model deformations in images. The basic assumption is that all the images contain similar structures and therefore there should exist mappings between pairs of images that are both one-to-one and onto. Hence bijective mappings are required.

From section 1.1.1 we have a deformation $\phi_{\mathbf{T},\mathbf{Y}}$ that, for a given set of n control points $\mathbf{T}$, is parameterized (linearly) by the destination points $\mathbf{Y}$. It is of interest knowing which $\mathbf{Y}$ gives a bijective deformation, i.e the set

$$\Omega_\mathbf{T} = \{\mathbf{Y} \in \mathbb{R}^{2n} | \phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x}) \text{ is bijective } \}.$$

Such a mapping $\phi : \mathbb{R}^2 \to \mathbb{R}^2$ is locally bijective at a point $\mathbf{x} \in \mathbb{R}^2$ iff its functional determinant $|J(\phi)|$ is non-zero. Here

$$|J(\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x}))| = \begin{vmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} \end{vmatrix} \tag{1.21}$$

using eq. (1.19)

$$\begin{aligned} \frac{\partial \phi_1}{\partial x_1} = \frac{\partial}{\partial x_1}\left(\left[\begin{array}{cccc} \mathbf{s}(\mathbf{x})^{\mathbf{T}} & 1 & x_1 & x_2 \end{array}\right]\left[\begin{array}{c} \Gamma^{11} \\ \Gamma^{21} \end{array}\right]\mathbf{Y}_1\right) = \\ \left(\mathbf{s}'_{\mathbf{x}_1}(\mathbf{x})^T\Gamma^{11} + 0\cdot\Gamma_0^{12^T} + 1\cdot\Gamma_1^{12^T} + 0\cdot\Gamma_2^{12^T}\right)\mathbf{Y}_1 = \\ \left(\mathbf{s}'_{\mathbf{x}_1}(\mathbf{x})^T\Gamma^{11} + \Gamma_1^{12^T}\right)\mathbf{Y}_1, \end{aligned} \tag{1.22}$$

and similarly

$$\frac{\partial \phi_2}{\partial x_1} = \left(\mathbf{s}'_{\mathbf{x}_1}(\mathbf{x})^T\Gamma^{11} + \Gamma_1^{12^T}\right)\mathbf{Y}_2, \tag{1.23}$$

$$\frac{\partial \phi_1}{\partial x_2} = \left(\mathbf{s}'_{\mathbf{x}_2}(\mathbf{x})^T\Gamma^{11} + \Gamma_2^{12^T}\right)\mathbf{Y}_1, \tag{1.24}$$

$$\frac{\partial \phi_2}{\partial x_2} = \left(\mathbf{s}'_{\mathbf{x}_2}(\mathbf{x})^T\Gamma^{11} + \Gamma_2^{12^T}\right)\mathbf{Y}_2. \tag{1.25}$$

Where

$$\begin{aligned} \mathbf{s}'_{\mathbf{x}_i}(\mathbf{x}) = \frac{\partial}{\partial x_i}\begin{bmatrix} \sigma(\mathbf{x}-\mathbf{t}_1) \\ \sigma(\mathbf{x}-\mathbf{t}_2) \\ \vdots \\ \sigma(\mathbf{x}-\mathbf{t}_n) \end{bmatrix} = \begin{bmatrix} (x_i - t_{1i})\,(1+\log(||\mathbf{x}-\mathbf{t}_1||)) \\ (x_i - t_{2i})\,(1+\log(||\mathbf{x}-\mathbf{t}_2||)) \\ \vdots \\ (x_i - t_{ni})\,(1+\log(||\mathbf{x}-\mathbf{t}_n||)) \end{bmatrix} = \\ = x_i\mathbf{1}_n + \mathbf{T}_i + \begin{bmatrix} (x_i - t_{1i})\,(\log(||\mathbf{x}-\mathbf{t}_1||)) \\ (x_i - t_{2i})\,(\log(||\mathbf{x}-\mathbf{t}_2||)) \\ \vdots \\ (x_i - t_{ni})\,(\log(||\mathbf{x}-\mathbf{t}_n||)) \end{bmatrix}. \end{aligned} \tag{1.26}$$

Inserting into (1.21) yields

$$
\begin{aligned}
&|J(\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x}))| = \\
&\left| \begin{array}{cc} \left(\mathbf{s}'_{\mathbf{x}_1}(\mathbf{x})^T\Gamma^{11} + \Gamma_1^{12^T}\right)\mathbf{Y}_1 & \left(\mathbf{s}'_{\mathbf{x}_2}(\mathbf{x})^T\Gamma^{11} + \Gamma_2^{12^T}\right)\mathbf{Y}_1 \\ \left(\mathbf{s}'_{\mathbf{x}_1}(\mathbf{x})^T\Gamma^{11} + \Gamma_1^{12^T}\right)\mathbf{Y}_2 & \left(\mathbf{s}'_{\mathbf{x}_2}(\mathbf{x})^T\Gamma^{11} + \Gamma_2^{12^T}\right)\mathbf{Y}_2 \end{array} \right| = \\
&= \underbrace{\left(\mathbf{s}'_{\mathbf{x}_1}(\mathbf{x})^T\Gamma^{11} + \Gamma_1^{12^T}\right)}_{=\mathbf{b}_1(\mathbf{x})^T}\mathbf{Y}_1\underbrace{\left(\mathbf{s}'_{\mathbf{x}_2}(\mathbf{x})^T\Gamma^{11} + \Gamma_2^{12^T}\right)}_{=\mathbf{b}_2(\mathbf{x})^T}\mathbf{Y}_2 - \\
&\underbrace{\left(\mathbf{s}'_{\mathbf{x}_2}(\mathbf{x})^T\Gamma^{11} + \Gamma_2^{12^T}\right)}_{=\mathbf{b}_2(\mathbf{x})^T}\mathbf{Y}_1\underbrace{\left(\mathbf{s}'_{\mathbf{x}_1}(\mathbf{x})^T\Gamma^{11} + \Gamma_1^{12^T}\right)}_{=\mathbf{b}_1(\mathbf{x})^T}\mathbf{Y}_2 = \\
&= \mathbf{Y}_1^T\underbrace{\left(\mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T - \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T\right)}_{D_{\mathbf{T}}(\mathbf{x})}\mathbf{Y}_2 = \\
&= \frac{1}{2}\left[\begin{array}{cc} \mathbf{Y}_1^T & \mathbf{Y}_2^T \end{array}\right]\underbrace{\left[\begin{array}{cc} 0 & D_{\mathbf{T}}(\mathbf{x}) \\ D_{\mathbf{T}}(\mathbf{x})^T & 0 \end{array}\right]}_{B(\mathbf{x})}\underbrace{\left[\begin{array}{c} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{array}\right]}_{\hat{\mathbf{Y}}} = \\
&= \frac{1}{2}\hat{\mathbf{Y}}^T B(\mathbf{x})\hat{\mathbf{Y}} = \frac{1}{2}h_{\mathbf{T}}(\mathbf{Y},\mathbf{x}). \qquad (1.27)
\end{aligned}
$$

Using lemma 1.1.2, $\mathbf{b}_i(\mathbf{x})$ can be simplified

$$
\begin{aligned}
&\mathbf{b}_i(\mathbf{x})^T = \left(\Gamma^{11}\mathbf{s}'_{\mathbf{x}_i}(\mathbf{x}) + \Gamma_1^{12}\right) = \\
&= \Gamma^{11}\left(x_i\mathbf{1}_n + \mathbf{T}_i + \underbrace{\left[\begin{array}{c} (x_i - t_{1i})\left(\log(||\mathbf{x} - \mathbf{t}_1||)\right) \\ (x_i - t_{2i})\left(\log(||\mathbf{x} - \mathbf{t}_2||)\right) \\ \vdots \\ (x_i - t_{ni})\left(\log(||\mathbf{x} - \mathbf{t}_n||)\right) \end{array}\right]}_{\gamma_i(\mathbf{x})}\right) + \Gamma_i^{12} = \\
&= \left[\begin{array}{c} \Gamma^{11}\mathbf{1}_n = 0 \\ \Gamma^{11}\mathbf{T}_i = 0 \end{array}\right] = \Gamma^{11}\gamma_i(\mathbf{x}) + \Gamma_i^{12}. \qquad (1.28)
\end{aligned}
$$

Each point $\mathbf{x} \in \mathbb{R}^2$ gives a quadratic constraint on $\mathbf{Y}$, $(\hat{\mathbf{Y}}^T B_{\mathbf{T}}(\mathbf{x})\hat{\mathbf{Y}} \neq 0)$ for local bijectivity. In order to simplify notation, $\mathbf{Y}$ will be used to denote its vectorized version $\hat{\mathbf{Y}}$ as well. The intended form of $\mathbf{Y}$ should be clear from the context. Since $\phi_{\mathbf{T},\mathbf{Y}}$ is a continous mapping, for it to be globally bijective this constraint must either be $> 0,\ \forall \mathbf{x} \in \mathbb{R}^2$ or $< 0,\ \forall \mathbf{x} \in \mathbb{R}^2$.

The set $\Omega_{\mathbf{T}}$ can thus be written

$$\Omega_{\mathbf{T}} = \{\mathbf{Y} \in \mathbb{R}^{2n} | \mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} > 0,\ \forall \mathbf{x} \in \mathbb{R}^2 \text{ or } \mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} < 0,\ \forall \mathbf{x} \in \mathbb{R}^2\}.$$

Defining

$$\Omega^+_{\mathbf{T}} = \{\mathbf{Y} \in \mathbb{R}^{2n} | \mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} > 0,\ \forall \mathbf{x} \in \mathbb{R}^2\}$$

and with $\Omega^-_{\mathbf{T}}$ defined similarly, one can write $\Omega_{\mathbf{T}} = \Omega^+_{\mathbf{T}} \cup \Omega^-_{\mathbf{T}}$. Seeing that if $\mathbf{Y} \in \Omega^+_{\mathbf{T}}$ then $\begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} \mathbf{Y} \in \Omega^-_{\mathbf{T}}$, it does, without loss of generality, suffice to examine $\Omega^+_{\mathbf{T}}$. Hence, references to bijective thin-plate spline mappings will from here on be with respect to the set $\Omega^+_{\mathbf{T}}$.

The sought after set is evidently the intersection of an infinite number of high-dimensional quadratic forms each on the form (1.27). In an attempt at visualisation one can take 2-dimensional intersections of these constraints and plot the resulting quadratic curve for any number of points in $\mathbb{R}^2$, see fig. 1.2.

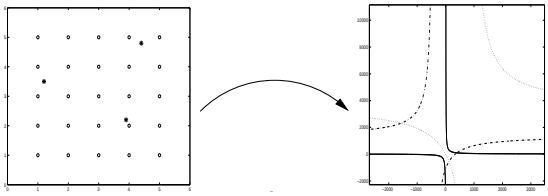


Figure 1.2: The constraints imposed by three points in $\mathbb{R}^2$ on a 2-D affine subset of $\mathbb{R}^{2n}$. Left: The source configuration with three arbitrarily chosen points in $\mathbb{R}^2$ marked. Right: The three resulting quadratic constraints.

This is clearly a somewhat impractical representation of $\Omega^+_{\mathbf{T}}$, an implicit representation would be preferrable. That is a function $e(\mathbf{Y})$ such that

$$\begin{cases} e(\mathbf{Y}) > 0 \Leftrightarrow Y \in \Omega^+_{\mathbf{T}}, \\ e(\mathbf{Y}) \leq 0 \Leftrightarrow Y \notin \Omega^+_{\mathbf{T}}. \end{cases}$$

Such an implicit representation of $\Omega^+_{\mathbf{T}}$ is contained in the affine variety defined by the envelope equations

$$\mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} = 0, \tag{1.29}$$

$$\mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})'_{x_1}\mathbf{Y} = 0, \tag{1.30}$$

$$\mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})'_{x_2}\mathbf{Y} = 0. \tag{1.31}$$

This comes from the fact that equations (1.27) form a family of quadrics in $\mathbb{R}^{2n}$, parametrised by points in $\mathbb{R}^2$. Then the implicit representation must be a subset of the envelope of these functions.

An alternative way of viewing these equations is that points on the boundary of $\Omega_{\mathbf{T}}^+$ must be global minimizers of $h_{\mathbf{T}}(\mathbf{x}, \mathbf{Y}^*)$ over $\mathbb{R}^2$ with global minima 0. With this interpretation (1.29)-(1.31) are the first-order-conditions for such a minima.

However, the task of solving this system of log-linear equations is a formidable one that has yet not been accomplished.

1.3 Properties of $\Omega_{\mathbf{T}}^+$

Despite the high degree of complexity that $\Omega_{\mathbf{T}}^+$ posesses there are still a number of properties that can be identified. Firstly, the set in question actually is of a very familiar shape

Lemma 1.3.1. *The closure of* $\Omega_{\mathbf{T}}^+$, $cl(\Omega_{\mathbf{T}}^+)$ *is*
(i) a generalised double cone
(ii) star-convex around 0
(iii) connected

Proof. It is only necessary to show that $cl(\Omega_{\mathbf{T}}^+)$ is a cone since this implies star-convex around 0 and star-convex implies connected.
The closure of $\Omega_{\mathbf{T}}^+$ can be written
$cl(\Omega_{\mathbf{T}}^+) = \{\mathbf{Y} \in \mathbb{R}^{2n} | \mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} \geq 0,\ \forall \mathbf{x} \in \mathbb{R}^2\}$ then for any $y \in cl(\Omega_{\mathbf{T}}^+)$ obviuosly λy, $\lambda \in \mathbb{R}$, is also in $\in cl(\Omega_{\mathbf{T}}^+)$, hence $cl(\Omega_{\mathbf{T}}^+)$ is a double cone. With a similar reasoning it can easily be shown that $\Omega_{\mathbf{T}}^+$ is a double cone with the origin removed. □

The defining matrix of the quadratic constraints $B_{\mathbf{T}}(\mathbf{x})$ and its subordinate $D_{\mathbf{T}}(\mathbf{x})$ are also suprisingly simple in their form. Some of their characteristics can be summed up in the following two lemmas

Lemma 1.3.2. *The* $n \times n$ *matrix* $D_{\mathbf{T}}(\mathbf{x}) = \mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T - \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T$ *defined in section 1.2*

(i) is non-zero for all $\mathbf{x} \in \mathbb{R}^2 \Rightarrow \mathbf{b}_1(\mathbf{x})$ *and* $\mathbf{b}_2(\mathbf{x})$ *are never parallel,*

(ii) is zero-diagonal,

(iii) is skew-symmetric,

(iv) column rank 2,

(v) has eigenvalues $\lambda = \pm i\lambda_D$, $\lambda_D = \sqrt{\mathbf{b}_1(\mathbf{x})\hat{D}_{\mathbf{T}}(\mathbf{x})\mathbf{b}_2(\mathbf{x})}$.

Proof.

(i) The matrix $\hat{D}_{\mathbf{T}}(\mathbf{x})$ defines the bijectivity constraints on $\mathbf{Y}$ for a given point $\mathbf{x} \in \mathbb{R}^2$. If there exist an $\mathbf{x}$ such that $\hat{D}_{\mathbf{T}}(\mathbf{x}) = 0$ then $\mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} = 0$, for any destination configuration $\mathbf{Y}$. The thin-plate spline mapping is never locally bijective around that point regardless the choice of $\mathbf{Y}$. However, since we know that setting $\mathbf{Y} = \mathbf{T}$ gives the identity mapping, which is bijective. From this contradiction it is concluded that $D_{\mathbf{T}}(\mathbf{x})$ must be non-zero for all $\mathbf{x} \in \mathbb{R}^2$.

