Image Partitioning based on Semidefinite Programming

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von
Dipl.-Math. Jens Keuchel
aus Osnabrück

Mannheim, 2004

Dekan:	Professor Dr. Jürgen Potthoff, Universität Mannheim
Referent:	Professor Dr. Christoph Schnörr, Universität Mannheim
Korreferent:	Professor Dr. Joachim M. Buhmann, ETH Zürich
Tag der mündlichen Prüfung:	13. September 2004

Abstract

Many tasks in computer vision lead to combinatorial optimization problems. Automatic image partitioning is one of the most important examples in this context: whether based on some prior knowledge or completely unsupervised, we wish to find coherent parts of the image. However, the inherent combinatorial complexity of such problems often prevents to find the global optimum in polynomial time.

For this reason, various approaches have been proposed to find good approximative solutions for image partitioning problems. As an important example, we will first consider different spectral relaxation techniques: based on straightforward eigenvector calculations, these methods compute suboptimal solutions in short time.

However, the main contribution of this thesis is to introduce a novel optimization technique for discrete image partitioning problems which is based on a semidefinite programming relaxation. In contrast to approximation methods employing annealing algorithms, this approach involves solving a convex optimization problem, which does not suffer from possible local minima. Using interior point techniques, the solution of the relaxation can be found in polynomial time, and without elaborate parameter tuning. High quality solutions to the original combinatorial problem are then obtained with a randomized rounding technique. The only potential drawback of the semidefinite relaxation approach is that the number of variables of the optimization problem is squared. Nevertheless, it can still be applied to problems with up to a few thousand variables, as is demonstrated for various computer vision tasks including unsupervised segmentation, perceptual grouping and image restoration.

Concerning problems of higher dimensionality, we study two different approaches to effectively reduce the number of variables. The first one is based on probabilistic sampling: by considering only a small random fraction of the pixels in the image, our semidefinite relaxation method can be applied in an efficient way while maintaining a reliable quality of the resulting segmentations. The second approach reduces the problem size by computing an over-segmentation of the image in a preprocessing step. After that, the image is partitioned based on the resulting "superpixels" instead of the original pixels. Since the real world does not consist of pixels, it can even be argued that this is the more natural image representation.

Initially, our semidefinite relaxation method is defined only for binary partitioning problems. To derive image segmentations into multiple parts, one possibility is to apply the binary approach in a hierarchical way. Besides this natural extension, we also discuss how multiclass partitioning problems can be solved in a direct way based on semidefinite relaxation techniques.

Zusammenfassung

Viele Bildverarbeitungsaufgaben lassen sich auf kombinatorische Optimierungsprobleme zurückführen. Eines der wichtigsten Beispiele in diesem Kontext ist die automatische Zerlegung von Bildern in kohärente Bestandteile, sei es unter Zuhilfenahme von Vorwissen oder völlig unüberwacht. Allerdings erlaubt es die hohe Komplexität derartiger Probleme häufig nicht, optimale Lösungen in polynomieller Zeit zu berechnen.

Aus diesem Grund wurden verschiedenartige Verfahren entwickelt, um gute Näherungslösungen für Bildpartitionierungsprobleme zu bestimmen. Als wichtiges Beispiel vergleichen wir zunächst mehrere spektrale Relaxations-Methoden, welche solche Approximationen mit Hilfe von speziellen Eigenvektor-Berechnungen ermitteln.

Der wesentliche Beitrag dieser Arbeit besteht jedoch darin, ein neuartiges Optimierungsverfahren zur diskreten Bildpartitionierung vorzustellen. Wir verwenden dazu einen Relaxationsansatz, der letztlich ein spezielles konvexes Optimierungsproblem liefert, welches mittels semidefiniter Programmierung gelöst werden kann. Im Gegensatz zu anderen Näherungsverfahren, die beispielsweise auf Annealing-Algorithmen beruhen, besteht somit keine Gefahr, in einem lokalen Minimum zu landen. Außerdem kann die Lösung eines semidefiniten Programms ohne aufwändige Parameteroptimierung mit Interior-Point-Methoden in polynomieller Zeit bestimmt werden. Qualitativ hochwerige diskrete Lösungen des Originalproblems lassen sich anschließend mit Hilfe einer probabilistischen Rundungstechnik ermitteln. Der einzige potenzielle Nachteil der semidefiniten Relaxation besteht darin, dass eine Quadrierung der Variablenanzahl notwendig ist. Nichtsdestotrotz lassen sich Probleme mit bis zu einigen tausend Variablen zufriedenstellend bearbeiten, wie wir anhand unterschiedlicher Bildverarbeitungsaufgaben aus der unüberwachten Segmentierung, perzeptuellen Gruppierung oder Bildrekonstruktion demonstrieren werden.

Für Problemstellungen höherer Dimension untersuchen wir zwei verschiedene Verfahren, welche die Variablenanzahl effektiv reduzieren. Zum einen handelt es sich dabei um einen Ansatz, der auf probabilistischem Sampling beruht: Durch zufällige Auswahl eines kleinen Prozentsatzes der Bildpixel erhalten wir ein Optimierungsproblem, auf das unser semidefinites Relaxationsverfahren effizient angewandt werden kann, und zwar unter Beibehaltung einer zufriedenstellenden Qualität der endgültigen Segmentierung. Das zweite Verfahren reduziert die Problemgröße, indem zunächst in einem Vorverarbeitungsschritt eine Übersegmentierung des Bildes berechnet wird. Anschließend werden anstelle der Pixel die resultierenden „Superpixel" als Grundlage zur Zerlegung des Bildes verwendet. Da die reale Welt nicht aus Pixeln besteht, erscheint dies sogar die natürlichere Bild-Repräsentation zu sein.

Unser semidefinites Relaxationsverfahren ist ursprünglich nur für binäre Problemstellungen definiert. Eine naheliegende Möglichkeit, um mehrteilige Zerlegungen zu erhalten, besteht in der hierarchischen Anwendung der binären Methode. Neben dieser Erweiterung untersuchen wir zudem, inwiefern Bildsegmentierungen in mehrere Teile auf direkte Weise mittels semidefiniter Relaxation bestimmt werden können.

Acknowledgements

First of all, I would like to thank Prof. C. Schnörr for supervising my dissertation and for giving me the opportunity to work in his group. He introduced me to the exciting field of computer vision, and his enthusiasm for this research field has always been very motivating. Moreover, I am grateful to Prof. J. Buhmann for serving as an external referee of this thesis.

Second, I would like to thank all the current and former members of the CVGPR group at the University of Mannheim for the inspiring environment and the wonderful atmosphere, which made research enjoyable around the clock. In particular, I want to thank M. Heiler and C. Schellewald who read preliminary versions of this thesis and gave many valuable comments for improvement.