If $\mathbf{b}_1(\mathbf{x})$ or $\mathbf{b}_2(\mathbf{x})$ are parallel then $\hat{D}(\mathbf{x}) = 0$ and the implication follows.

(ii) The matrix $\hat{D}_{\mathbf{T}}(\mathbf{x})$ is zero-diagonal since

$$(D_{\mathbf{T}}(\mathbf{x}))_{ii} = (\mathbf{b}_1(\mathbf{x}))_i(\mathbf{b}_2(\mathbf{x}))_i - (\mathbf{b}_2(\mathbf{x}))_i(\mathbf{b}_1(\mathbf{x}))_i = 0.$$

(iii) It is skew-symmetric as

$$\begin{aligned} D_{\mathbf{T}}(\mathbf{x})^T = \big(\mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T - \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T\big)^T = \\ \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T - \mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T = \\ = -\big(\mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T - \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T\big) = -D_{\mathbf{T}}(\mathbf{x}). \end{aligned}$$

(iv) The column rank follows from that each column in $\hat{D}_{\mathbf{T}}(\mathbf{x})$ are the linear combination of non-zero vector non-parallel vectors $\mathbf{b}_1(\mathbf{x})$ and $\mathbf{b}_2(\mathbf{x})$.

(v) Assuming that the eigenvectors of $\hat{D}_{\mathbf{T}}(\mathbf{x})$ can be written $v = \mathbf{b}_1(\mathbf{x}) + \alpha\mathbf{b}_2(\mathbf{x})$. The eigenvalue problem then becomes

$$\begin{aligned} \hat{D}_{\mathbf{T}}(\mathbf{x})v = \lambda v, \\ \Big(\mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T - \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T\Big)\Big(\mathbf{b}_1(\mathbf{x}) + \alpha\mathbf{b}_2(\mathbf{x})\Big) = \\ = \lambda\Big(\mathbf{b}_1(\mathbf{x}) + \alpha\mathbf{b}_2(\mathbf{x})\Big), \\ \Big(\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_2(\mathbf{x}) + \alpha\mathbf{b}_2(\mathbf{x})^T\mathbf{b}_2(\mathbf{x})\Big)\mathbf{b}_1(\mathbf{x}) + \\ \Big(-\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_1(\mathbf{x}) - \alpha\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_2(\mathbf{x})\Big)\mathbf{b}_2(\mathbf{x}) = \lambda\mathbf{b}_1(\mathbf{x}) + \lambda\alpha\mathbf{b}_2(\mathbf{x}). \end{aligned}$$

Then for equality the following must hold

$$\begin{cases} \mathbf{b}_1(\mathbf{x})^T\mathbf{b}_2(\mathbf{x}) + \alpha\mathbf{b}_2(\mathbf{x})^T\mathbf{b}_2(\mathbf{x}) = \lambda, \\ -\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_1(\mathbf{x}) - \alpha\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_2(\mathbf{x}) = \lambda\alpha. \end{cases}$$

Eliminating α gives

$$\lambda^2 + \Big((\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_1(\mathbf{x}))(\mathbf{b}_2(\mathbf{x})^T\mathbf{b}_2(\mathbf{x})) - (\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_2(\mathbf{x}))^2\Big) = 0$$

$$\lambda = \pm\sqrt{\underbrace{-\Big((\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_1(\mathbf{x}))(\mathbf{b}_2(\mathbf{x})^T\mathbf{b}_2(\mathbf{x})) - (\mathbf{b}_1(\mathbf{x})^T\mathbf{b}_2(\mathbf{x}))^2\Big)}_{\geq 0 \text{ (Cauchy-Schwarz)}}} =$$

$$= \pm i\sqrt{\mathbf{b}_1(\mathbf{x})^T\hat{D}_{\mathbf{T}}(\mathbf{x})\mathbf{b}_2(\mathbf{x}))}.$$

Since $\mathbf{b}_1(\mathbf{x})$ and $\mathbf{b}_2(\mathbf{x})$ are never parallell λ_D is always non-zero. Hence the two non-zero eigenvalues of $\hat{D}_{\mathbf{T}}(\mathbf{x})$ must be $\pm i\lambda_D$. It can be noted that as these eigenvalues are both imaginary and sums to zero ($i\lambda_D + (-i\lambda_D) = 0 = Tr(\hat{D}_{\mathbf{T}}(\mathbf{x}))$) in accordance with (ii) and (iii).

□

Lemma 1.3.3. $B_{\mathbf{T}}(\mathbf{x}) = \begin{bmatrix} 0 & D_{\mathbf{T}}(\mathbf{x}) \\ -D_{\mathbf{T}}(\mathbf{x}) & 0 \end{bmatrix}$ *is a zero-diagonal, symmetric* $2n \times 2n$ *matrix with column rank 4 and eigenvalues* $\pm\lambda_D$

Proof. If v and u are the eigenvectors to $\hat{D}_{\mathbf{T}}(\mathbf{x})$ with eigenvalues $i\lambda_D$ and $-i\lambda_D$ it is readily shown that $\begin{bmatrix} v \\ iv \end{bmatrix}$, $\begin{bmatrix} -v \\ iv \end{bmatrix}$, $\begin{bmatrix} u \\ iu \end{bmatrix}$ and $\begin{bmatrix} -u \\ iu \end{bmatrix}$ are eigenvectors to $B_{\mathbf{T}}(\mathbf{x})$ with eigenvalues $-\lambda_D$, λ_D, λ_D and $-\lambda_D$ respectively.

$$\begin{bmatrix} 0 & D_{\mathbf{T}}(\mathbf{x}) \\ -D_{\mathbf{T}}(\mathbf{x}) & 0 \end{bmatrix}\begin{bmatrix} v \\ iv \end{bmatrix} = \begin{bmatrix} ii\lambda_D v \\ -i\lambda_D v \end{bmatrix} = -\lambda_D\begin{bmatrix} v \\ iv \end{bmatrix}$$

Similarly it can be shown for the remaining three. Zero-diagonality, symmetry and column rank follows trivially from the preceding lemma. □

The matrix $\mathbf{B}_{\mathbf{T}}(\mathbf{x})$ is evidently of high dimension and low rank. Its vector- and null-space both vary with $\mathbf{x}$. A linear subspace of $\mathbb{R}^{2n}$ that is a subset of the null space of $\mathbf{B}_{\mathbf{T}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^2$, as well as parts of the affine variety of the quadratic equation $\mathbf{B}_{\mathbf{T}}(\mathbf{x})$ defines, can nevertheless be found.

Lemma 1.3.4. *The function* $h_{\mathbf{T}}(\mathbf{Y}, \mathbf{x})$ *of eq. (1.27) is the zero-function,*

$$h_{\mathbf{T}}(\mathbf{Y}, \mathbf{x}) = \mathbf{Y}^T B(\mathbf{x})\mathbf{Y} = 0,\ \forall \mathbf{x} \in \mathbb{R}^2,$$

if

$$\mathbf{Y} \in \begin{bmatrix} \mathbf{1}_n & 0 \\ 0 & \mathbf{1}_n \end{bmatrix}\bigcup N,$$

where

$$N = \left\{ v = \begin{bmatrix} \mu w \\ \nu w \end{bmatrix} \mid w \in \mathbb{R}^n,\ \mu, \nu \in \mathbb{R}^2 \right\}.$$

Proof. If $\mathbf{Y} \in \begin{bmatrix} \mathbf{1}_n & 0 \\ 0 & \mathbf{1}_n \end{bmatrix}$ then $\mathbf{Y} = \begin{bmatrix} \mathbf{1}_n & 0 \\ 0 & \mathbf{1}_n \end{bmatrix} \tilde{\mathbf{Y}}$.
Using lemma 1.1.2 it follows that

$$h_{\mathbf{T}}(\tilde{\mathbf{Y}}, \mathbf{x}) = \tilde{\mathbf{Y}}^T \begin{bmatrix} \mathbf{1}_n^T & 0 \\ 0 & \mathbf{1}_n^T \end{bmatrix} B_{\mathbf{T}}(\mathbf{x}) \begin{bmatrix} \mathbf{1}_n & 0 \\ 0 & \mathbf{1}_n \end{bmatrix} \tilde{\mathbf{Y}} =$$

$$= \tilde{\mathbf{Y}}^T \begin{bmatrix} -\mathbf{1}_n^T D_{\mathbf{T}}(\mathbf{x}) \mathbf{1}_n & 0 \\ 0 & \mathbf{1}_n^T D_{\mathbf{T}}(\mathbf{x}) \mathbf{1}_n \end{bmatrix} \tilde{\mathbf{Y}}.$$

Expanding gives

$$\mathbf{1}_n^T D_{\mathbf{T}}(\mathbf{x}) \mathbf{1}_n = \mathbf{1}_n^T (b_1(\mathbf{x}) b_2(\mathbf{x})^T - b_2(\mathbf{x}) b_1(\mathbf{x})^T) \mathbf{1}_n =$$

$$= (\mathbf{1}_n^T b_1(\mathbf{x}))(b_2(\mathbf{x})^T \mathbf{1}_n) - (\mathbf{1}_n^T b_2(\mathbf{x}))(b_1(\mathbf{x})^T \mathbf{1}_n)$$

$$= \begin{bmatrix} \mathbf{1}_n^T b_1(\mathbf{x}) = \mathbf{1}_n^T (\Gamma^{11} \gamma_1(\mathbf{x}) + \Gamma_1^{12}) = \mathbf{1}_n^T \Gamma^{11} \gamma_1(\mathbf{x}) + \mathbf{1}_n^T \Gamma_1^{12} = 0 \\ \mathbf{1}_n^T b_2(\mathbf{x}) = \mathbf{1}_n^T (\Gamma^{11} \gamma_2(\mathbf{x}) + \Gamma_2^{12}) = \mathbf{1}_n^T \Gamma^{11} \gamma_2(\mathbf{x}) + \mathbf{1}_n^T \Gamma_2^{12} = 0 \end{bmatrix} = 0$$

$$\Rightarrow h_{\mathbf{T}}(\tilde{\mathbf{Y}}, \mathbf{x}) = \tilde{\mathbf{Y}}^T \begin{bmatrix} -\mathbf{1}_n^T D_{\mathbf{T}}(\mathbf{x}) \mathbf{1}_n & 0 \\ 0 & \mathbf{1}_n^T D_{\mathbf{T}}(\mathbf{x}) \mathbf{1}_n \end{bmatrix} \tilde{\mathbf{Y}} = 0.$$

If $\mathbf{Y} \in N$ then $\mathbf{Y} = \begin{bmatrix} \mu w \\ \nu w \end{bmatrix}$ and

$$h_{\mathbf{T}}\left(\begin{bmatrix} \mu w \\ \nu w \end{bmatrix}, \mathbf{x} \right) = \begin{bmatrix} \mu w^T & \nu w^T \end{bmatrix} B_{\mathbf{T}}(\mathbf{x}) \begin{bmatrix} \mu w \\ \nu w \end{bmatrix} =$$

$$= \begin{bmatrix} \mu w^T & \nu w^T \end{bmatrix} \begin{bmatrix} 0 & D_{\mathbf{T}}(\mathbf{x}) \\ -D_{\mathbf{T}}(\mathbf{x}) & 0 \end{bmatrix} \begin{bmatrix} \mu w \\ \nu w \end{bmatrix} =$$

$$= \mu\nu w^T D_{\mathbf{T}}(\mathbf{x}) w - \nu\mu w^T D_{\mathbf{T}}(\mathbf{x}) w = 0.$$

□

Next we address the issues of boundedness and convexity.

1.3.1 Boundedness

Obviously, from the equivariance property of lemma 1.1.3, the set in question is indeed unbounded. Since the composition of bijective deformations is also bijective, any bijective target configuration can be deformed by any mapping from the unbounded set of bijective affine transformations $\mathbb{R}^2 \to \mathbb{R}^2$ and still belong to $\Omega^+_{\mathbf{T}}$.

However, if the affine transformations are disregarded is $\Omega^+_{\mathbf{T}}$ still unbounded? By studying one-dimensional intersections of the set, it can be shown (for specific configurations $\mathbf{T}$) that the set is indeed bounded if this restriction is introduced. Consider the subset E of configurations in which the first three points have the same positions as corresponding points in $\mathbf{T}$, i.e. E is formed by perturbing all but the first three points of $\mathbf{T}$. Define the subset $\Omega^+_{\mathbf{T},E}$ as

$$\Omega^+_{\mathbf{T},E} = \{\mathbf{Y} \in E | h_{\mathbf{T}}(\mathbf{Y}, \mathbf{x}) > 0,\ \forall \mathbf{x} \in \mathbb{R}^2\}.$$

These are the configurations in E which gives bijective thin-plate-spline transformations. Now study one-dimensional affine subspaces of E containing $\mathbf{T}$, i.e.

$$E_{\mathbf{d}} = \{\mathbf{T} + s\mathbf{d} | s \in \mathbb{R}\},$$

where $\mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}$ represents changes in configurations. Here $\mathbf{d}$ must be zero at the elements corresponding to the fixed points so that $E_{\mathbf{d}} \subset E$. This intersection of $E_{\mathbf{d}}$ with $\Omega^+_{\mathbf{T},E}$ is

$$\Omega^+_{\mathbf{T},E_d} = \{\mathbf{Y} \in E_{\mathbf{d}} | h_{\mathbf{T}}(\mathbf{Y}, \mathbf{x}) > 0,\ \forall \mathbf{x} \in \mathbb{R}^2\}.$$

Here

$$h_{\mathbf{T}}(\mathbf{Y}, \mathbf{x}) = h_{\mathbf{T}}(\mathbf{T} + s\mathbf{d}, \mathbf{x}) = a_{\mathbf{d}}(\mathbf{x})s^2 + b_{\mathbf{d}}(\mathbf{x})s + c_{\mathbf{d}}(\mathbf{x}).$$

Since $h_{\mathbf{T}}(\mathbf{Y}, \mathbf{x})$ is quadratic in its first argument, for each point $\mathbf{x} \in \mathbb{R}^2$, we thus get a quadratic constraint on s. Here the coefficients of the second order constraints are given by

$$\begin{aligned} a_{\mathbf{d}}(x) &= \mathbf{d}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{d}, \\ b_{\mathbf{d}}(x) &= 2\mathbf{T}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{d}, \\ c_{\mathbf{d}}(x) &= \mathbf{T}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{T}. \end{aligned} \tag{1.32}$$

Lemma 1.3.5. *The function $b_{\mathbf{d}}(\mathbf{x})$ can be simplified as*

$$b_{\mathbf{d}}(\mathbf{x}) = [\mathbf{b}_1(\mathbf{x})^T \ \ \mathbf{b}_2(\mathbf{x})^T]\mathbf{d} = \mathbf{b}(\mathbf{x})^T\mathbf{d}.$$