Furthermore, the financial support of the Deutsche Forschungsgemeinschaft (DFG) is greatly acknowledged.

Last but not least, I wish to thank my wife Eva for her tireless moral support and her never-ending patience, which especially helped me during the last busy months of thesis writing.

In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.
(R.T. Rockafellar, SIAM Review 1993)

Contents

Chapter 1

Introduction

1.1 Motivation and Overview

In this first part of the introduction, we motivate our work from two different points of view: on the one hand, we propose a novel optimization approach in the field of computer vision that has many favorable properties (Section 1.1.1), while on the other hand, three different image partitioning tasks are presented to which this approach can conveniently be applied (Section 1.1.2).

1.1.1 Optimization Approach: Semidefinite Programming Relaxation

Tasks that are formulated as optimization problems appear in almost all fields of computer vision and pattern recognition. Concerning the design of appropriate problem formulations, one of the most important issues is to find an adequate compromise between a precise optimization criterion which correctly models the given problem, and the difficulty to compute the corresponding solution. On the one hand, an inappropriate optimization criterion will not describe the application in the desired way, no matter how easy it can be solved. On the other hand, a sophisticated mathematical model of the problem may be useless in practice unless a corresponding solution can be computed efficiently and conveniently (i.e. without elaborate parameter tuning or the need to provide exact a priori knowledge like good starting points, for example). Hence, an optimization approach is especially attractive if it represents an adequate tradeoff between such competing forces as accuracy, precision, speed and flexibility.

Concerning the mathematical modeling of optimization tasks in computer vision, or more specifically of image partitioning problems, we basically distinguish two different approaches: *continuous* methods are based on formulating the optimization objective as a functional defined on a continuous domain, which means that the arguments of the functional are continuous-valued. For example, such objective criteria emerge if an image (interpreted as a sampled instance of a continuous function) is segmented by approximating it with continuous-valued, piecewise smooth functions. This model allows using variational methods based on partial differential equations to define the corresponding optimization problems, which then are viable to be solved by employing

techniques from numerical analysis like e.g. gradient descent methods (cf. [186]).

In contrast to continuous methods, *discrete* approaches directly operate on optimization problems involving discrete-valued decision variables. The inherent combinatorial complexity of these problems in general prevents to find the solution in polynomial time, especially if the objective functional is globally defined. On the other hand, discrete approaches are more flexible than continuous methods, since they can be applied to numerous tasks in computer vision, including e.g. (un-)supervised segmentation, partitioning and perceptual grouping problems. For this reason, it is important to develop appropriate discrete optimization techniques that are able to conveniently handle such problems.

The main purpose of this work is to make a step into this direction by introducing a novel optimization technique to the field of computer vision, which is mainly based on the mathematically appealing and well-understood class of *convex programming* problems. To this end, we consider general minimization problems that comprehend a quadratic objective functional which is defined over binary decision variables and which may be subject to additional linear constraints. In contrast to related work, *no specific assumptions* are made with respect to the functional form apart from an unproblematic symmetry condition. As a consequence, our optimization approach can be utilized for a wide range of applications from computer vision, like unsupervised and supervised classification tasks, graph-based segmentation problems, or restoration based on first-order Markov random field estimates.

Such quadratic functionals allow us to deal with the combinatorial complexity of the optimization task by applying a general *semidefinite programming relaxation*. To be more precise, this approach targets to find a good approximative solution in two steps: first, the decision variables are lifted into a higher-dimensional space where the optimization problem can be tightly relaxed to a *convex* optimization problem. Specifically, the resulting semidefinite program comprises a linear objective function which is defined over a particular convex set (a so-called *cone*) in a matrix space, and a number of application-dependent linear constraints. After computing the global optimum of this relaxed problem, the decision variables are recovered in the second step by using a randomized rounding technique.

The fact that this approach involves solving a convex optimization problem results in various advantageous properties:

+ Due to its convexity, the *global optimum* of the *transformed* problem can be computed under mild conditions.

+ Interior-point algorithms are able to numerically determine an approximation of arbitrary precision to this global optimum in *polynomial time*.

+ In contrast to alternative optimization approaches, *no tuning parameters* are involved that critically influence the quality of the solution.

However, there is also an obvious drawback:

− The number of variables of the optimization problem is squared by the lifting step to a higher-dimensional space.

Figure 1.1: Two real world images illustrating the unsupervised segmentation problem. Based only on pairwise similarities between local measurements of image features like color or texture, we wish to partition these scenes into coherent groups.

Therefore, the semidefinite relaxation approach is limited to problems with at most a few thousand variables. Yet for many image partitioning and perceptual grouping tasks encountered in practice, this is already sufficient. Moreover, it is inevitable to increase the problem dimension if the intricate *combinatorial* constraints should be approximated closely by more comfortable *convex* sets that are eligible for applying convenient numerical optimization methods.

Concerning the computation of corresponding discrete solutions of the original optimization problem, we add the following favorable properties:

+ The randomized rounding procedure avoids the need to choose a *suitable threshold value*, as it is essential e.g. in spectral relaxation methods.

+ Combinatorial solutions of *high quality* are obtained based on the result of the semidefinite programming relaxation, i.e. the final solution is close to the unknown global optimum, which is intractable to compute.

From the optimization point of view, the absence of any specific assumptions about the objective criterion as well as the many "+"-properties listed above motivated our investigation of the semidefinite relaxation approach for computer vision problems.

1.1.2 Application to Image Partitioning Tasks: Segmentation, Grouping, Restoration

In the last section, we already presented the main motivation of our work: to devise a novel, mathematically well founded technique to approach a group of intricate combinatorial optimization problems. In this section, we motivate our work from the application point of view by illustrating three important tasks from early and mid-level computer vision which lead to different instances of the considered problem class. Besides indicating the significance of the present work in the context of computer vision, these tasks also serve as non-trivial specific examples that will be used throughout this thesis to demonstrate the performance of the semidefinite relaxation approach. Formal definitions of the corresponding optimization problems are given in Section 2.

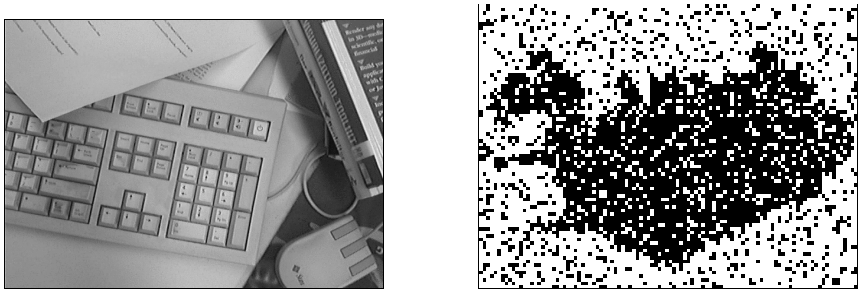


Figure 1.2: Perceptual grouping and restoration problems. The keyboard probably attracts most attention of the observer (left). How to compute this global figure-ground discrimination based on pairwise structural similarities between locally extracted image primitives? The right image shows a noisy binary image (map of Iceland) that should be restored.