The function $c_{\mathbf{d}}(\mathbf{x})$ is independent of both $\mathbf{d}$ and $\mathbf{x}$. In fact

$$c_{\mathbf{d}}(\mathbf{x}) = 2.$$

Proof. Using lemma 1.1.2 gives

$$\mathbf{T}_1^T\mathbf{b}_1(\mathbf{x}) = \mathbf{T}_1^T(\Gamma^{11}\gamma_1(\mathbf{x}) + \Gamma_1^{12^T}) = \mathbf{T}_1^T\Gamma_1^{12^T} = 1,$$

$$\mathbf{T}_1^T\mathbf{b}_2(\mathbf{x}) = \mathbf{T}_1^T(\Gamma^{11}\gamma_2(\mathbf{x}) + \Gamma_2^{12^T}) = \mathbf{T}_1^T\Gamma_2^{12^T} = 0,$$

$$\mathbf{T}_2^T\mathbf{b}_1(\mathbf{x}) = \mathbf{T}_2^T(\Gamma^{11}\gamma_1(\mathbf{x}) + \Gamma_1^{12^T}) = \mathbf{T}_2^T\Gamma_1^{12^T} = 0,$$

$$\mathbf{T}_y^T\mathbf{b}_2(\mathbf{x}) = \mathbf{T}_2^T(\Gamma^{11}\gamma_2(\mathbf{x}) + \Gamma_2^{12^T}) = \mathbf{T}_3^T\Gamma_2^{12^T} = 1.$$

So

$$\mathbf{T}_1^T D_{\mathbf{T}}(\mathbf{x}) = \mathbf{T}_1^T(\mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T - \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T) = \mathbf{b}_2(\mathbf{x})^T,$$
$$\mathbf{T}_2^T D_{\mathbf{T}}(\mathbf{x}) = \mathbf{T}_2^T(\mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T - \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T) = -\mathbf{b}_1(\mathbf{x})^T.$$

This implies that

$$b_{\mathbf{d}}(x) = 2\mathbf{T}^T B_{\mathbf{T}}(x)\mathbf{d} = [\mathbf{T}_1^T \;\; \mathbf{T}_2^T]B_{\mathbf{T}}(\mathbf{x})\mathbf{d} = [-\mathbf{T}_2^T D_{\mathbf{T}}(\mathbf{x}) \;\; \mathbf{T}_1^T D_{\mathbf{T}}(\mathbf{x})]\mathbf{d} =$$
$$= [\mathbf{b}_1(\mathbf{x})^T \;\; \mathbf{b}_2(\mathbf{x})^T]\mathbf{d} = \mathbf{b}(\mathbf{x})^T\mathbf{d}$$

and

$$c_{\mathbf{d}}(\mathbf{x}) = 2\mathbf{T}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{T} = [\mathbf{b}_1(\mathbf{x})^T \;\; \mathbf{b}_2(\mathbf{x})^T]\mathbf{T} = \mathbf{b}_1(\mathbf{x})^T\mathbf{T}_1 + \mathbf{b}_2(\mathbf{x})^T\mathbf{T}_2 = 2.$$

□

A sufficent condition on the boundedness of $\Omega^+_{\mathbf{T},E_d}$ is that there exists a point $\mathbf{x} \in \mathbb{R}^2$ such that $a_{\mathbf{d}}(\mathbf{x}) < 0$ since this will limit the distance s for which the spline mapping is bijective. To prove that $\Omega^+_{\mathbf{T},E}$ is unbounded it is sufficient to show that $\Omega^+_{\mathbf{T},E_d}$ is bounded for every direction $\mathbf{d}$, i.e. that $a_{\mathbf{d}}(\mathbf{x})$ never can be non-negative.

Here we need to study the space of all functions $a_{\mathbf{d}}(\mathbf{x})$ as the direction d is varied.

Lemma 1.3.6. *Given a thin-plate-spline defined by n separate control points, assume that the first three points define an affine basis. All possible functions $a_{\mathbf{d}}(\mathbf{x})$ given by (1.32) lie in the $D = (n+1)^2 - n$ dimensional space A of functions spanned by functions $a_{ij}(\mathbf{x})$,*

$$a_{ij}(\mathbf{x}) = f_{1,i}(\mathbf{x})f_{2,j}(\mathbf{x}), \tag{1.33}$$

where

$$f_1(\mathbf{x}) = \begin{bmatrix} \gamma_1(\mathbf{x}) \\ 1 \end{bmatrix},$$
$$f_2(\mathbf{x}) = \begin{bmatrix} \gamma_2(\mathbf{x}) \\ 1 \end{bmatrix}.$$

Proof. The function $a_{\mathbf{d}}(\mathbf{x})$ can be written

$$a_{\mathbf{d}}(\mathbf{x}) = \mathbf{d}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{d} = 2\mathbf{b}_1(\mathbf{x})^T(\mathbf{d}_1\mathbf{d}_2^T - \mathbf{d}_2\mathbf{d}_1^T)\mathbf{b}_1(\mathbf{x})^T =$$
$$= 2(\gamma_1(\mathbf{x})^T\Gamma^{11} + (\Gamma_1^{12})^T)\Big(\mathbf{d}_1\mathbf{d}_2^T - \mathbf{d}_2\mathbf{d}_1^T\Big)(\Gamma^{11}\gamma_2(\mathbf{x}) + \Gamma_2^{12}) =$$
$$= \gamma_1(\mathbf{x})^T \underbrace{\Big(\Gamma^{11}\Big(\mathbf{d}_1\mathbf{d}_2^T - \mathbf{d}_2\mathbf{d}_1^T\Big)\Gamma^{11}\Big)}_{\text{zero-diagonal}} \gamma_2(\mathbf{x}) +$$
$$+2(\Gamma_1^{12})^T\Big(\mathbf{d}_1\mathbf{d}_2^T - \mathbf{d}_2\mathbf{d}_1^T\Big)\gamma_2(\mathbf{x}) - 2(\Gamma_2^{12})^T\Big(\mathbf{d}_1\mathbf{d}_2^T - \mathbf{d}_2\mathbf{d}_1^T\Big)\gamma_1(\mathbf{x}) +$$
$$+2(\Gamma_1^{12})^T\Big(\mathbf{d}_1\mathbf{d}_2^T - \mathbf{d}_2\mathbf{d}_1^T\Big)\Gamma_2^{12} = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\alpha_{ij}f_{1,i}(\mathbf{x})f_{2,j}(\mathbf{x}) =$$
$$= \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\alpha_{ij}a_{ij}(\mathbf{x})$$

With $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ defined as above. As the matrix $\Gamma^{11}\Big(\mathbf{d}_1\mathbf{d}_2^T - \mathbf{d}_2\mathbf{d}_1^T\Big)\Gamma^{11}$ is zero-diagonal, $\alpha_{ij} = 0$ for all $i = j$, except for $i = j = 1$ giving the dimension of A. □

Theorem 1.3.1. *For a number of grids T, including rectangular regular grids of $l \times m$ with $l < 10$ and $m < 10$, the set of perturbations that leave three of the corner points fixed and gives bijective thin-plate splines is bounded in all directions for which $\mathbf{d}_1$ is not parallell to $\mathbf{d}_2$.*

Proof. The proof follows from explicit study of the basis functions $a_{ij}(\mathbf{x})$ for these grids. Thus for a given point configuration $\mathbf{T}$ and assuming that three of the points in $\mathbf{T}$ constitute an affine basis it is possible to calculate a basis A of functions which contain all possible functions $a_{\mathbf{d}}(\mathbf{x})$ with $\mathbf{d}$ leaving the affine basis fixed. By studying the feasibility of the convex set

$$\{z \in \mathbb{R}^D,\ z \neq 0|, Az \geq 0, (\mathbf{1}^T A)^T z = 1\},$$

with A containing as rows the D basis functions sampled at a discrete number of points. It can be shown that there exists no non-negative functions in A, except the zero function. The only directions $\mathbf{d}$ for which $a_{\mathbf{d}}(\mathbf{x})$ is constantly equal to zero are those with which $\mathbf{d}_1$ and $\mathbf{d}_2$ are parallel. □

1.3.2 Convexity

In certain computer vision applications it is desirable to find deformations that map two or more images onto each other optimally. In optimization theory the main issue is not that of linearity and nonlinearity, but convexity and nonconvexity, any convex properties

of $\Omega_{\mathbf{T}}^{+}$ is therefore of great interest. Since the set in question is the intersection of an infinite number of non-convex sets, it would be expected that $\Omega_{\mathbf{T}}^{+}$ is non-convex. This is also the case.

Lemma 1.3.7. *In general* $\Omega_{\mathbf{T}}^{+}$ *is not a convex set.*

Proof. Proof by counter-example. For $\Omega_{\mathbf{T}}^{+}$ to be convex then for any $\mathbf{y}_1, \mathbf{y}_2 \in \Omega_{\mathbf{T}}^{+}$ the line $\lambda\mathbf{y}_1 + (1-\lambda)\mathbf{y}_2$ must also be in $\Omega_{\mathbf{T}}^{+}$ for $0 \leq \lambda \leq 1$. A simple counter-example where this convexity requirement is not met can be found by choosing $\mathbf{T}$ to be a regular 3×3 rectangular grid, and $\mathbf{y}_1$, $\mathbf{y}_2$ slightly altered versions of $\mathbf{T}$, see fig 1.3. □

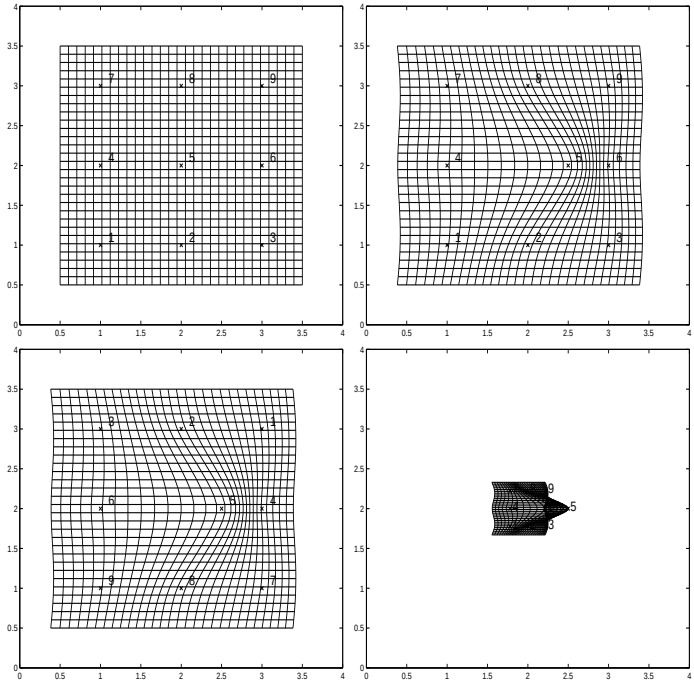


Figure 1.3: A simple example illustrating the non-convexity of $\Omega_{\mathbf{T}}^{+}$ Top left: Source configuration $\mathbf{T}$, Top right: Target configuration $\mathbf{y}_1$, Bottom left: Target configuration $\mathbf{y}_2$, Bottom right: Target configuration $\mathbf{y}_{1+2} = \lambda\mathbf{y}_1 + (1-\lambda)\mathbf{y}_2$, $\lambda = 0.4$. Clearly $\mathbf{y}_1$ and $\mathbf{y}_2$ are bijective but $\mathbf{y}_{1+2}$ is not.

Adopting the approach of disregarding affine transformations from section 1.3.1 does not make the set display any convex characteristics. As in the proof of the preceeding lemma a counterexample can easily be constructed, see fig 1.4.

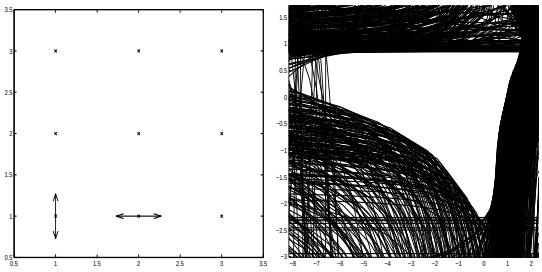


Figure 1.4: Example of non-convex intersection with $\Omega_{\mathbf{T}}^{+}$ under affine restriction. Here only the two left-most points in the bottom row are permitted to move in one dimension, as indicated by the arrows. The resulting set is clearly non-convex.

As it was observed that these non-convex intersections often involved points on the boundary of the convex hull of $\{\mathbf{t}_i\},\ i = 1, ..., n$. The idea was to examine if not permitting points on this boundary to move would ensure convexity. This proved not to be the case, a example of this can be seen in fig. 1.5.

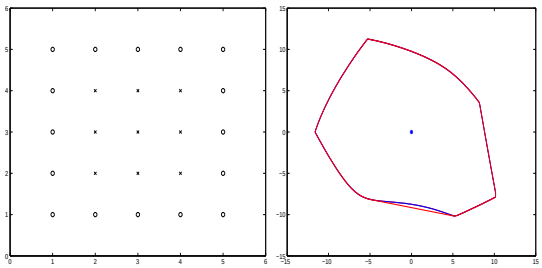


Figure 1.5: An example of a non-convex intersection with $\Omega_{\mathbf{T}}^{+}$ with the restriction that only points in the interior of the convex hull of $\{\mathbf{t}_i\},\ i = 1, ..., n$ were permitted to move.

Apparently $\Omega_{\mathbf{T}}^{+}$ is a highly non-convex set. However, there are restrictions for which convexity can be achieved.

Lemma 1.3.8. *The set $\Omega_{\mathbf{T},E}^{+}$ is convex if the affine subspace $E \subseteq N$, with N defined as in lemma 1.3.4.*

Proof. With

$$\Omega^+_{T,E} = \{\mathbf{Y} \in E | \mathbf{Y}^T \mathbf{B}(\mathbf{x}) \mathbf{Y} > 0 \, , \forall \mathbf{x} \in \mathbb{R}^2\},$$
$$E = \{U + Ey | E \in \mathbb{R}^{2n \times l}, \, U \in \mathbb{R}^{2n}, \, y \in \mathbb{R}^l \}$$

we get $Y = U + Ey$. Consequently, using lemma 1.3.4

$$\mathbf{Y}^T \mathbf{B}(\mathbf{x}) \mathbf{Y} = (U + Ey)^T \mathbf{B}(\mathbf{x})(U + Ey) =$$
$$= y^T (\underbrace{E^T \mathbf{B}(\mathbf{x}) E}_{=0}) y + 2U^T B(\mathbf{x}) y + U^T B(\mathbf{x}) U = 2U^T B(\mathbf{x}) y + U^T B(\mathbf{x}) U.$$

The set $\Omega^+_{T,E}$ is now defined by linear constraints, a polytope and is therefore convex. □

Corollary 1.3.1. *The feasible bijective thin-plate spline mappings when only displacing one target point location make up a convex set.*

Proof. This follows trivially from lemma 1.3.8 as the corresponding affine subset is contained in N □

Finally, there are strong indications that $\Omega^+_{\mathbf{T}}$ is star-convex around $\mathbf{T}$. That is, that the intersection of $\Omega^+_{\mathbf{T}}$ and any affine one-dimensional subspace of $\mathbb{R}^{2n}$ containing $\mathbf{T}$ is convex. However, proving this statement still remains open.

1.4 Sufficient Conditions for Bijectivity

Given the complexity of the set of bijective thin-plate spline deformations, the enterprise of finding the defining expressions analytically is a formidable one. Instead one can use numerical methods to derive conditions on $\Omega^+_{\mathbf{T}}$. By finding subsets of $\Omega^+_{\mathbf{T}}$, through different relaxation methods, sufficient conditions for bijectivity can be obtained. In this section we discuss some of these conditions.

1.4.1 Maximum-Volume Inscribed Sphere

A sufficient condition for bijectivity could be a sphere S contained in $\Omega^+_{\mathbf{T}}$, so that if $\mathbf{Y} \in S \Rightarrow \mathbf{Y} \in \Omega^+_{\mathbf{T}}$. Obviously the larger the volume of the sphere contained in $\Omega^+_{\mathbf{T}}$ is the better the sufficient condition would be.