Figure 1.1 shows two real world images taken from the VisTex database [187]. Partitioning such images in an *unsupervised* way (i.e. without incorporating specific a-priori knowledge) is a common goal of many low-level computer vision applications. Based on some locally computed features like color, texture, or motion, we wish to split the image into coherent groups of "similar" looking members. However, since no prototypes are given in advance to represent the different groups, defining the "correct" segmentation is not straightforward.

In this context, the representation of images by graph structures has recently attracted the interest of researchers [190, 168, 91, 62, 165, 60, 189]. More precisely, the resulting approaches partition an image by seeking minimal cuts in the underlying graph. In this work, we mainly study *binary* unsupervised image segmentation problems that are based on *constrained* minimal cuts. Specifically, we will show that the semidefinite relaxation method provides a tighter approximation of the corresponding combinatorial problem than techniques which are based on spectral graph theory. Concerning multiclass partitioning problems, a hierarchical framework is considered as well as direct extensions of the binary approach.

A different problem is depicted in the left image of Figure 1.2: for this section of an office table, probably most human observers focus on the keyboard as the most important object first. A typical task in mid-level computer vision is to model such global decisions of *perceptual grouping* by solving an optimization problem defined in terms of pairwise interactions between locally extracted image primitives [163, 157, 84, 142, 192, 173]. To this end, the optimization criterion considers a saliency measure with respect to decision variables indicating which primitives belong to the foreground or background, respectively. We will demonstrate that quadratic saliency measures which have been considered as difficult [192] due to their combinatorial complexity can conveniently be dealt with by using our semidefinite relaxation approach.

Finally, the right image in Figure 1.2 shows a binary image that has been degraded by noise. The *restoration* of such images has a long history in computer vision, in particular in the context of supervised segmentation tasks based on Markov random fields [64, 63, 193, 20]. We will show that our semidefinite

relaxation method allows us to model such labeling problems under less restrictive assumptions than those made by previous approaches [89, 24]. Moreover, a convenient extension to multiclass image reconstruction problems will be presented.

1.2 Related Work

Concerning image partitioning and segmentation problems, there exists a vast amount of literature considering various aspects of these computer vision tasks. To survey the entire field is beyond the scope of this work. In this section, we will therefore focus on some key publications that are directly related to our approach in the field of discrete optimization problems. More references will be given throughout this thesis when appropriate.

1.2.1 Optimization Approaches in Computer Vision

The examples presented in the previous section already indicate the importance of optimization problems that involve global objective criteria over discrete decision variables in the field of computer vision. Unfortunately, such problems are usually NP-hard, and only a few special cases can be solved to optimality in polynomial time [70, 5]. Accordingly, a lot of research has been done to develop optimization methods that are able to efficiently compute good approximate solutions.

An important class of optimization approaches that can deal with the combinatorial complexity involved in such problems is based on stochastic sampling and simulated annealing. Introduced in the context of computer vision in the seminal paper of Geman and Geman [64] on Bayesian image restoration, many applications have since been suggested in connection with Markov random fields [193, 20, 114]. Based on annealing schedules that are prescribed by theory, the corresponding algorithms can be guaranteed to find the global optimum of the combinatorial problem, yet at the cost of being impractically slow for real world applications. Nevertheless, the interest in these methods has still grown in recent years, especially in connection with interpretations of perception as Bayesian inference [106, 53], and with complex statistical texture models [207, 208].

In order to speed up the computations, various approaches to find suboptimal Markov random field estimates have been developed, including the ICM algorithm [16], the highest confidence first heuristic [30], the graduated nonconvexity strategy [17], flow-based local search heuristics [89, 24], or linear programming relaxations [105]. Other approximation methods are based on multi-scale approaches [77, 188, 148], biased importance sampling [180, 8], or deterministic versions of the annealing procedure for applications like perceptual grouping [84], data clustering [154, 86], or graph matching [67].

However, the accelerated computations take their toll: the above-mentioned methods can no longer guarantee to find the global optimum. In fact, this goal is illusive, considering the combinatorial complexity of the underlying optimization problems. Consequently, the following important question concerning the

performance of these approaches arises: is it possible to derive *bounds on the approximation quality* of the obtained solution in relation to the unknown global optimum, independent of the current problem instance? Although with respect to restricted problem classes, some of the mentioned approaches provide such performance bounds [17, 24, 105], none of the methods (apart from the original simulated annealing) seems to be immune *in general* against getting trapped in some poor local optimum, and hence meets this criterion.

A different problem concerns the *algorithmic properties* of this class of optimization approaches: apart from simple greedy strategies [16, 30], many methods imply some (sometimes hidden) parameters on which the computed local optimum critically depends. A typical example is given by the artificial temperature parameter in annealing approaches which governs the iterative annealing schedule. It is well known that without exact tuning mechanisms, the corresponding algorithms exhibit complex bifurcation phenomena [160], and may tend to oscillate in a parallel update mode [84, 143].

As we will see in Section 1.3, our semidefinite relaxation approach addresses both of the problems discussed above.

1.2.2 Semidefinite Programming

Optimization based on semidefinite programming still is a relatively novel field: first applications appeared around 1990 in control theory [21, 183] and combinatorial optimization [116, 3]. Since then, however, the interest has grown tremendously, motivated by the development of efficient algorithms [132, 199] as well as the discovery of new applications in diverse areas [196, 40]. As a special type of convex optimization [22], the corresponding duality theory is mathematically well-understood and established; for a detailed discussion of many aspects of semidefinite programming, we refer to [3, 183, 196, 179].

The method presented in this work is based on a semidefinite relaxation of combinatorial optimization problems. In fact, this technique to approximate intractable integer constraints is currently the main application of semidefinite programming. Initiated by the seminal work of Lovász and Schrijver [116] on 0-1 optimization, various applications in this context have been presented in the literature, like approximations for the max-cut problem [66], the independent set problem [71], graph coloring [95] and partitioning [153, 96], or the quadratic assignment problem [206, 26]. For 0-1 quadratic programming problems, Poljak et al. [145] present a general recipe to obtain semidefinite programming relaxations that is based on Lagrangian duality; a similar approach is pursued in this work. An overview of related recent developments is given in numerous surveys [65, 152, 78, 195, 115, 112].