Let S_R be a sphere with radius R defined by an quadratic inequality

$$S_R = \left\{ d \in \mathbb{R}^{2n} \| \, -\frac{1}{R^2} d^T d + 1 > 0 \right\}. \tag{1.34}$$

Using the notation from section 1.2, $\Omega^+_{\mathbf{T}}$ is the intersection of quadrics on the form

$$C(\mathbf{x}) = \left\{ d \in \mathbb{R}^{2n} \mid d^T B_{\mathbf{T}}(\mathbf{x}) d + 2\mathbf{b}(\mathbf{x})^T d + 2 > 0, \, \mathbf{x} \in \mathbb{R}^2 \right\}$$

it is clear that $S_R \subset \Omega_{\mathbf{T}}^{+}$ if $S \subset C(\mathbf{x}),\ \forall x \in \mathbb{R}^2$

Theorem 1.4.1. *A thin-plate spline mapping $\phi_{\mathbf{T},\mathbf{Y}} : \mathbb{R}^2 \to \mathbb{R}^2$ with n point constraints $\mathbf{T}$ and $\mathbf{Y}$ is bijective if*

$$||\mathbf{Y} - \mathbf{T}|| < R = \frac{1}{\sqrt{\max_{\mathbf{x}\in\mathbb{R}^2}(\lambda_M(\mathbf{x}))}}, \tag{1.35}$$

that is if $\mathbf{Y}$ is inside a sphere centered at $\mathbf{T}$ with radius R. Here $\lambda_M(\mathbf{x})$ are the eigenvalues of the matrix

$$M(\mathbf{x}) = \begin{bmatrix} \mathbf{b}_1(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T & \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T \\ \mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T & \mathbf{b}_2(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T \end{bmatrix} \tag{1.36}$$

Proof. The S-procedure, a commonly used method for dealing with quadratic constraints [7] , gives that S_R is in $C(\mathbf{x})$ if there exist a $\tau \geq 0$ such that

$$\begin{bmatrix} B_{\mathbf{T}}(\mathbf{x}) & \mathbf{b}(\mathbf{x}) \\ \mathbf{b}(\mathbf{x})^T & 2 \end{bmatrix} - \tau \begin{bmatrix} -\frac{1}{R^2}I & 0 \\ 0 & 1 \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} B_{\mathbf{T}}(\mathbf{x}) + \tau\frac{1}{R^2}I & \mathbf{b}(\mathbf{x}) \\ \mathbf{b}(\mathbf{x})^T & 2-\tau \end{bmatrix} \succeq 0,$$

By the Schur complement, this is equivalent to

$$\left(B_{\mathbf{T}}(\mathbf{x}) + \tau\frac{1}{R^2}I\right) - \frac{1}{2-\tau}\mathbf{b}(\mathbf{x})\mathbf{b}(\mathbf{x})^T \succeq 0,$$

$$0 \leq \tau \leq 2.$$

Setting $\tau = 1$ gives

$$\begin{aligned} &B_{\mathbf{T}}(\mathbf{x}) + \frac{1}{R^2}I - \mathbf{b}(\mathbf{x})\mathbf{b}(\mathbf{x})^T = \\ &= \begin{bmatrix} 0 & \mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T - \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T \\ \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T - \mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T & 0 \end{bmatrix} - \\ &- \begin{bmatrix} \mathbf{b}_1(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T & \mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T \\ \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T & \mathbf{b}_2(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T \end{bmatrix} + \frac{1}{R^2}I = \\ &= \frac{1}{R^2}I - \underbrace{\begin{bmatrix} \mathbf{b}_1(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T & \mathbf{b}_1(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T \\ \mathbf{b}_2(\mathbf{x})\mathbf{b}_1(\mathbf{x})^T & \mathbf{b}_2(\mathbf{x})\mathbf{b}_2(\mathbf{x})^T \end{bmatrix}}_{M}(\mathbf{x}) \succeq 0. \end{aligned}$$

This holds if $\frac{1}{R^2}$ is greater than the largest eigenvalue of $M(\mathbf{x})$, or

$$R \leq \frac{1}{\sqrt{\max_{\mathbf{x}\in\mathbb{R}^2}(\lambda_M(\mathbf{x}))}}.$$

□

Even though this theorem provides a simple sufficient condition for bijectivity it does require the computation of a large number eigenvalues. As eigenvalue computation involving large matrices is a notoriously arduous task, it should be avoided as much as possible. Fortunately, a closer look at the matrix $M(\mathbf{x})$ from the preceding theorem reveals a relatively simple expression of the largest eigenvalue of such matrices.

Theorem 1.4.2. *The largest eigenvalue of a matrix on the form*

$$M = \begin{bmatrix} uu^T & vu^T \\ uv^T & vv^T \end{bmatrix},$$

where $v,\ u \in \mathbb{R}^n$ *and not parallel, is equal to*

$$\lambda_{max} = \frac{1}{2}\left(u^T u + v^T v + \sqrt{(u^T u - v^T v)^2 + 4(u^T v)^2}\right). \tag{1.37}$$

Proof. Assuming that the symmetric rank 4 matrix M has eigenvectors on the form $\begin{bmatrix} u + av \\ cu + dv \end{bmatrix}$. Then finding the eigenvalues of M means solving

$$\begin{bmatrix} vv^T & uv^T \\ vu^T & uu^T \end{bmatrix} \begin{bmatrix} u + av \\ cu + dv \end{bmatrix} = \lambda \begin{bmatrix} u + av \\ cu + dv \end{bmatrix}.$$

Multiplying gives

$$\begin{bmatrix} uu^T u + auu^T v + cvu^T u + dvu^T v \\ uv^t u + auv^T v + cvv^T u + dvv^T v \end{bmatrix} = \lambda \begin{bmatrix} u + av \\ cu + dv \end{bmatrix},$$
$$\begin{bmatrix} (u^T u + au^T v)u + (cu^T u + du^T v)v \\ (v^T u + av^T v)u + (cv^T u + dv^T v)v \end{bmatrix} = \begin{bmatrix} \lambda u + \lambda av \\ \lambda cu + \lambda dv \end{bmatrix}.$$

For equality

$$\begin{cases} (u^T u + au^T v) = \lambda, \\ (cu^T u + du^T v) = \lambda a, \\ (v^T u + av^T v) = \lambda c, \\ (cv^T u + dv^T v) = \lambda d \end{cases}$$

must hold.
Continuing solving the equation system yields

$$\begin{cases} (u^T u + au^T v) = \lambda, \\ (cu^T u + du^T v) = (au^T u + a^2 u^T v), \\ (v^T u + av^T v) = cu^T u + acu^T v, \\ (cv^T u + dv^T v) = du^T u + adu^T v. \end{cases}$$

Eliminating λ from the second equation gives

$$c = \frac{v^T u + a v^T v}{u^T u + a u^T v}.$$

Inserted into the third and fourth equations

$$\begin{cases} ((\frac{v^T u + a v^T v}{u^T u + a u^T v})u^T u + d u^T v) = a u^T u + a^2 u^T v, \\ ((\frac{v^T u + a v^T v}{u^T u + a u^T v})v^T u + d v^T v) = d u^T u + a d u^T v. \end{cases}$$

Now solving for d

$$d = v^T u \frac{v^T u + a v^T v}{(u^T u + a u^T v)(u^T u + a u^T v - v^T v)},$$

and back-substitition gives a single polynomial equation in a

$$\begin{aligned}(u^T v)^3 a^4 + (u^T v)^2 (3(u^T u) - v^T v) a^3 + 3 u^T v((u^T u)^2 - (u^T u)(v^T v)) a^2 + \\ + ((u^T u)^3 - (u^T u)^2 v^T v - u^T u (u^T v)^2 - (u^T u)^2 v^T v + u^T u (v^T v)^2 - \\ - v^T v (u^T v)^2) a - (u^T u)^2 u^T v + (u^T u)(v^T v)(u^T v) - (u^T v)^3 = 0,\end{aligned}$$

which can be factorized into

$$\begin{aligned}&\Big((u^T v) a^2 + (u^T u - v^T v) a - u^T v\Big) \\ &\quad \Big((u^T v)^2 a^2 + 2(u^T v)(u^T u) a + ((u^T u)^2 + (u^T v)^2 - (u^T u)(v^T v))\Big) = 0.\end{aligned}$$

The first parenthesis gives

$$\begin{aligned}a_{1,2} &= -\frac{u^T u - v^T v}{2 u^T v} \pm \sqrt{\left(\frac{u^T u - v^T v}{2 u^T v}\right)^2 + 1} = \\ &= \frac{-(u^T u - v^T v) \pm \sqrt{(u^T u - v^T v)^2 + 4 (u^T v)^2}}{2 u^T v}\end{aligned}$$

and the second

$$\begin{aligned}a_{3,4} = -\frac{u^T u}{u^T v} \pm \sqrt{\left(\frac{u^T u}{u^T v}\right)^2 - \frac{(u^T u)^2 + (u^T v)^2 - (u^T u)(v^T v)}{(u^T v)^2}} = \\ = \frac{-u^T u \pm \sqrt{(u^T v)^2 - (u^T u)(v^T v)}}{u^T v}.\end{aligned}$$

Finally with $\lambda = u^T u + a u^T v$ the eigenvalues of M can written

$$\lambda_{1,2} = u^T u + a_{1,2} u^T v = \frac{1}{2}\left(u^T u + v^T v \pm \sqrt{(u^T u - v^T v)^2 + 4(u^T v)^2} \right)$$
$$\lambda_{3,4} = \pm u^T u + a_{3,4} u^T v = \sqrt{(u^T u)(v^T v) - (u^T v)^2}$$

Since M is a rank 4 matrix and $\lambda_1, ..., \lambda_4 \neq 0$ the initial assumption on the form of the eigenvectors is correct and the non-zero eigenvalues of M are the ones given above. It only remains to determine which of these eigenvalues is the largest. Obviously $\lambda_1 \geq \lambda_2$ and $\lambda_3 \geq \lambda_4$. Comparing λ_1 and λ_3

$$\begin{aligned}
\lambda_1^2 - \lambda_3^2 = &\frac{1}{4}\left((u^T u + v^T v) + \sqrt{(u^T u - v^T v)^2 + 4(u^T v)^2} \right)^2 - \\
&- \Big((u^T u)(v^T v) - (u^T v)^2 \Big) = \frac{1}{2}(u^T u)^2 - (u^T u)(v^T v) + \frac{1}{2}(v^T v)^2 + \\
&+ (u^T v)^2 + \frac{1}{2}(u^T u + v^T v)\sqrt{(u^T u - v^T v)^2 + 4(u^T v)^2} = \\
&= \frac{1}{2}\underbrace{\Big((u^T u) - (v^T v) \Big)^2}_{\geq 0} + \underbrace{(u^T v)^2}_{\geq 0} + \\
&+ \underbrace{\frac{1}{2}(u^T u + v^T v)}_{\geq 0}\underbrace{\sqrt{(u^T u - v^T v)^2 + 4(u^T v)^2}}_{\geq 0} \geq 0 \\
\Rightarrow &\lambda_1^2 \geq \lambda_3^2.
\end{aligned}$$

Since $\lambda_1, \lambda_3 > 0$ this implies that $\lambda_1 > \lambda_3$. □

Applying this result to theorem 1.4.1 results in the following corollary.

Corollary 1.4.1. *The largest eigenvalue of the matrix $M(\mathbf{x})$ from theorem 1.4.1 is given by*

$$\lambda_M(\mathbf{x}) = \frac{1}{2}\Big(b_1(\mathbf{x})^T b_1(\mathbf{x}) + b_2(\mathbf{x})^T b_2(\mathbf{x}) + \sqrt{(b_1(\mathbf{x})^T b_1(\mathbf{x}) - b_2(\mathbf{x})^T b_2(\mathbf{x}))^2 + 4(b_1(\mathbf{x})^T b_2(\mathbf{x}))^2} \Big). \tag{1.38}$$

Remark 1.4.1. It can be noted that two of the smaller eigenvalues of $M(\mathbf{x})$ are identical to the eigenvalues of the matrix $B_{\mathbf{T}}(\mathbf{x})$.

$$\lambda_{3,4} = \pm\sqrt{(b_1(\mathbf{x})^T b_1(\mathbf{x}))(b_2(\mathbf{x})^T b_2(\mathbf{x})) - (b_1(\mathbf{x})^T b_2(\mathbf{x}))^2 -} = \\ [\text{see lemma 1.3.3}] = \pm\lambda_D.$$

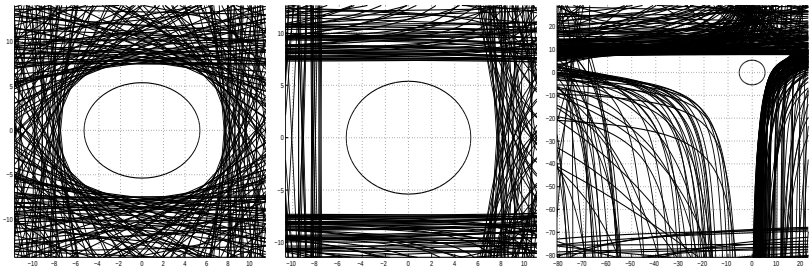


Figure 1.6: The intersection of three arbitrarily chosen hyperplanes and $\Omega_{\mathbf{T}}^{+}$ along with the resulting inscribed sphere condition of theorem 1.4.1.

An example of a maximum-volume sphere conditions for a generic source configuration $\mathbf{T}$ is shown in fig. 1.6.

1.4.2 Maximum-Volume Inscribed Ellipsoid

The condition from the previous section can be improved by instead finding the maximum-volume ellipsoid $\mathscr{E} = \left\{ d \in \mathbb{R}^{2n} \mid d^T A d + 2a^T d - 1 < 0 \right\}$ inscribed in $\Omega_{\mathbf{T}}^{+}$. Finding such extremal volume ellipsoids can be formulated as optimization problems [8, 5].

$$\begin{aligned} &\max \quad \text{volume of } E \\ &\text{s.t. } E \subset C(\mathbf{x}), \forall \ \mathbf{x} \in \mathbb{R}^2. \end{aligned}$$

However, since there are finitely many variables and an infinite number of constraints this is a semi-infinite program [4]. In order to avoid this we approximate $\Omega_{\mathbf{T}}^{+}$ by the intersection of a finite subset of these constraints,

$$\tilde{C}_i = \left\{ d \in \mathbb{R}^{2n} \mid d^T B_i d + 2b_i^T d + 2 > 0, \ i = 1...L, \ \ B_i = B_{\mathbf{T}}(\mathbf{x}_i), \ \mathbf{x}_i \in \mathbb{R}^2 \right\}.$$

Using that the volume of E is proportional to $\log(\det A)$ the maximum-volume inscribed ellipsoid optimization problem can be formulated.

Lemma 1.4.1. *The ellipsoid* $\mathscr{E} = \left\{ d \in \mathbb{R}^{2n} \mid d^T A^* d + 2a^{*T} d - 1 < 0 \right\}$ *where* A^* *and*

a^* are the global optimizers of

$$\begin{aligned}
&\min\ \log(\det A)\\
&\text{s.t.} \begin{bmatrix} B_i & b_i \\ b_i^T & 2 \end{bmatrix} - \tau_i \begin{bmatrix} -A & a \\ a^T & 1 \end{bmatrix} \succeq 0,\\
&\tau_i \geq 0,\\
&i = 1...L
\end{aligned}$$

is the maximum-volume ellipsoid inscribed in $\bigcap_{i=1}^{L} \tilde{C}_i$.

Proof. The volume of an ellipsoid on the given form is inversely proportional to $\log\det(A)$. The constraints follows directly from the S-procedure, see the proof of theorem 1.4.1. □

This is a non-linear program with a convex objective function and bilinear matrix inequality constraints. It can been shown [8] that it is a convex program if $\Omega_{\mathbf{T}}^{+}$ is a convex set. Following that this formulation is less constrained than lemma 1.4.1, the ellipsoid $\mathscr{E}$ should provide a superior sufficient constraint on $\mathbf{Y}$ for bijectivity. However, the disadvantage of this approach is that it involves a more computationally complex optimization problem.

1.4.3 Improving Sufficient Conditions for Bijectivity

The sufficient conditions derived in sections 1.4.1 and 1.4.2 are on a very compact and simple form but can in cases be overly tight. Using properties discussed in section 1.3, such convex bounded quadratic constraints can be further improved while still keeping their appealing representation.

First the following lemma that connects the null space of $B_{\mathbf{T}}(\mathbf{x})$ to bijective target configurations is formulated.

Lemma 1.4.2. *If $\mathbf{Y}$ gives a bijective bijective mapping, that is $\mathbf{Y} \in \Omega_{\mathbf{T}}^{+}$, so do all points in the hyperplane $\mathbf{Y} + \begin{bmatrix} \mathbf{1}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix}$,* where $\mu, \nu \in \mathbb{R}$

Proof. If $\mathbf{Y} \in \Omega_{\mathbf{T}}^{+}$ then $\mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} > 0, \forall x \in \mathbb{R}^2$.

$$
\begin{aligned}
&(\mathbf{Y} + \begin{bmatrix} \mathbf{1}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix})^T B_{\mathbf{T}}(\mathbf{x})(\mathbf{Y} + \begin{bmatrix} \mathbf{1}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix}) = \\
&= \mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} + 2\mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x}) \left(\begin{bmatrix} \mathbf{1}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix} \right) + \\
&+ \left(\begin{bmatrix} \mathbf{1}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix} \right)^T B_{\mathbf{T}}(\mathbf{x}) \left(\begin{bmatrix} \mathbf{1}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix} \right) = \\
&= [\text{ from lemma 1.3.4 we know that } \begin{bmatrix} \mathbf{1}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_n \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix} \text{ is in the null space of} \\
&B_{\mathbf{T}}(\mathbf{x}) \text{for all } x \in \mathbb{R}^2] = \mathbf{Y}^T B_{\mathbf{T}}(\mathbf{x})\mathbf{Y} > 0, \ \forall x \in \mathbb{R}^2. \qquad (1.39)
\end{aligned}
$$

□

To each bijective configuration there is an entire set of associated configurations, guaranteed also to be bijective. This, in conjunction with the cone properties of $\Omega_{\mathbf{T}}^{+}$, allows for the extension of any convex, bounded quadratic sufficient constraint as in the ensuing theorem.

Theorem 1.4.3. *If the ellipsoid $\mathscr{E} = \{y | y^T A y + 2a^T y + c < 0, \ y \in \mathbb{R}^{2n}\}$ is contained in $\Omega_{\mathbf{T}}^{+}$ then so is the set*

$$K = \{y | y^T \tilde{A} y < 0, \ y \in \mathbb{R}^{2n}\}, \qquad (1.40)$$

where

$$
\begin{aligned}
\tilde{A} = \Big(&(a^T G^{-T} E E^T G^{-1} a + c)(A - GEE^T G^T) - \\
&-(I - GEE^T G^{-1}) a a^T (I - (GEE^T G^{-1})^T) \Big) \qquad (1.41)
\end{aligned}
$$

and

$$E = \begin{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1}_n & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{n}}\mathbf{1}_n \end{bmatrix}.$$

Here G is the upper-triangular matrix from the Cholesky-factorization of $A = GG^T$.
The set K is a double cone with the origin removed, it contains $\mathscr{E}$ and is also in $\Omega_{\mathbf{T}}^{+}$, i.e.

$$\mathscr{E} \subset K \subset \Omega_{\mathbf{T}}^{+}.$$

Proof. From the cone property of $\Omega_{\mathbf{T}}^{+}$ from lemma 1.3.1, we know that if $y \in \Omega_{\mathbf{T}}^{+}$ then the entire line λy, $\lambda \in \mathbb{R}$ is also in $\Omega_{\mathbf{T}}^{+}$, except at the origin. Combined with lemma

1.4.2 this means that if $y \in \Omega_{\mathbf{T}}^{+}$ then the linear hull

$$L_y = \left\{ \begin{bmatrix} \mathbf{1}_n & \mathbf{0} & y \\ \mathbf{0} & \mathbf{1}_n & \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \nu \end{bmatrix} \mid \forall\, \lambda, \mu, \nu \in \mathbb{R} \right\}$$

is a subset of $\Omega_{\mathbf{T}}^{+}$.

An open ball S centered at m with radius r can be written

$$S = \{y | (y-m)^T (y-m) < r^2,\ y \in \mathbb{R}^{2n}\}.$$

That is, a point y is in S if its distance to m is less than r.

If $S \subset \Omega_{\mathbf{T}}^{+}$ then $y \in \Omega_{\mathbf{T}}^{+}$ if L_y intersects S, i.e. the distance from m to L_y is less than r. An orthogonal basis for L_y can be written

$$F = \begin{bmatrix} \underbrace{\begin{matrix} \frac{1}{\sqrt{k}}\mathbf{1}_k & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{k}}\mathbf{1}_k \end{matrix}}_{E} & \tilde{y} \end{bmatrix}.$$

Here

$$\tilde{y} = \frac{(I - EE^T)}{\sqrt{y^T (I - EE^T)^T (I - EE^T) y}} y = \frac{(I - EE^T)}{\sqrt{y^T (I - EE^T) y}} y.$$

The distance $d(m, L_y)$ between m and the hyperplane L_y is the lenght of the vector v

$$v = m - FF^T m = (I - FF^T) m.$$

Thus we obtain

$$\begin{aligned} d(m, L_y)^2 &= v^T v = m^T (I - FF^T)^T (I - FF^T) m = \\ &= m^T (I - 2FF^T + FF^T FF^T) m = m^T (I - 2FF^T + FF^T) m = \\ &= m^T (I - FF^T) m. \end{aligned}$$

The constraint $d^2 < r$ then becomes

$$\begin{aligned} d(m, L_y)^2 &= m^T (I - FF^T) m = \\ &= m^T \left(I - \begin{bmatrix} E & \frac{(I-EE^T)}{\sqrt{y^T (I-EE^T) y}} y \end{bmatrix} \begin{bmatrix} E^T \\ \left(\frac{(I-EE^T)}{\sqrt{y^T (I-EE^T) y}} y \right)^T \end{bmatrix} \right) m = \\ &= m^T \left((I - EE^T) - \frac{(I - EE^T) y y^T (I - EE^T)^T}{y^T (I - EE^T) y} \right) m < r^2. \end{aligned}$$

Simplyfying

$$
\begin{aligned}
&m^T(I-EE^T)my^T(I-EE^T)y- \\
&\qquad -m^T(I-EE^T)yy^T(I-EE^T)^Tm < r^2y^T(I-EE^T)y \\
&y^T((m^T(I-EE^T)m)(I-EE^T)- \\
&\qquad (I-EE^T)mm^T(I-EE^T)^T - r^2(I-EE^T))y < 0 \\
&y^T((m^T(I-EE^T)m - r^2)(I-EE^T)- \\
&\qquad -(I-EE^T)mm^T(I-EE^T)^T)y < 0 \\
&y^T\Big((m^T(I-EE^T)m - r^2)I - (I-EE^T)mm^T\Big)\Big(I-EE^T\Big)y < 0. \qquad (1.42)
\end{aligned}
$$

Eq. (1.42) can then be generalised to handle ellipsoidal constraints on the form

$$\mathscr{E} = \{y | y^TAy + 2a^Ty + c < 0,\ y \in \mathbb{R}^{2n}\}.$$

Here A is a symmetric and positive definite matrix so it has a Cholesky decomposition $A = GG^T$ as well as an inverse. Using this we can write

$$
\begin{aligned}
&y^TAy + 2a^Ty + c = (y + A^{-1}a)^TA(y + A^{-1}a) + \underbrace{(-a^TA^{-1}a + c)}_{\tilde{c}} = \\
&= (G^Ty + G^T(GG^T)^{-1}a)^T(G^Ty + G^T(GG^T)^{-1}a) + \tilde{c} = \\
&= (\underbrace{G^Ty}_{\tilde{y}} + \underbrace{G^{-1}a}_{\tilde{m}})^T(G^Ty + G^{-1}a) + \tilde{c} = \\
&= (\tilde{y} + \tilde{m})^T(\tilde{y} + \tilde{m}) + \tilde{c}.
\end{aligned}
$$

Inserting this into (1.42) with $r^2 = -\tilde{c}$ gives

$$
\begin{aligned}
&\tilde{y}^T\Big((\tilde{m}^T(I-EE^T)\tilde{m} + \tilde{c})I - (I-EE^T)\tilde{m}\tilde{m}^T\Big)\Big(I-EE^T\Big)\tilde{y} = \\
&= y^TG\Big((a^TG^{-T}(I-EE^T)G^{-1}a - a^TA^{-1}a + c)I- \\
&\quad -(I-EE^T)G^{-1}aa^TG^{-T}\Big)\Big(I-EE^T\Big)G^Ty = \\
&\quad = y^T\Big((a^TG^{-T}EE^TG^{-1}a + c)(A - GEE^TG^T)- \\
&\quad -(I-GEE^TG^{-1})aa^T(I-(GEE^TG^{-1})^T)\Big)y < 0. \qquad (1.43)
\end{aligned}
$$

□

In the case of the maximum-volume inscribed sphere this results in the following corollary.

Corollary 1.4.2. *A thin-plate spline mapping $\phi_{\mathbf{T},\mathbf{Y}} : \mathbb{R}^2 \to \mathbb{R}^2$ with n point constraints $\mathbf{T}$, $\mathbf{Y}$ and $c = \frac{1}{\sqrt{\max_{\mathbf{x}\in\mathbb{R}^2}(\lambda_M(\mathbf{x}))}}$, as defined in theorem 1.4.1, is bijective if*

$$Y^T\Big((\mathbf{T}^T EE^T\mathbf{T} + c) - (I - EE^T)\mathbf{T}\mathbf{T}^T\Big)(I - EE^T)\mathbf{Y} < 0. \tag{1.44}$$

Proof. Follows trivially from insertion of theorem 1.4.1 into eq. (1.40). □

1.5 Conclusion

Even though this section does not provide a complete theory on the set of bijective thin-plate spline mappings, it does contain a formulation of how to characterize this set, as well as proofs of many of its properties. It also includes a discussion of some experimentally derived indications of other atributes of this set, as well as methods for finding sufficient conditions for bijectivity. Future work includes finding such conditions analytically as well as attempting to further determine its convexity and boundness properties.

Bibliography

[1] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11, 1989.

[2] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. *Constructive Theory of Functions of Several Variables*, 1987.

[3] P.J. Green and B.W. Silverman. *Nonparametric Regression and Generalized Linear Models*. Number 58 in Monographs on Statistics and Applied Probability. Chapman & Hall, 1994.

[4] R. Hettich and K.O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35(3), 1993.

[5] S. Boyd L. Vandenberghe and S.W. Wu. Determinant Maximization with Linear Matrix Inequality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.

[6] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics*, 30, 1979.

[7] E. Feron S. Boyd, L. El Ghaoui and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

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

[9] G. Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990.

PAPER V

In *Proceedings International Conference on Pattern Recognition,*
Hong Kong, China 2006.

Main Entry: reg · is · tra · tion
Pronunciation:\,re-jə-'strä-shən\
Function: noun
Origin: 1350-1400; from late Latin *regesta* "list, matters recorded"; from Latin *regerere* "to record" literally "to carry back"; from *re-* "back" + *gerere* "carry, bear". Some senses influenced by association with Latin *regere* "to rule."

1: the act of registering
2: an entry in a register
3: the number of individuals registered : enrollment
4 a: the art or act of selecting and adjusting pipe organ stops
b: the combination of stops selected for performing a particular organ work
5: a document certifying an act of registering

Bijective Image Registration using Thin-Plate Splines

Anders Eriksson and Kalle Åström

Centre for Mathematical Sciences
Lund University, Sweden

Abstract

Image registration is the process of geometrically aligning two or more images. In this paper we describe a method for registering pairs of images based on thin-plate spline mappings. The proposed algorithm minimizes the difference in gray-level intensity over bijective deformations. By using quadratic sufficient constraints for bijectivity and a least squares formulation this optimization problem can be addressed using quadratic programming and a modified Gauss-Newton method. This approach also results in a very computationally efficient algorithm. Example results from the algorithm on three different types of images are also presented.

1.1 Introduction.

This paper addresses the problem of image registration. It is the process of geometrically aligning two or more images and has been the subject of extensive research over the last decade, see [7]. This field is widely applied in computer vision, remote sensing and medical imaging.

The approach presented here is based on the thin-plate spline mapping, a commonly used method for deforming images. Using this mapping we wish to find dense and bijective correspondences between pairs of images. In computer vision, non-linear mappings in $\mathbb{R}^2$ of this sort are frequently used to model deformations in images. The underlying assumption is that all the images contain similar structures and therefore there should exist mappings between pairs of images that are both one-to-one and onto, i.e. bijective.

The contribution of this paper is in addition to highlighting of some interesting properties of the thin-plate spline mapping also the incorporation of sufficient quadratic conditions for bijectivity into that framework. A description of how to combine this into a simple but efficient algorithm based on a least-square minimization formulation is also provided. Similar methods have been proposed [5], however without addressing the issue of bijectivity.

1.2 Thin-Plate Spline mappings.

Thin-plate splines are a class of widely used non-rigid spline interpolating functions. It is a natural choice of interpolating function in two dimensions and has been a commonly used tool in computer vision for years. Introduced and developed by Duchon [3] and Meinguet [6] and popularized by Bookstein [1], its attractions include an elegant mathematical formulation along with a very natural and intuitive physical interpretation.

Mappings of this type are constructed by combining two thin-plate interpolants, each describing the x- and y-displacements respectively, a new function, the thin-plate spline mapping $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ can be constructed. Given a set $\mathbf{T}$ of k control points in $\mathbb{R}^2$ and a set $\mathbf{Y}$ of k destination points also in $\mathbb{R}^2$. It has been shown that such a bivariate function ϕ that fulfills $\phi_{\mathbf{T},\mathbf{Y}}(t_i) = y_i,\ i = 1..k$ is in the form (for details see [1])

$$\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathbf{s}(\mathbf{x}) & 1 & \mathbf{x} \end{bmatrix} \begin{bmatrix} W^T \\ c \\ A \end{bmatrix}, \tag{1.1}$$

where

$$\sigma(h) = ||h||^2 \log(||h||), \tag{1.2}$$

$$\mathbf{s}(\mathbf{x}) = [\sigma(|\mathbf{x} - \mathbf{t_1}|)...\sigma(|\mathbf{x} - \mathbf{t_k}|)], \tag{1.3}$$

$$\begin{bmatrix} W & c^T & A^T \end{bmatrix} = \begin{bmatrix} \mathbf{Y}^T \mathbf{0}\ \mathbf{0} \end{bmatrix} \Gamma^{-1}. \tag{1.4}$$

with

$$\Gamma = \begin{bmatrix} S & 1_k & T \\ 1_k^T & 0 & 0 \\ T^T & 0 & 0 \end{bmatrix},\ (S)_{ij} = \sigma(|\mathbf{t}_i - t_j|) \tag{1.5}$$

Combining eq. 1.1 and 1.4, and with Γ^{ij} the block partitioning of Γ^{-1}, the transformation can be written

$$\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x}) = \begin{bmatrix} s(\mathbf{x})^T & 1 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \mathbf{Y}. \tag{1.6}$$

This gives us a deformation $\phi_{\mathbf{T},\mathbf{Y}}$ that for a fixed set of control points $\mathbf{T}$ is parameterized linearly by the destination points $\mathbf{Y}$.