While most of the afore-mentioned papers focus on deriving *tight bounds on the objective value* of combinatorial problems, Goemans and Williamson [66] were the first to present an algorithm that really computes a *suboptimal solution as an approximation* to the optimal combinatorial solution. Moreover, they were able to prove a performance guarantee for their randomized hyperplane technique: for the classical max-cut problem, the obtained suboptimal solutions cannot be worse (concerning the objective value) than 87.8% in relation to the

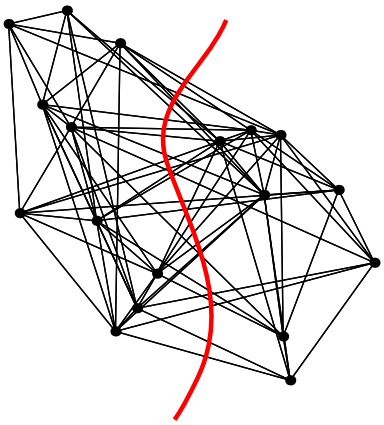


Figure 1.3: Representing image partitionings by graph cuts: the weights of all cut edges provide a measure for the (dis-)similarity of the resulting sets.

unknown global optimum. This fact has motivated us to adopt this approximation technique as the second step of our semidefinite relaxation approach to recover a combinatorial solution. Concerning possible modifications of the randomized approximation technique for different combinatorial problems, we refer to [55, 76].

1.2.3 Image Partitioning Tasks

As already indicated in Section 1.1.2, there is a wide range of applications in computer vision to which our semidefinite relaxation method can be applied. While it is beyond the scope of this thesis to give an in-depth discussion of all possible applications, we next focus on work related to the image partitioning tasks used here to illustrate our approach.

Unsupervised Image Segmentation and Clustering

Many recent approaches for unsupervised image segmentation are based on *graph partitioning* methods; see e.g. [168, 62, 173] and references therein. In this context, an image is naturally represented as a graph by considering locally extracted image elements (e.g. pixels) as vertices which are connected by weighted edges defined through pairwise (dis-)similarity values. The objective then consists in partitioning the vertices into disjoint sets according to some coherency criterion.

A popular criterion for such partitions is based on extremal cuts through the graph (cf. Figure 1.3). In computer vision, this idea was introduced by Wu and Leahy [198]. However, their min-cut criterion favors the separation of small sets, which brings up the important issue of appropriate cluster normalization to avoid too unbalanced partitionings. Initiated by the early work of Fiedler [51], a classical technique in this context employs spectral decomposition of the Laplacian matrix of the graph to efficiently compute balanced cuts. Later, this approach was also suggested as relaxation of a constrained graph

partitioning problem [129, 39]. Since then, this idea has been the subject of extensive research [74, 174, 73], and has found applications in many different fields [146, 38, 159]. In this work, we use the same basic idea of computing image partitionings based on constrained minimal cuts of the underlying graph, but pursue a different convex relaxation approach to find better suboptimal combinatorial solutions.

An alternative technique is proposed by Shi and Malik [168]: their "normalized cut" criterion prevents unbalanced cuts by directly normalizing the objective function. Again, an approximate solution of the resulting combinatorial problem is found by reverting to methods from spectral graph theory. For further developments and applications of the normalized cut, see e.g. [190, 125, 204, 54].

More segmentation approaches related to extremal graph cuts were recently presented in the literature, based on such different ideas as defining contour-based ratio regions [36], seeking minimal cost separations of each pixel from an artificial point outside the image [185], or normalizing the cut objective function by the length of the boundary between the segments [189]. In contrast to the afore-mentioned methods, these approaches lead to optimization problems that can no longer be solved by spectral techniques. Further interesting graph cut methods include calculating "typical average" cuts based on a stochastic sampling method [62, 165], or employing "Swendsen-Wang cuts" to obtain big candidate moves between partition stages [8]. However, as these methods consider probability distributions over the set of possible segmentations, they are not directly related to combinatorial optimization problems. Other segmentation methods based on graph partitioning techniques include e.g. recursive multiscale algorithms [164, 60] or the dominant set approach of Pavan and Pelillo [141, 140].

Supervised Image Partitioning and Labeling

Approaches to supervised image partitioning (or labeling) are often based on Markov random field models [193, 114] and therefore can be handled by applying the techniques mentioned in Section 1.2.1. Recent investigations of the corresponding optimization problems in the context of graph partitioning include [89, 24] and references therein. These authors derive a graph representation of the labeling problem by defining edges based on the neighborhood structure given by the Markov random field, and introducing additional connections to artificial terminal vertices that represent the given labels. It can be shown that for special pairwise interaction functions (e.g. semi-metrics), the optimal solution of the optimization problem can be found by graph cut techniques [107].

In contrast to this special result, our semidefinite relaxation approach is more general: we do not make any assumptions with respect to the pairwise interactions between the labels, which in fact can be negative or may not vanish for equal labels.

Perceptual Grouping

There is a vast literature on perceptual grouping in computer vision; see e.g. [122, 157, 90, 6] and references therein. In this work, we focus on the quadratic energy function proposed by Hérault and Horaud [84] for figure-ground discrimination. From an optimization point of view, the application of this saliency measure has been considered as difficult due to the computational cost resulting from its combinatorial complexity [192]. We will demonstrate that nevertheless, this grouping criterion can conveniently be optimized by applying our semidefinite relaxation approach.

1.3 Contribution and Organization

As has already been indicated, the main contribution of this work is to introduce a novel optimization technique to the field of computer vision, which is based on the mathematically appealing class of *semidefinite programming* problems. The corresponding convex relaxation approach can be applied to a wide range of combinatorial optimization problems, like unsupervised segmentation, perceptual grouping or restoration tasks. In this section, we give a more detailed overview by summarizing the most important contributions for each of the following chapters of this thesis separately.

In **Chapter 2**, we provide formal definitions of the optimization problems corresponding to the computer vision tasks considered in this work. As has already been illustrated in Section 1.1.2, these tasks comprise unsupervised segmentation based on graph partitioning, perceptual grouping (figure-ground discrimination), and image restoration as a special case of supervised classification. Focusing on *binary* versions of the corresponding combinatorial optimization problems, we show that all these tasks can conveniently be formulated as minimization problems which involve a quadratic objective functional over binary $(-1,+1)$-decision variables which may additionally be subject to a linear constraint. Apart from a symmetry condition, no specific assumptions are made with respect to the objective functional. Hence, our problem formulation is less restrictive than the ones used in related work [89, 24], and therefore potentially covers many other vision tasks.

Possible extensions to non-binary problems are discussed in later chapters, considering both a hierarchical approach for unsupervised segmentation (Section 5.1), and direct multiclass extensions for unsupervised and supervised image partitioning (Chapter 6).