1.3 Thin-Plate Spline Based Image Registration.

The registration of two images requires finding the deformation of the first image that makes it as similar as possible to the second image. Here, the non-linear deformation used is the thin-plate spline mapping and the similarity function will simply be the sum of squared differences in gray-level intensity.

Denote the image to be warped $I(x, y)$, the reference image $I_{ref}(x, y)$ and the thin-plate spline mapping by
$\phi_{\mathbf{T}}(\mathbf{x}, \mathbf{Y})$. Introducing the finite set $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ of points where the two images are to be compared, typically all the pixel positions of the reference image, the similarity function can then be written

$$f(\mathbf{Y}) = \sum_{i=1}^{N} (r_i(\mathbf{Y}))^2 = \sum_{i=1}^{N} (I(\phi_{\mathbf{T}}(x_i, \mathbf{Y})) - I_{ref})^2. \tag{1.7}$$

Minimizing such sum of squares is a frequently occurring problem and a number of methods exist that take advantage of its particular structure.

The Gauss-Newton method addresses the problem in a very simple but appealing manner. This iterative algorithm converges linearly towards the minima if the starting point is sufficiently close. With the Jacobian of $r(\mathbf{Y}) = [r_1(\mathbf{Y})...r_N(\mathbf{Y})]$ defined as the $N \times 2n$ matrix $(J(\mathbf{Y}))_{ij} = (\frac{\partial r_i}{\partial \mathbf{Y}_j})$, the gradient and Hessian of eq. 1.7 can be written

$$\nabla f(\mathbf{Y}) = 2J(\mathbf{Y})^T r_i(\mathbf{Y}), \tag{1.8}$$

$$H(\mathbf{Y}) = J(\mathbf{Y})^T J(\mathbf{Y}) + 2\sum_{i=1}^{N} r_i(\mathbf{Y}) \nabla^2 r_i(\mathbf{Y}). \tag{1.9}$$

In order to avoid having to compute the Hessian $\nabla^2 r_i(\mathbf{Y})$ in every iteration the second part of eq. 1.9 is assumed small and is simply neglected and $H(\mathbf{Y}) \approx \tilde{H}(\mathbf{Y}) = J(\mathbf{Y})^T J(\mathbf{Y})$.

Now by approximating $f(\mathbf{Y})$ by its second-order Taylor expansion of degree near $\mathbf{Y}_k$we get

$$\begin{aligned} f(\mathbf{Y}) \approx f(\mathbf{Y}_k) + \nabla f(\mathbf{Y}_k)^T (\mathbf{Y} - \mathbf{Y}_k) + \\ \frac{1}{2}(\mathbf{Y} - \mathbf{Y}_k)^T \tilde{H}(\mathbf{Y}_k)(\mathbf{Y} - \mathbf{Y}_k). = \tilde{f}(\mathbf{Y}) \end{aligned} \tag{1.10}$$

The unconstrained minimization of this quadratic approximation of the objective function $\tilde{f}(\mathbf{Y})$ is carried out by the normal equation,

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k - (J(\mathbf{Y}_k) J(\mathbf{Y}_k)^T) J(\mathbf{Y}_k) r_i(\mathbf{Y}_k). \tag{1.11}$$

By applying this method iteratively $\mathbf{Y}_k$ will then converge to a local minima of $f(\mathbf{Y})$.

However, since we want to minimize eq. 1.7 over bijective mappings only, a slight alteration of this method is required. From [4] we can obtain convex quadratic sufficient constraints on $\mathbf{Y}$ for bijectivity of the mapping $\phi_{\mathbf{T},\mathbf{Y}}(\mathbf{x})$ on the form $\mathbf{Y}^T A\mathbf{Y} + b^T\mathbf{Y} + c \geq 0$. As the minimization of eq. 1.10 is now no longer unconstrained the final step of the original Gauss-Newton method is replaced by the quadratically constrained quadratic program, also convex if $H(\mathbf{Y}_k)$ is positive definite

$$\begin{aligned} \min \quad & \tilde{f}(\mathbf{Y}_k) = f(\mathbf{Y}_k) + \nabla f(\mathbf{Y}_k)^T(\mathbf{Y} - \mathbf{Y}_k) + \\ & + \tfrac{1}{2}(\mathbf{Y} - \mathbf{Y}_k)^T H(\mathbf{Y}_k)(\mathbf{Y} - \mathbf{Y}_k) \\ \text{s.t.} \quad & \mathbf{Y}^T A\mathbf{Y} + b^T\mathbf{Y} + c \geq 0. \end{aligned}$$

The solution $\mathbf{Y}^*$ of this optimization is taken as the next point in the iteration

At each iteration of the modified Gauss-Newton method requires the computation of $r(\mathbf{Y}) = [r_1(\mathbf{Y})...r_N(\mathbf{Y})]^T$ and $J(\mathbf{Y})$. This can be done very efficiently. Using eq. 1.6 the mapping of all points in X can be written

$$\begin{aligned} & \begin{bmatrix} \phi_{\mathbf{T}}(\mathbf{x}_1, \mathbf{Y}) \\ \vdots \\ \phi_{\mathbf{T}}(\mathbf{x}_n, \mathbf{Y}) \end{bmatrix} = \\ & = \underbrace{\begin{bmatrix} \begin{bmatrix} \mathbf{s}(\mathbf{x_1})^{\mathbf{T}} & 1 & x_{11} & x_{12} \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \mathbf{s}(\mathbf{x_n})^{\mathbf{T}} & 1 & x_{(n)1} & x_{(n)2} \end{bmatrix} \begin{bmatrix} \Gamma^{11} \\ \Gamma^{21} \end{bmatrix} \end{bmatrix}}_{H_{\mathbf{T},X}} \mathbf{Y} = \\ & = H_{\mathbf{T},X}\mathbf{Y}. \end{aligned} \tag{1.12}$$

Since the $N \times 2n$ matrix $H_{\mathbf{T},X}$ is not dependent of $\mathbf{Y}$ it can be precomputed, reducing the computation of the mapping of X by $\phi(\mathbf{Y}_k)$ to a single matrix multiplication. This then allows for an efficient calculation of the deformed image. The jacobian of r_i is also needed,

$$\begin{aligned} & (J(\mathbf{Y}))_{ij} = \frac{\partial}{\partial \mathbf{Y}_j}\left(I(\phi(x_i, \mathbf{Y})) - I_{ref}\right) = \\ & I'_x(\phi(x_i, \mathbf{Y}))\frac{\partial}{\partial \mathbf{Y}_j}\phi_1(x_i, \mathbf{Y}) + I'_y(\phi(x_i, \mathbf{Y}))\frac{\partial}{\partial \mathbf{Y}_j}\phi_2(x_i, \mathbf{Y}). \end{aligned} \tag{1.13}$$

Here I'_x and I'_y are the horizontal and vertical components of the gradient of I. Further-

more, since the mapping $\phi_{\mathbf{T}}(x, \mathbf{Y})$ is linear in $\mathbf{Y}$ its partial derivatives are all constant

$$\begin{aligned} \phi_{\mathbf{T}}(X, \mathbf{Y}) = [\phi_1(X, \mathbf{Y})\ \phi_2(X, \mathbf{Y})] = H_{\mathbf{T},X}[\mathbf{Y}_1\ \mathbf{Y}_2] \Rightarrow \\ \Rightarrow \begin{cases} \frac{\partial}{\partial \mathbf{Y}_j}\phi_1(x_i, \mathbf{Y}) = \left(\begin{bmatrix} H_{\mathbf{T},X} \\ 0 \end{bmatrix} \right)_{ij}, \\ \frac{\partial}{\partial \mathbf{Y}_j}\phi_2(x_i, \mathbf{Y}) = \left(\begin{bmatrix} 0 \\ H_{\mathbf{T},X} \end{bmatrix} \right)_{ij}. \end{cases} \end{aligned} \tag{1.14}$$

So eq. 1.13 can be computed through componentwise multiplications of elements from $I'_x(\phi(x_i, \mathbf{Y}))$, $I'_y(\phi(x_i, \mathbf{Y}))$ and $H_{\mathbf{T},X}$ Combining all of the above then enables us to write the proposed algorithm as follows.

Algorithm for thin-plate spline based image registration.

1. **Pre-computation.**
   For a given thin-plate spline source configuration $\mathbf{T}$ and a pair of images I and I_{ref} to be compared at a finite number of positions $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ compute the following:
   - The image gradient, $\nabla I = (\frac{\partial I}{\partial x},\ \frac{\partial I}{\partial y}) = [I'_x, I'_y]$.
   - The matrix $H_{\mathbf{T},X}$ from eq. (1.12)
   - The quadratic bijectivity constraints on $\mathbf{Y}$ for $\mathbf{T}$, according to [4]
2. **Initialization.**
   Choose an starting point $\mathbf{Y}_0$ for the algorithm. Either by employing some coarse search method or by simply selecting $\mathbf{Y}_0 = \mathbf{T}$, the unit deformation.
   Set $k = 0$.
3. **Iteration start.**
   - Compute $\phi^k_{\mathbf{T}}(X, \mathbf{Y}_k) = H_{\mathbf{T},X}\mathbf{Y}_k$.
   - Find $I(\phi^k_{\mathbf{T}}(X, \mathbf{Y}_k))$, $I(\phi^k_{\mathbf{T}}(X, \mathbf{Y}_k))$ and $I(\phi^k_{\mathbf{T}}(X, \mathbf{Y}_k))$.
   - Calculate the residual $r_i = I(\phi^k_{\mathbf{T}}(X, \mathbf{Y}_k)) - I_{ref}$.
   - Use eq. 1.13 to determine the Jacobian $J(\mathbf{Y}_k)$.
   - Compute the gradient $\nabla f(\mathbf{Y}_k)$ and the approximated Hessian $\tilde{H}(\mathbf{Y}_k)$ according to eqs. (1.8) and (1.9)
4. **Optimization.**
   Find the solution $\mathbf{Y}^*$ to the quadratically constrained quadratic program

$$\begin{aligned} \min \quad & \tilde{f}(\mathbf{Y}) \\ \text{s.t.} \quad & \mathbf{Y}^T A\mathbf{Y} + b^T\mathbf{Y} + c \geq 0 \end{aligned}$$

5. **Parameter update.**
   Set $\mathbf{Y}_{k+1} = \mathbf{Y}^*$ and $k = k + 1$.

6. **Return to 3.**

1.4 Experimental Results.

We applied the suggested registration algorithm on three different types of images. First, a pair of simple, artificially constructed images. Second, two magnetic resonance images of a human brain, the types of images in medical imaging where image registration techniques are commonly applied. Finally, we attempted the registration of a pair of images of human faces. In this case the initial assumption of dense one-to-one mappings does not necessarily hold as self-occlusion can easily occur for these types of images. However, bijective registrations of natural objects like faces is still of great interest, for instance in the automatic construction of the Active Appearance Models of [2].

For these experiments a source configuration $\mathbf{T}$ as a regular rectangular 10×10 grid was used. The quadratic constraint was pre-computed and used in all three instances. The images used were roughly 100×100 pixels in size. On a standard personal computer the entire registration procedure, including all pre-computations except for the bijectivity constraints, took approximately 60 seconds. The results can be seen in figs 1.1, 1.2 and 1.3.

In these three experiments our algorithm converges to at least a satisfactory registration of the image pairs. The artificial images are overlayed very accurately, as would be expected. The images of the faces are also successfully registered, differences are slight but distinguishable. We believe that this is the result of fundamental dissimilarities between the images, such as inconsistent lighting. However, in the case of the two magnetic resonance images of a human brain the registration process is not entirely successful. Some of the discernable features does not seem to have been correctly overlayed. We assume that this is caused by shortcomings inherent in our algorithm. Firstly, and this was briefly mentioned earlier, some of the assumptions the Gauss-Newton method, on which our approach is based, makes requires that the initial starting point of the algorithm is sufficiently close to the global optima. What constitutes sufficiently close is debateable but is a requirement for the method to converge successfully. Secondly, a 10×10 grid thin-plate spline mapping can only parametrize a subset of all bijective deformations of $\mathbb{R}^2$ and in addition, since the bijectivity conditions of [4] are sufficient but not necessary, we can only reach a subset of this set. This means that our method is perhaps better suited for image registrations requiring smaller deformations. Nevertheless, we do believe that the results presented here still indicate the applicability of such an algorithm.

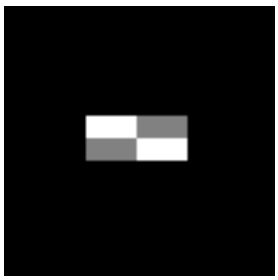
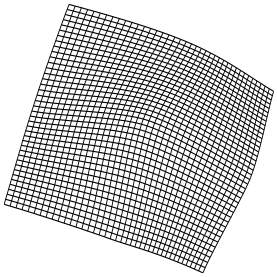


Image I.

Image I_{ref}.



Resulting deformation $\phi_{\mathbf{T}}$.

Resulting registration $I(\phi_{\mathbf{T}})$.

Figure 1.1: Registration of a pair of simple artificial images.

1.5 Concluding Remarks.

In this paper we have presented a method for performing pairwise registration of images. An algorithm, based on the thin-plate spline mapping, for efficiently finding the necessary deformation is proposed. Experiments on three different types of images with promising results were also presented.

Improvements are still achievable. In order to overcome the drawback of the Gauss-Newton method an initial stage to the algorithm should be added. One that performs a larger-scale optimization, for instance over affine deformations only, providing a better starting point for the thin-plate spline mapping optimization. The number and distribution of the control points should also be investigated. More points parametrize a larger subset of the bijective deformations. Obviously, improving the bijectivity constraints could also enhance the performance of the algorithm, but that is perhaps outside the scope of the work carried out in this paper. A different objective function than eq. 1.7 might also improve on our method. Finally, a more efficient representation of the matrix $H_{\mathbf{T},X}$ should be examined, as its size grows quadratically with the size of the image, even for moderately large images the matrix can become unmanageable.

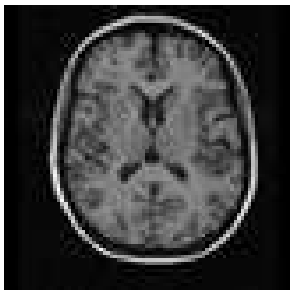


Image I.

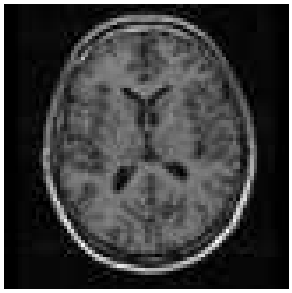
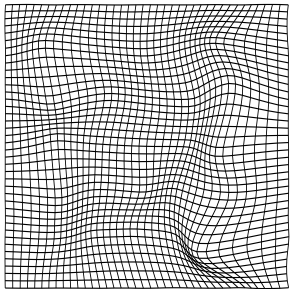
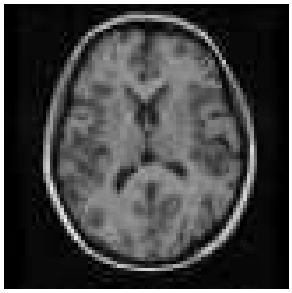


Image i_{ref}.



Resulting deformation $\phi_{\mathbf{T}}$

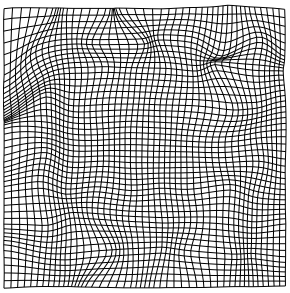


Resulting registration $I(\phi_{\mathbf{T}})$.

Figure 1.2: Registration of a pair of brain MR images.

Image I.

Image I_{ref}.



Resulting deformation $\phi_{\mathbf{T}}$.

Resulting registration $I(\phi_{\mathbf{T}})$.

Figure 1.3: Registration of a pair of images of faces.

Bibliography

[1] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11, 1989.

[2] T.F. Cootes, G.J. Edwards, and Taylor C.J. Active appearance models. *Proc. 5th European Conf. on Computer Vision*, 1998.

[3] J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. *Constructive Theory of Functions of Several Variables*, 1987.

[4] A. Eriksson. Bijective thin-plate spline mappings with applications in computer vision. *Licentiate thesis, Lund University*, 2006.

[5] J. Lim and M.H. Yang. A direct method for modeling non-rigid motion with thin plate spline. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[6] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics*, 30, 1979.

[7] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21, 2003.

PAPER VI

Chapter 4 in Licentitate Thesis
Lund University, 2005.

Main Entry: ap · pear · ance
Pronunciation:\ə-'pir-əns\
Function: noun
Origin: 1350-1400; from French *aper-*; stem of *apareir*; from Latin *apparere*; "to appear"; from *ad-* "to" + *perere* "to come forth, be visible." Appearance "look, aspect" is from c.1385.

1 a: external show : semblance <*although hostile, he preserved an appearance of neutrality*>
b: outward aspect : look <*had a fierce appearance*>
2 a: a sense impression or aspect of a thing <*the blue of distant hills is only an appearance*>
b: the world of sensible phenomena
3 a: the act, action, or process of appearing
b: the presentation of oneself in court as a party to an action often through the representation of an attorney
4 a: something that appears : phenomenon
b: an instance of appearing : occurrence

Groupwise Image Registration and Automatic Active Appearance Model Generation

Anders Eriksson

Centre for Mathematical Sciences
Lund University, Sweden

1.1 Introduction

This work is concerned with groupwise image registration, the simultaneous alignment of a large number of images. As opposed to pairwise registration the choice of a reference image is not equally obvious, therefore an alternate approach must be taken.

Groupwise registration has received equal amounts of attention from the research community as pairwise registration. It has been especially addressed in shape analysis under the name Procrustes analysis, [4]. The areas of application are still remote sensing, medical imaging and computer vision, but now the aggregation of images allows for a greater understanding of their underlying distribution.

The focus in this work is towards a specific task, the use of image registration to automatically construct deformable models for image analysis.

1.2 Automatic Active Appearance Model Generation through Groupwise Image Registration

The outset in this work, that of automatic model construction, is approached by attempting to extend the algorithm of [6] to handle several images. The method chosen for representing deformable models was the widely used Active Appearance Model approach.

Owing to the resemblance between registration of shapes and of images, as formulated here, many of the issues encountered in this section have been considered by the shape analysis community [2] and a number of the ideas presented here are influenced by existing shape matching techniques.

1.2.1 Active Appearance Models

Active Appearance Models (AAM) is a statistical method, introduced by Cootes et al. [1], for interpreting images. From the shape and texture of objects of interest in a number of images, compact models based on the distribution of these features are formed.

The texture, or appearance of the objects are the gray-level image intensities and their shape are usually represented by a finite number of points of correspondence through the entire set of images.

Then, using principal component analysis, eigenspace representations of these two descriptors are extracted. Depending on the application, the shape parameters are generally pre-aligned to eliminate effects from translation, scaling and rotation. By applying yet another principal component analysis, this time to the shape- and appearance parameters combined, an even more concise model describing the joint variability of the objects of interest is achieved. The resulting active appearance model is a compact description of a deformable model based on prior knowledge of the essential characteristics of the object at hand. Through image synthesis, that is by fitting an AAM to unseen images, this approach can be used in a wide variety of image analysis applications.

There is however one disadvantage to this method. The required correspondence calls for manual annotation of landmarks across the entire set of training images. A both tedious and exhausting undertaking. Here an alternative approach is suggested, the automatic generation of Active appearance Models through groupwise image registration.

1.2.2 Groupwise Image Registration

Consider a set of N images $I_1, ..., I_N$, a groupwise registration of this set implies finding deformations $\theta_1, ..., \theta_N,\ \theta_l : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that maximizes the similarity between the corresponding deformed images $I_1(\theta_1), ..., I_N(\theta_N)$. Since registration is carried out with Active Appearance Models in mind, similarity is defined as to what degree an eigenspace method can represent the registered images. Using the squared distance to the eigenspace as a measurement of how well one image is represented by such a statistical model, the total dissimilarity between images $I_1(\theta_1), ..., I_N(\theta_N)$ can be written

$$S(\theta_1, ..., \theta_N) = \sum_{l=1}^{N} (\text{distance between image } I_l(\theta_l) \text{ and } E)^2 = \\ = \sum_{l=1}^{N} ||(I - EE^T)\hat{I}_l(\theta_l)||^2. \qquad (1.1)$$

Here E is the M-dimensional orthogonal basis for a conventional eigenspace representation. The columns of E are the eigenvectors corresponding to the M largest eigenvalues of the covariance matrix of the statistical distribution of the image vectors. As in [6], comparison is made at a finite number of locations in $\mathbb{R}^2$, the set of such locations is

written as $X = \{\mathbf{x}_1, ..., \mathbf{x}_p\}$. The notation $I_l(\mathbf{x})$ is used to represent images both on matrix- and vector-form, the intended form should be evident from the context.

Though this formulation of groupwise registration has the advantage of simplicity, it is actually ill-posed. A global optima of (1.1) is achieved by mapping X bijectively onto one and the same pixel in each image. As this results in zero covariance between the deformed images $S(\theta_1, ..., \theta_N)$ will be equal to zero. This is also an issue in shape analysis and has been identified and addressed by [7, 3, 8]. Here it is simply ignored, the assumption is that if the initial starting point of the algorithm is sufficiently good the degenerate solution will not be attained but instead the optimizer used will terminate in the desired local optima. This vagueness stems from the underlying problem itself. What constitutes as similar objects in images is highly subjective. Hence, the formulation of a method for automatically finding and aligning areas with similar appearance in a number of images will be equally ambiguous.

With the given problem statement we can move on to the proposed method for finding local minima to (1.1). A direct optimizing of this objective function is impractical as this would involve a very large number of variables, N times the number of parameters needed to describe each deformation θ_l. Instead an iterative approach is proposed, by sequentially attending to each image individually, the number of variables in each optimization step can be greatly reduced. That is the repeated minimization of functions

$$S_l = ||(I - EE^T)I_l(\theta_l(X))||^2 = \sum_{j=1}^{p} \Big((I - EE^T)I_l(\theta_l(X))\Big)_j^2. \tag{1.2}$$

Using the thin-plate spline mapping of chapter 2 in [5] to represent the mappings, along with the sum of squares formulation, (1.2) allows for much of the algorithm proposed in [6] to be adopted in groupwise image registration. The assertions made regarding bijective deformation in pairwise image registration are still valid and are hence also applied here. The residual for image l becomes

$$r_l(\mathbf{Y}_l) = (I - EE^T)I_l(\phi(\mathbf{x}, \mathbf{Y}_l)), \tag{1.3}$$

and the corresponding Jacobian

$$\begin{aligned}
&(J_l(\mathbf{Y}_l))_{ij} = \frac{\partial r_{li}}{\partial \mathbf{Y}_{lj}} = \frac{\partial}{\partial \mathbf{Y}_{lj}} \left((I - EE^T)I_l(\phi(x_i, \mathbf{Y}_l))\right)_i = \\
&= \left((I - EE^T)\frac{\partial}{\partial \mathbf{Y}_{lj}} I_l(\phi(x_i, \mathbf{Y}_l))\right)_i \Rightarrow \\
&J_l(\mathbf{Y}_l) = (I - EE^T)\tilde{J}_l(\mathbf{Y}_l).
\end{aligned} \tag{1.4}$$

Here

$$(\tilde{J}_l(\mathbf{Y}_l))_{ij} = \frac{\partial}{\partial \mathbf{Y}_{lj}} I_l(\phi(\mathbf{x}_i, \mathbf{Y}_l)) =$$
$$= I'_{lx}(\phi(\mathbf{x}_i, \mathbf{Y}_l)) \frac{\partial}{\partial \mathbf{Y}_{lj}} \phi_1(\mathbf{x}_i, \mathbf{Y}_l) + I'_{ly}(\phi(\mathbf{x}_i, \mathbf{Y}_l)) \frac{\partial}{\partial \mathbf{Y}_{lj}} \phi_2(\mathbf{x}_i, \mathbf{Y}_l). \quad (1.5)$$

Where $I'_{lx}(\mathbf{x})$ and $I'_{lx}(\mathbf{x})$ as the gradient of $I_l(\mathbf{x})$ and $\frac{\partial \phi_1}{\partial \mathbf{Y}_{lj}}$ and $\frac{\partial \phi_2}{\partial \mathbf{Y}_{lj}}$ defined as in [6].

By adhering to the least square formulation of [6] the algorithm for pairwise image registration can be readily extended to handle groupwise registration as defined here. Neither does this extension make the required computations significantly more demanding, resulting in an algorithm of comparable computational complexity per iteration.

Algorithm for thin-plate spline based groupwise image registration.

1. **Pre-computation.**
   For a given thin-plate spline source configuration $\mathbf{T}$ and N images $I_1, ..., I_N$ to be compared at a finite number of positions $X = \{\mathbf{x}_1, ..., \mathbf{x}_p\}$ compute the following:

   - The gradient of all images.

     $$\nabla I_l = (\frac{\partial}{\partial x} I_l,\ \frac{\partial}{\partial y} I_l) = [I'_{lx}, I'_{ly}].$$

   - The matrix $H_{\mathbf{T},X}$ is defined as in [6]
   - The quadratic bijectivity constraints on $\mathbf{Y}$ for $\mathbf{T}$, according to section 1.4 in [5]

   Note that both $H_{\mathbf{T},X}$ and the bijectivity conditions are independent of which image they are applied to.

2. **Initialization.**
   Choose starting points $\mathbf{Y}_1^0, ..., \mathbf{Y}_N^0$ for the algorithm.
   Compute the initial eigenspace representation E^0 by finding the eigenvectors corresponding to the M largest eigenvalues of the covariance matrix of
   $\left[\ I_1(\phi_{\mathbf{T}}(X, \mathbf{Y}_1^0)) \quad ... \quad I_N(\phi_{\mathbf{T}}(X, \mathbf{Y}_N^0))\ \right]$.
   Set $k = 0$.

3. **Iteration start.**

   - For each image l from 1 to N
     - Compute $\phi_{\mathbf{T}}(X, \mathbf{Y}_l^k) = H_{\mathbf{T},X} \mathbf{Y}_l^k$.

- Find $I_l(\phi_{\mathbf{T}}(X, \mathbf{Y}_l^k))$, $I'_{lx}(\phi_{\mathbf{T}}(X, \mathbf{Y}_l^k))$ and $I'_{ly}(\phi_{\mathbf{T}}(X, \mathbf{Y}_l^k))$.
- Calculate the residual $r_l(\mathbf{Y}_l^k) = (I - E^0(E^0)^T)I_l(\phi_{\mathbf{T}}(X, \mathbf{Y}_l^k))$.
- Use (1.4) to determine the Jacobian $J_l(\mathbf{Y}_k)$.
- Compute the gradient and the approximated Hessian of $S_l(\mathbf{Y}_l^k)$ of (1.2).

$$\nabla S_l(\mathbf{Y}_l^k) = 2J_l(\mathbf{Y}_l^k)^T r_l(\mathbf{Y}_l^k)$$

$$\tilde{H}_l(\mathbf{Y}_l^k) = J_l(\mathbf{Y}_l^k)^T J_l(\mathbf{Y}_l^k)$$

- **Optimization.**
  Find the solution $\mathbf{Y}^*$ to the quadratically constrained quadratic program

$$\begin{aligned} \min \quad & S_l(\mathbf{Y}_l^k) + \nabla S_l(\mathbf{Y}_l^k)^T(\mathbf{Y} - \mathbf{Y}_l^k) + \\ & + \tfrac{1}{2}(\mathbf{Y} - \mathbf{Y}_l^k)^T \tilde{H}_l(\mathbf{Y}_l^k)(\mathbf{Y} - \mathbf{Y}_l^k) \\ s.t. \quad & \mathbf{Y}^T A \mathbf{Y} + b^T \mathbf{Y} + c > 0 \end{aligned}$$

- **Parameter update.**
  Set $\mathbf{Y}_l^{k+1} = \mathbf{Y}^*$.
  $k = k + 1$.

4. **Update the eigenspace representation.**
   Compute E^k from the covariance matrix of
   $\left[\; I_1(\phi_{\mathbf{T}}(X, \mathbf{Y}_1^k)) \quad ... \quad I_N(\phi_{\mathbf{T}}(X, \mathbf{Y}_N^k)) \; \right]$.

5. **Until convergence return to 3.**

1.3 Experimental Results

The proposed algorithm was tested on a set consisting of 400 portrait-style images of male faces, see figure 1.1. A thin-plate spline mapping with 100 control points, evenly spaced on a regular square 10-by-10 grid, was used. As the faces were fairly centered in the images, the initial deformations $\mathbf{Y}_1^0, ..., \mathbf{Y}_{400}^0$ were all set to the identity mapping, centered near the middle of the images, see figure 1.2.

Figure 1.1: A sample of the dataset used.

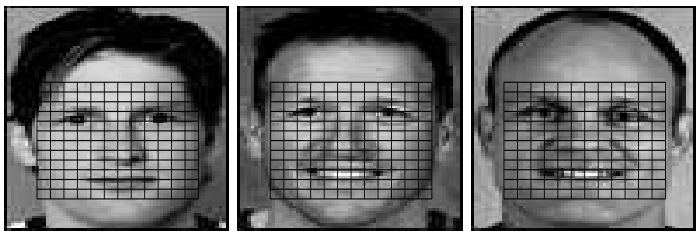


Figure 1.2: Examples of the initial deformations $\mathbf{Y}_l^0$.

The dimension of the eigenspace representation was set to $M = 30$. A set of 1600 points on a 40×40 grid were used as the set of locations for comparison X.

The algorithm described in the preceeding section was applied to the set of images at hand, with the above parameters. A termination criterion simply limiting the number of iterations to 200 was used. The proposed method did converge and sample results can be seen in figure 1.3.