In **Chapter 3**, we study different (un-)supervised segmentation methods which are related to our approach, and have already been applied successfully to various image partitioning problems. Especially, we closely investigate *spectral techniques* that have been proposed in the literature for graph-based image segmentation [168, 159, 158, 142]. All these methods have in common that they use certain extremal eigenvectors of the similarity matrix of the graph or of the corresponding Laplacian (either directly or normalized) to derive binary partitionings. In contrast to other comparing works [190, 168, 173, 162], we consider these techniques in a unifying framework as approximations to spe-

cial instances of a general "scaled cut" cost function. In this way, the relation between the average and normalized cut measures is clearly identified. Furthermore, this point of view enables the definition of an average association criterion which directly corresponds to the normalized association from [168]. The appropriate relaxation of this criterion leads to the calculation of eigenvectors of a centered similarity matrix, which also sheds some light on a related approach in the context of unsupervised learning [38]. Moreover, we provide different interpretations of spectral methods as approximations to other graph-related optimization criteria, and clarify the special meaning of the Fiedler vector.

Besides, this chapter also briefly recapitulates the *mean shift algorithm* [33] as an alternative unsupervised clustering technique that works directly in the (Euclidean) feature space of the extracted image elements. In particular, this method will later be used to reduce the size of image segmentation problems arising in practice.

Finally, we consider the *iterated conditional modes (ICM)* algorithm, which was proposed by Besag [16] in the context of supervised segmentation problems. Specifically, it is shown that this fast, local greedy technique approximates the global optimum of the energy functional used for image restoration in this work by calculating a local minimum. In contrast to that, our semidefinite relaxation approach yields a much tighter approximation of the underlying combinatorial problem, and hence most likely finds a better suboptimal solution.

Chapter 4 covers the main contribution of this thesis: the *semidefinite relaxation approach* can be applied to the general class of combinatorial optimization problems arising in computer vision that are presented in Chapter 2. Most details of this method have already been published in various conference proceedings [161, 100, 102] and in a journal article [103]. Here, we extend the corresponding results by considering a slightly more general balancing constraint, that also allows specifying the proportion of each part for unsupervised segmentation tasks.

The emerging convex optimization problems belong to the special class of *semidefinite programs*. For this reason, we first provide a concise introduction to this topic (Section 4.1) by discussing the main aspects of the corresponding duality theory, the geometry of the cone of positive semidefinite matrices, and related algorithms. On this mathematical basis, our semidefinite relaxation approach is explained in Section 4.2. Specifically, we perform Lagrangian relaxation [113] to derive a semidefinite program for the general class of combinatorial optimization problems studied here. By considering geometry and feasibility issues, it is shown that under mild assumptions, a globally optimal solution exists for this *convex* optimization problem, which can be computed in polynomial time with clear numerical algorithms. Abstracting from the computational process, due to the lack of local optima we can simply think of a mapping taking the data to this solution. Thus, evidently, no hidden parameters are involved.

Moreover, we discuss the randomized hyperplane technique [66] as approximation procedure to calculate suboptimal solutions of the original combinatorial problem. In this context, we also present bounds with respect to the quality of these solutions that under certain conditions can be derived from bounds given

in the literature for other optimization problems [66, 131, 200]. As is revealed by statistical results for ground-truth experiments, these theoretical bounds are not tight in practice, where a much better performance can be measured. Yet note that for most alternative optimization methods, similar results and a corresponding route of research seem to be missing.

In comparison to certain spectral techniques for unsupervised partitioning tasks, we show that our semidefinite relaxation approach is superior from the approximation point of view, as it yields a tighter relaxation of the underlying combinatorial problem (Section 4.3). This theoretical result is also confirmed in practical applications, where spectral relaxation may produce unsatisfactory solutions when no appropriate threshold value is chosen or can be found. In contrast to this, our approach does not depend on tuning such a parameter.

In this context, it has also been criticized that methods based on spectral graph theory are not able to partition highly skewed data distributions or non-compact clusters [62]. We will demonstrate that a straightforward remedy in this case is to base the similarity measure on a suitable path metric, as is also advocated in related work (e.g. [52]). However, the derivation of new similarity measures is not the topic of this work — although we are aware that successful similarity-based clustering always depends on the choice of a suitable similarity measure. Rather, the focus of this thesis is on the application of a convenient optimization technique for computer vision problems.

Finally, a thorough experimental investigation on real scenes demonstrates the versatility of the semidefinite relaxation approach in practice (Section 4.4). For a broad range of difficult combinatorial problems obtained from different image partitioning tasks, meaningful solutions can be obtained in a convenient way. Instead of having to worry about technical details of the optimization procedure, the user can focus on choosing appropriate constraints according to the desired application. Although the computational effort increases, the results reveal that for problems with up to 10,000 variables, the solution can still be found efficiently.

Chapter 5 is devoted to methods that enable the application of our semidefinite relaxation approach to larger problem instances as they arise for real world unsupervised partitioning tasks. In general, graph partitioning techniques become computationally demanding (or even intractable) with increasing size of the images, since for example the corresponding similarity matrices do no longer fit into memory completely. A common idea in this case is to revert to sparse graph representations by connecting pixels only within a certain neighborhood [168, 62, 204]. However, this is of no avail for our approach, as other matrices involved in the solution process still are usually dense.

We therefore propose two other methods which immensely reduce the problem size by *preprocessing* the image appropriately. The basic idea of the first method resembles the perceptual grouping task: abandoning pixels as the basic image elements, we instead use small image patches of coherent structure to define the corresponding graph representation. It can be argued that this is even a more natural image representation, since the pixels are merely the result of the digital image discretization process. To obtain such an *over-segmentation*, we apply the mean shift technique [33] at a small spatial scale. Based on other

preprocessing methods, this "superpixel"-idea has also been advocated in recent related work [198, 118, 189, 8, 151].

The second method (which first was presented in a workshop paper [101]) is based on a completely different idea: following mathematical approximation techniques, we *probabilistically sample* the entries of the similarity matrix to obtain a good low-rank approximation to the complete matrix. In connection to image segmentation, this amounts in randomly picking a small number of pixels to obtain an optimization problem of smaller scale, the solution of which can afterwards be generalized well to a solution of the original, large problem. Similar ideas have already been applied successfully for spectral graph cut techniques [54] and in a different clustering context [48].

To obtain more meaningful results, the binary semidefinite relaxation approach is extended in this chapter: by computing partitions consecutively in a *hierarchical way*, we derive image segmentations into multiple parts (cf. [98, 198, 118]).

In **Chapter 6**, we analyze how *multiclass image partitioning* problems can be solved by semidefinite relaxation approaches in a direct way — in contrast to a hierarchical application of the binary method. To this end, we assume that the number of parts the image should be decomposed into is always known in advance; we do not investigate how this number can be found automatically. In this case, the binary formulations of both the unsupervised segmentation and the restoration problem can be naturally extended to appropriate combinatorial multiclass optimization problems, which however do no longer fit into a common framework. Nevertheless, we derive suitable convex relaxations for both problems, which are mainly based on recent investigations concerning semidefinite relaxations of the quadratic assignment problem [206] and the multiclass partitioning problem [55, 96, 197], respectively.

Since the corresponding research is still in progress, the results presented in this chapter should be considered as preliminary. However, first experiments indicate that although the computational burden increases even faster than for binary problems, semidefinite relaxation is promising for non-binary problems, too.

Finally, **Chapter 7** briefly summarizes the main results of the present work, and prospects directions for future research. In the **Appendix A**, we state some important mathematical facts that arise in connection with symmetric and positive semidefinite matrices which are needed throughout this thesis, and note where the corresponding proofs can be found.

1.4 Notation

The following table lists the basic notation that is used throughout this thesis. In this context, we remark that with slight abuse of mathematical accuracy, we always use the notation of min/max for the considered optimization problems, although the optimum sometimes may not be attained (which strictly would require the notation of inf/sup). In this way, we would like to emphasize that we are interested in finding an *optimal solution* and not just the optimal value of the objective function.

Numbers and Vectors

$\mathbb{R}$	real numbers
$\mathbb{R}_0^+$	positive real numbers, including 0
$\lvert a\rvert$	absolute value of $a \in \mathbb{R}$
$\operatorname{sgn} a$	sign of $a \in \mathbb{R}$
$\binom{n}{k}$	binomial coefficient: $\binom{n}{k} = \frac{n!}{(n-k)!k!}$
$\mathbb{R}^n$	space of n-dimensional real vectors
$x^\top$	transpose of x
$\lVert x\rVert$	Euclidean norm of the vector x: $\lVert x\rVert^2 = x^\top x$
e	vector of all ones: $e = (1, \ldots, 1)^\top$
e^k	vector of all ones of dimension k: $e^k \in \mathbb{R}^k$
e_i	i-th unit vector: $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$

Matrices

$\mathbb{R}^{n\times k}$	space of $n \times k$-dimensional matrices
$\mathcal{S}^n$	space of symmetric $n \times n$ matrices
$\mathcal{S}_+^n$	space of symmetric, positive semidefinite $n \times n$ matrices
$A \succeq 0$	matrix A is positive semidefinite
I	identity matrix
E	matrix of all ones $E = ee^\top$
$E_k, E_{n\times k}$	matrix of all ones of dimension $k \times k$ or $n \times k$, respectively
$\lambda_i(A)$	i-th eigenvalue of A, with $\lambda_1(A) \leq \cdots \leq \lambda_n(A)$
$\operatorname{rank}(A)$	rank of the matrix A
$\operatorname{Tr}(A)$	trace of the matrix A: $\operatorname{Tr}(A) = \sum_i A_{ii}$
$\lVert A\rVert_{\mathrm{F}}$	Frobenius norm of the matrix A: $\lVert A\rVert_{\mathrm{F}} = \sqrt{\sum_{i,j} A_{ij}^2}$
$A \bullet B$	standard matrix inner product: $A \bullet B = \operatorname{Tr}(A^\top B)$
$A \circ B$	Hadamard product of A and B: $(A \circ B)_{ij} = A_{ij}B_{ij} \;\; \forall\, i, j$
$A \otimes B$	Kronecker product of A and B
$\operatorname{vec}(A)$	vector containing the stacked columns of A
$\operatorname{Diag}(x)$	diagonal matrix with vector x on its main diagonal
$\operatorname{diag}(A)$	diagonal of the matrix A as a column vector
$\ker(A)$	null space (kernel) of A: $\ker(A) = \{x \in \mathbb{R}^n \mid Ax = 0\}$

Graphs

$G(V, E)$	undirected graph with vertex set $V = \{1, \ldots, n\}$ and edges $E \subseteq V \times V$
W	weighted symmetric adjacency matrix with entries w_{ij}, $i, j \in V$
L	Laplacian matrix of the graph: $L = \operatorname{Diag}(We) - W$
$\overline{S}$	complement of the vertex subset S: $\overline{S} = V \setminus S$
$\omega(S)$	general sum of the vertex weights ω_i for $i \in S$: $\omega(S) = \sum_{i\in S} \omega_i$
$\lvert S\rvert$	number of vertices contained in the subset S
$d(S)$	degree of the subset S: $d(S) = \sum_{i\in S, j\in V} w_{ij}$
$\operatorname{cut}(S, \overline{S})$	weight of the cut separating S and $\overline{S}$: $\operatorname{cut}(S, \overline{S}) = \sum_{i\in S, j\in \overline{S}} w_{ij}$
$\operatorname{assoc}(S)$	inner association of the subset S: $\operatorname{assoc}(S) = \sum_{i,j\in S} w_{ij}$

Miscellaneous

$\mathrm{vol}(T)$	volume of the hypersphere T
$\Pr[x]$	probability of x
$\Pr(x \mid g)$	conditional probability of x, given g
$E[x]$	expectation of x
δi	set of neighbors of a pixel i

Chapter 2

Binary Optimization Problems in Computer Vision

In this thesis, we mainly consider combinatorial optimization problems of the following general form:

$$\begin{aligned} \min_{x} \quad & x^\top Q x + 2d^\top x + const \\ \text{s.t.} \quad & x \in \{-1,+1\}^n \\ & c^\top x = \beta \,, \\ \text{with} \quad & Q \in \mathcal{S}^n,\ c,d \in \mathbb{R}^n,\ \beta \in \mathbb{R} \,, \end{aligned} \tag{2.1}$$

where $\mathcal{S}^n$ denotes the space of symmetric $n \times n$-matrices. Thus, the objective is to minimize a quadratic functional over binary decision variables subject to a linear constraint.

In computer vision, global optimization problems of the form (2.1) arise in various contexts, including e.g. grouping tasks, Markov random field estimates or graph-optimization problems. In the following sections, we will give formal definitions of three different partitioning and segmentation problems which can be cast in this form, and briefly present methods of how they are handled in the literature.

Note that apart from symmetry, no further constraints are imposed on the matrix Q in (2.1). Hence, the objective function does not need to be convex in general. This property along with the integer constraint $x_i \in \{-1,+1\},\ i = 1,\ldots,n$ makes the minimization problem intrinsically difficult (which usually means NP-hard). In Chapter 4, we will present a semidefinite relaxation approach for problems of this type, which computes suboptimal solutions of high quality in polynomial time by closely approximating the binary problem (2.1).