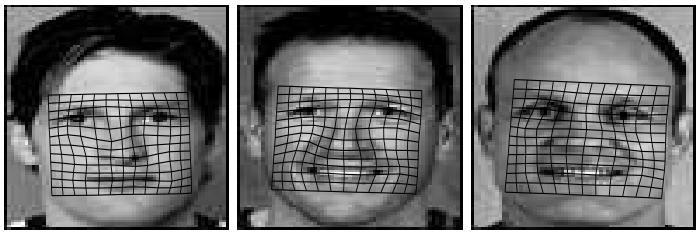


Figure 1.3: Resulting registration for the sample images.

These results are representative of the entire resulting groupwise registration and do indicate the potential of the proposed approach. This can be further realized by examining the evolution of the mean of the registered images after each iteration, $I^k_{mean} = \sum_{l=1}^{M} I_l(\phi_{\mathbf{T}}(X^k, \mathbf{Y}_l))$, see figure 1.4.

Figure 1.4: The evolution of the mean image I^k_{mean}, $(k = 1, 10, ..., 200)$.

Here the increased degree of geometric alignment is clearly seen. To quantify the performance of this algorithm further is difficult, since, as discussed earlier, what is meant by similarity within a set of images is unclear so is the evaluation of groupwise image registration algorithms. Nevertheless, as the outset was the automatic construction of active appearance models, an indication of the quality of the resulting registration could be achieved by examining the performance of the models they produce.

Constructing active appearance models using the proposed approach is extremely straightforward. The required distributions of shape and appearance are given directly by the parameters of the thin-plate spline mappings $\mathbf{Y}_l$ and the deformed images $I_l(\phi_{\mathbf{T}}(X, \mathbf{Y}_l))$.

Using the 400 aligned images an active appearance-like model was constructed. In contrast to [1], here the shape and appearance representations were kept separated in order to be able to ensure bijective deformations in the fitting process as well. With F and E as the eigenspace basis for shape and appearance respectively. The deformation parameters $\mathbf{Y}$ for an individual mapping can be written as

$$\mathbf{Y} = Fy. \tag{1.6}$$

Since this constitutes a subset of $\Omega^+_{\mathbf{T}}$, new and hopefully improved bijectivity conditions ($\tilde{A}$, $\tilde{b}$ and $\tilde{c}$) can be computed. Using the notation from the definition of the registration algorithm the fitting of an active appearance model onto an image $I(\mathbf{x})$ is formulated as minimizing

$$S(y) = ||(I - EE^T)I(\phi_{\mathbf{T}}(X, Fy))||^2 = \sum_{j=1}^{p} \Big((I - EE^T)I(\phi_{\mathbf{T}}(X, Fy)) \Big)_j^2. \tag{1.7}$$

under the condition of bijective deformations. This is solved by the repeated solution of

$$\begin{aligned} \min \quad & S(y^k) + \nabla S(y^k)^T (y - y^k) + \\ & + \tfrac{1}{2}(y - y^k)^T \tilde{H}(y^k)(y - y^k) \\ s.t. \quad & y^T \tilde{A} y + \tilde{b}^T y + \tilde{c} > 0. \end{aligned}$$

An example model-fitting procedure on an image not present in the set of registered images is shown in figure 1.5. Further examples of model adaptations are shown in figures 1.6 and 1.7. These images should be read as follows. The top left images shows the original image with the boundary of the deformed points superimposed. The resulting deformation can be seen at the top right image. The middle row shows, to the left the deformed image $I(\phi_{\mathbf{T}}(X, Fy))$ and to the right its eigenspace representation $EE^T I(\phi_{\mathbf{T}}(X, Fy))$. At the bottom left is the image $I(\phi^{-1}_{\mathbf{T}}(\phi_{\mathbf{T}}(X, Fy), Fy))$, this adds the same interpolation errors introduced in the fitting procedure to the original image as well. This makes the evaluation of the quality of the resulting model fit more unprejudiced. Finally, the bottom right shows the fitted active appearance model overlayed on the original image.

1.4 Conclusion

A method for carrying out non-linear geometric alignment of a large number images, especially geared towards the automatic generation of Active Appearance Models, has

been proposed here. By adhering to the sum of squares formulation of [6] much of the techniques used there could effortlessly be extended to groupwise image registration. The suggested algorithm was tested on a data set of faces and the results were presented. As the nature of the problem is such that the evaluation of its performance is highly subjective, in addition to its ill-posed problem statement. These issues should be addressed by adopting ideas from shape analysis, where similar topics have been investigated. Nevertheless, as the initial results are convincing the presented approach does show promise.

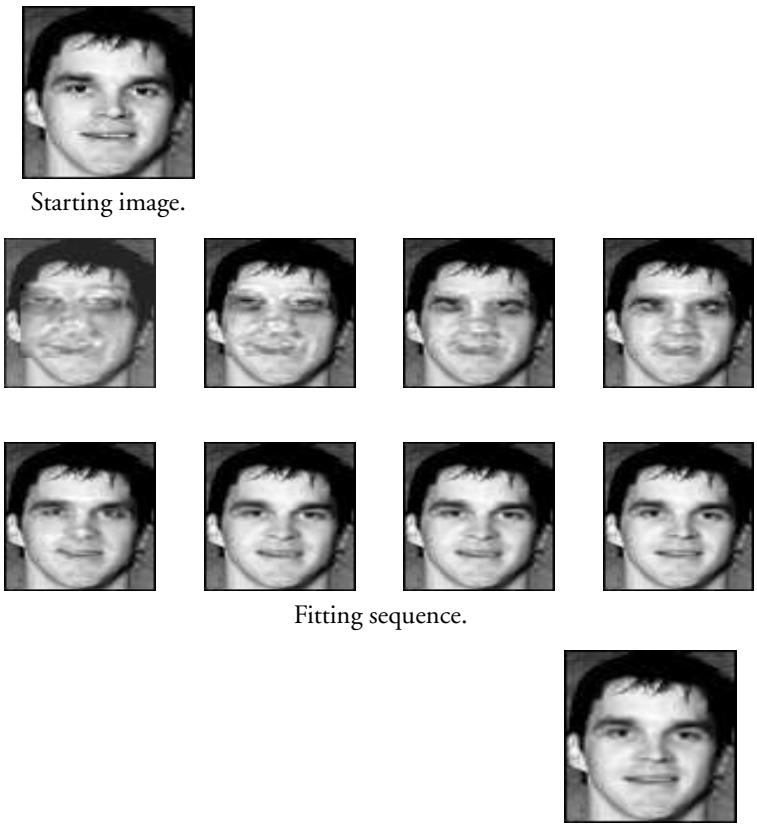


Figure 1.5: An example AAM-fitting. The current model superimposed onto the original after number of different iterations of the proposed fitting algorithm.

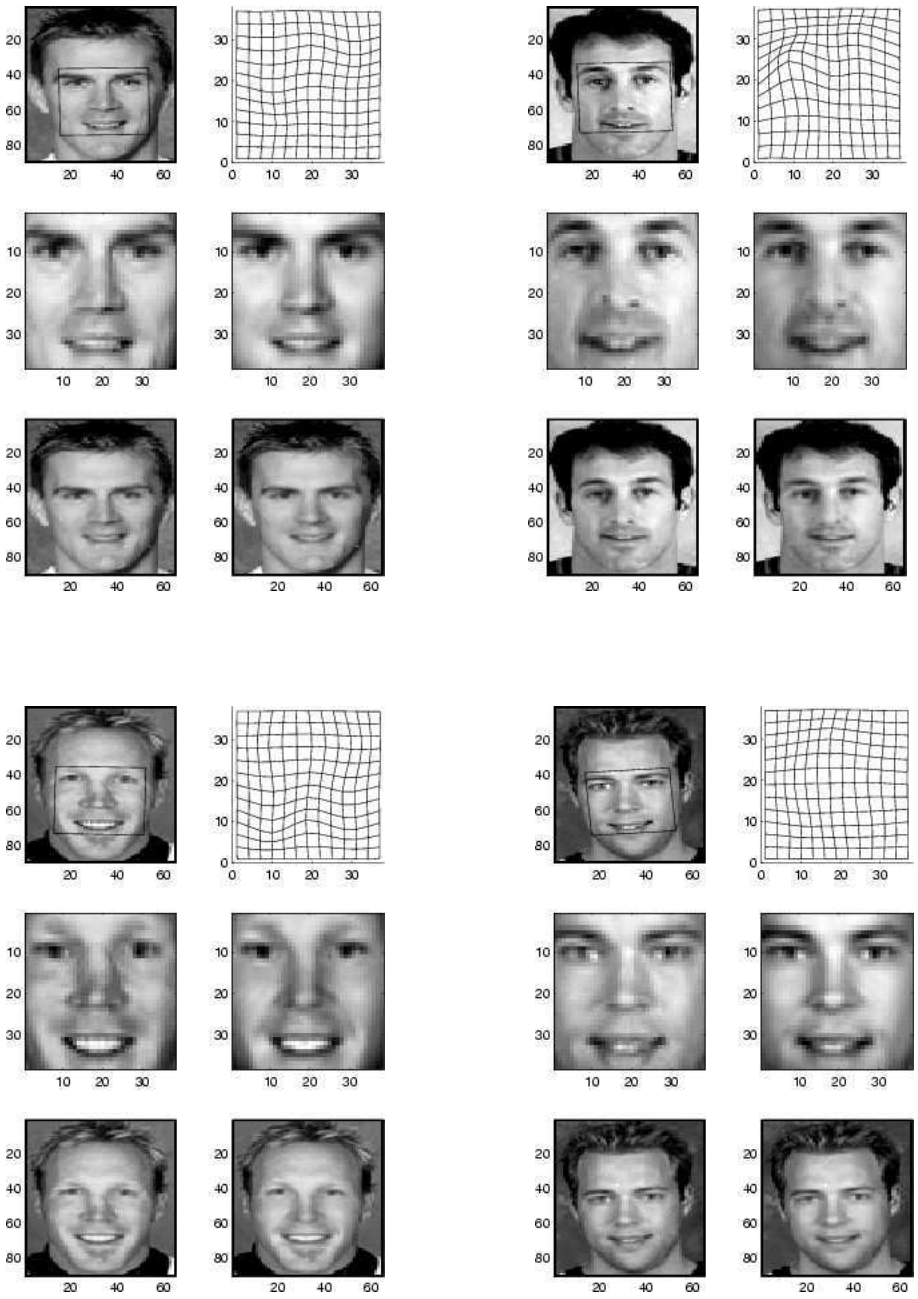


Figure 1.6: Example AAM fittings.

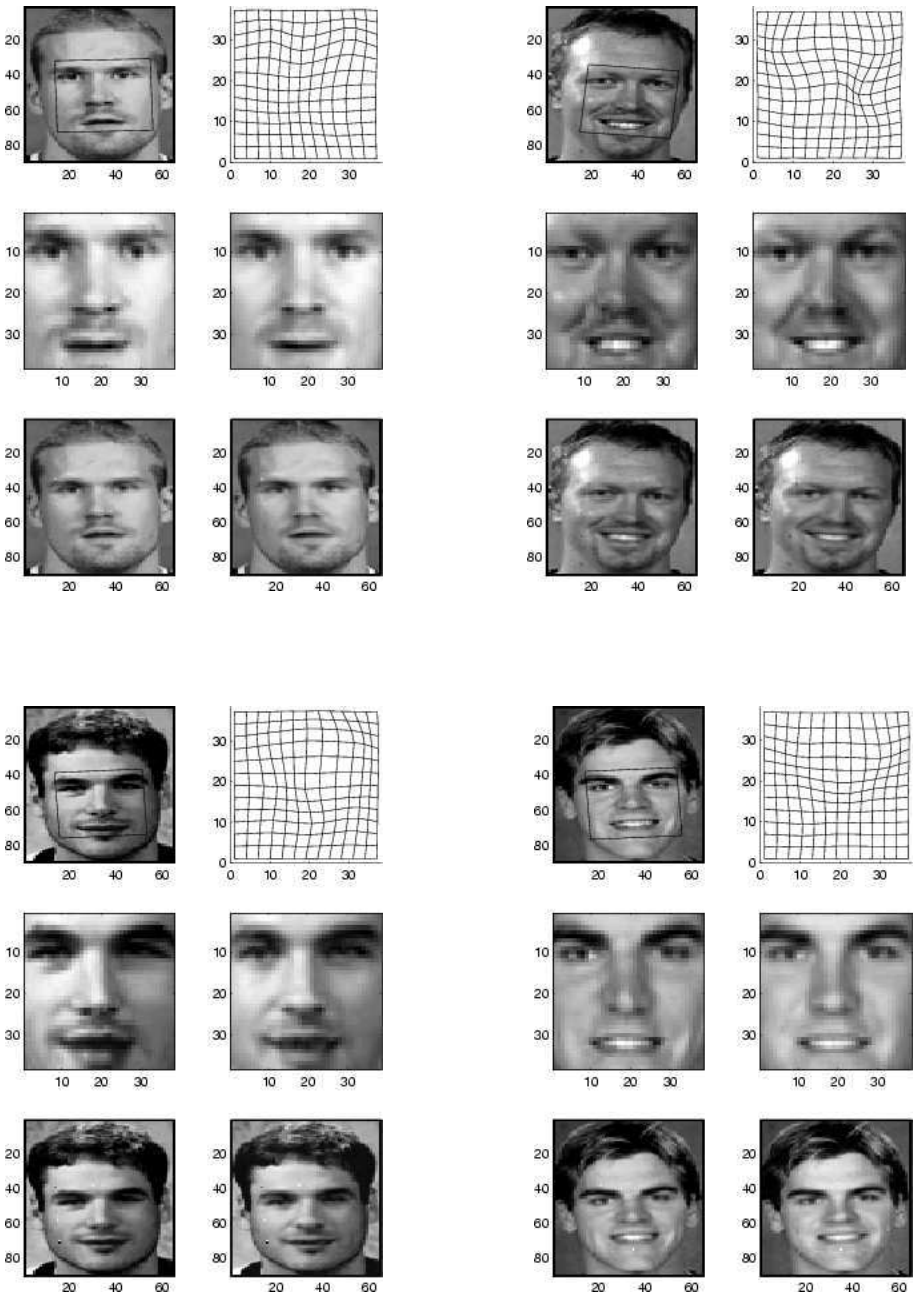


Figure 1.7: Example AAM fittings.

Bibliography

[1] G.J. Cootes, T.F. Edwards and Taylor C.J. Active appearance models. In *Proc. 5th European Conf. on Computer Vision, Freiburg, Germany*, 1998.

[2] T.. Cootes. Statistical models of shape and appearance. Technical report, Imaging Science and Biomedical Engineering, 2004.

[3] R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, and C.J. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Transactions on Medical Imaging*, 21(5):525–537, 2002.

[4] I.L. Dryden and K.V. Mardia. *Statistical Shape Analysis*. John Wiley, 1998.

[5] A. Eriksson. Bijective thin-plate spline mappings with applications in computer vision. *Licentiate thesis, Lund University*, 2006.

[6] A. Eriksson and K. Åström. Image registration using thin-plate splines. In *International Conference on Pattern Recognition*, Hong Kong, China, 2006.

[7] A. Ericsson J. Karlsson and K. Åström. Parameterisation invariant statistical shape models. In *Proc. International Conference on Pattern Recognition, Cambridge, UK*, 2004.

[8] H.H. Thodberg. Minimum description length shape and appearance models. In *Image Processing Medical Imaging, IPMI 2003*, 2003.