2.1 Unsupervised Partitioning via Graph Cuts

Segmenting an image into its main parts in an unsupervised way is an important task in computer vision. To this end, the image is often represented by feature vectors comprising e.g. position, brightness, color, or texture information for each basic image element (usually the pixels). A segmentation then corresponds

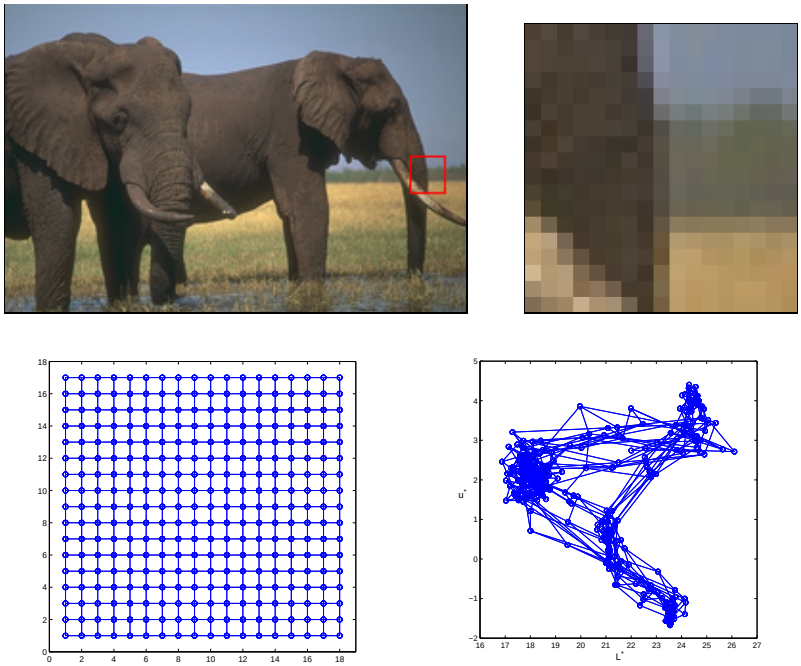


Figure 2.1: A typical image to be segmented. For the small patch marked in the image (top), a corresponding unweighted graph is constructed by connecting only neighboring pixels by an edge (bottom left). Representing the color differences of the pixels (in the perceptually uniform L*u*v* space) as distances in $\mathbb{R}^2$ results in the weighted graph depicted on the bottom right: a certain cluster structure is clearly visible.

to finding groups of similar image elements. However, since no prototypes for the different groups are given in advance, it is difficult to define the "correct" partitioning. Therefore, many partitioning methods directly try to cluster the feature vectors in the corresponding Euclidean space. One example of such a clustering method is the *mean shift algorithm* [33], which we will briefly present in Section 3.2.

An alternative to define "good" segmentations is based on a different representation of the image which uses pairwise (dis-)similarity relations between the image elements. Although such relations of course can be obtained from feature vectors by computing (Euclidean) distances between them, signal variability may be captured in a better way by calculating (dis-)similarities between image elements directly (i.e. without using feature vectors) [181]. In either case, the (dis-)similarity relations lead to the following graph representation of an image: consider the locally extracted image elements as vertices V of the weighted graph $G(V, E)$, and connect two vertices $i, j \in V$ by an edge $(i, j) \in E$ weighted with the corresponding (dis-)similarity value $w_{ij} \in \mathbb{R}_0^+$. If no edge is present between two vertices, this is expressed by an edge-weight of $w_{ij} = 0$ (or $w_{ij} = \infty$ for dissimilarities). Figure 2.1 shows an example of such a graph for a small patch of an image.

The binary unsupervised segmentation problem is now equivalent to partitioning the set V into two disjoint groups S and $\overline{S} = V \setminus S$. To measure

the quality of a segmentation, a common idea is to define a cost function $f(S)$ which depends on the weight of the corresponding cut in the graph:

$$\mathrm{cut}(S, \overline{S}) = \sum_{i \in S, j \in \overline{S}} w_{ij} \; . \tag{2.2}$$

If the weights w_{ij} encode a similarity measure between image elements (i.e. small values for w_{ij} correspond to low similarity), a good segmentation into coherent groups leads to low cut-values.

Using $f(S) = \mathrm{cut}(S, \overline{S})$ directly as the cost function as was suggested in [198] yields a minimization problem which can be solved in polynomial time. However, this method has the disadvantage that it favors unbalanced segmentations: since separating a single vertex from the rest of the graph cuts the fewest edges, f usually grows with the number of vertices contained in either S or $\overline{S}$ (cf. Figure 2.1). To avoid this problem, several more suitable optimization criteria have been suggested in the literature. In this context, a popular approach to obtain balanced partitions is to scale the cost function with factors related to the areas of the segments [74, 168, 45]:

$$f(S) = \frac{\mathrm{cut}(S, \overline{S})}{\omega(S)} + \frac{\mathrm{cut}(S, \overline{S})}{\omega(\overline{S})} \; , \tag{2.3}$$

where $\omega(S)$ denotes a (positive) weight of the subset S (e.g. the number of vertices $|S|$). Minimizing (2.3) then leads to segments of similar (high) weight which are separated by a small cut. Cost functions of this form yield NP-hard minimization problems, but they can be solved approximately with spectral relaxation methods [74, 4, 168, 159]. In Section 3.1, we will investigate several of these approaches in more detail.

An alternative technique to avoid unbalanced partitions originates from the classical equipartition approach from spectral graph theory (cf. [129]): instead of scaling the cut-value, an additional linear constraint is used to find favorable partitions. To formulate the corresponding optimization problem mathematically, let $n = |V|$ and represent a partitioning into S and $\overline{S}$ by an indicator vector $x \in \{-1, +1\}^n$. Denoting by D the diagonal degree matrix with $D_{ii} = \sum_{j \in V} w_{ij}$ and using the *Laplacian matrix* $L = D - W$ of the graph, the weight of a cut is given by

$$\mathrm{cut}(S, \overline{S}) = \frac{1}{8} \sum_{i,j \in V} w_{ij} (x_i - x_j)^2 = \frac{1}{4} x^\top L x \; . \tag{2.4}$$

The equipartition problem now consists in determining a cut with minimal weight subject to the constraint that the number of vertices in both groups is equal:

$$\begin{aligned} \min_{x \in \{-1,+1\}^n} \quad & x^\top L x \\ \text{s.t.} \quad & e^\top x = 0 \; . \end{aligned} \tag{2.5}$$

Here, $e = (1, \ldots, 1)^\top$ denotes the vector of all ones.

A natural relaxation of the NP-hard problem (2.5) is to drop the integer constraint on x. As e is the eigenvector corresponding to the smallest eigenvalue 0 of the Laplacian matrix L, this results in computing the second smallest eigenvector of L (the so-called *Fiedler vector*) and thus in a method which is of the same type as the spectral techniques used to approximate the scaled cost function (2.3) — see Section 3.1.3.

The main topic of this work, however, is to present an alternative approach to spectral relaxation. It is motivated by some questions that arise naturally in this context: for image segmentation, the equipartition constraint in (2.5) may be too strict; which other constraints are useful replacements? How can the integer constraint with respect to x_i, $i = 1, \ldots, n$, be better taken into account to derive an appropriate relaxation of the combinatorial optimization problem (as opposed to just dropping it and thresholding x afterwards like the spectral relaxation methods do)? To investigate these topics, we define the following generalized (graph-bisection) criterion for unsupervised image partitioning:

$$\begin{aligned} \min_{x \in \{-1,+1\}^n} \quad & x^\top L x \\ \text{s.t.} \quad & c^\top x = \beta \, . \end{aligned} \tag{2.6}$$

The vector $c \in \mathbb{R}^n$ and the value $\beta \in \mathbb{R}$ in the linear constraint can be set application-dependent and define our notion of a "balanced cut". Note that this generalization of the partitioning criterion renders spectral relaxation approaches inappropriate, because those require special fixed balancing constraints. In Chapter 4 we will therefore focus on a more advanced method to relax and solve the problem (2.6) which not only is able to handle the generalized linear constraint, but also takes the integer constraint on x into account in a more adequate way.

If the linear constraint variables c and β are chosen inappropriately, the combinatorial problem (2.6) may have no feasible solution (a simple example is the case $c = e$, $\beta = 0$ with n being an odd number). In this case, an additional variable $x_0 \in \mathbb{R}$ can be incorporated to close the gap in the balancing constraint: $c^\top x + x_0 = \beta$. In order to find a solution which is as feasible as possible to (2.6), the variable x_0 is minimized by including it into the objective function. Hence we arrive at the following problem formulation:

$$\begin{aligned} \min_{x_0 \in \mathbb{R},\, x \in \{-1,+1\}^n} \quad & \begin{pmatrix} x_0 \\ x \end{pmatrix}^\top \begin{pmatrix} \alpha & 0 \\ 0 & L \end{pmatrix} \begin{pmatrix} x_0 \\ x \end{pmatrix} \\ \text{s.t.} \quad & \begin{pmatrix} 1 \\ c \end{pmatrix}^\top \begin{pmatrix} x_0 \\ x \end{pmatrix} = \beta \, , \end{aligned} \tag{2.7}$$

where α is a sufficiently large number. However, as we will show in Section 4.2.2, an infeasible instance of problem (2.6) not necessarily prevents that the relaxation works: under specific conditions for c and β, which are usually satisfied in practice, the relaxation will be feasible even when the corresponding instance of (2.6) is not. In this case, meaningful combinatorial solutions which approximately satisfy the balancing constraint can be obtained from the relaxation without using extension (2.7). For this reason, we will not consider it any further here.

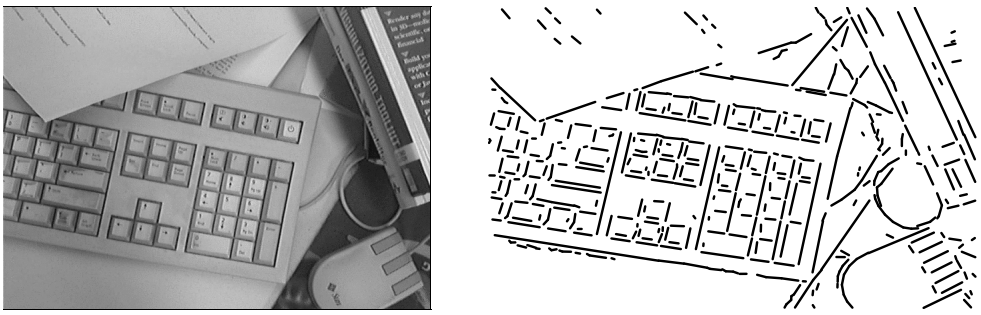


Figure 2.2: Perceptual grouping. Using a line finder, the edges shown in the right image are extracted from the left image. The keyboard as the object which probably attracts most attention from the observer consists of very regular configurations of image features, namely mostly orthogonal and parallel edges.

Finally, note that the binary unsupervised partitioning approach can easily be extended to segment an image into multiple parts by applying it in a hierarchical way (e.g. [198, 118]): based on certain decision rules, we continue to split segments into two parts until a stopping criterion (e.g. a pre-specified number of segments) is met. We will closer investigate the combination of our semidefinite relaxation method with this hierarchical approach to multiway partitioning in Section 5.1. Moreover, an extension to direct multiclass segmentation will be introduced in Section 6.2.

2.2 Perceptual Grouping

Another central problem connected to image partitioning is based on figure-ground discrimination: taking locally extracted low-level image features like edges or corners as input, the objective is to separate them into shape and noise [84], or into foreground and background. Phrased differently, this means to group together features which for human perception are likely to be similar (as they belong to the same object), while eliminating less important elements. This idea is motivated by the so-called "principle of good form": as ordered geometric arrangements of image features are unlikely to arise accidentally, emerging objects in an image are closely related to a mostly regular, stable, balanced configuration of the features [157]. Figure 2.2 shows an example: based on the edges extracted from the image, we wish to separate regular configurations (which in this case most probably belong to the keyboard) from the background clutter.

One possibility to find structure in the image is to apply the well-known *Hough transform*, which maps the image elements into a parameter space associated with a pre-defined curve type [147]. Standard clustering techniques can then be used to find groups of elements in accordance to this curve type. However, this method has one important drawback: the perceptual grouping problem is usually posed at an early stage of the visual system, where it is required to decide whether some features may form a shape or not, without

special knowledge about the curves the shape is generated of.

To model perceptual grouping mechanisms on an early level of the visual process, a basic definition is necessary of what is meant by shape and noise, respectively. To this end, interactions between pairs of image elements are usually measured based on Gestalt associations like cocircularity, continuity, proximity, or parallelism/perpendicularity [84, 158, 142]. These interactions can be combined to yield a pairwise similarity measure $w_{ij} \geq 0$ for image primitives i and j. Based on this measure, numerous energy minimization criteria have been proposed in the literature to find salient configurations [163, 138, 122, 84, 158, 142, 6, 192]. These methods differ in several aspects, e.g. by assuming prior distributions of the image elements or by using local or global optimization techniques.

In this work, we investigate the figure-ground discrimination problem from a combinatorial optimization perspective based on an energy function proposed by Hérault and Horaud [84]. For perceptually grouping n primitives, they consider the problem of minimizing the following functional in terms of binary labels $p \in \{0,1\}^n$:

$$E_{HH}(p) := E_{\text{saliency}}(p) + \lambda E_{\text{constraint}}(p) \;, \tag{2.8}$$

where a label $p_i = 1$ indicates a figure element i, while a label $p_i = 0$ corresponds to background or noise. The interaction energy in (2.8) is defined as

$$E_{\text{saliency}}(p) = -\sum_{i,j} w_{ij} p_i p_j \;.$$

This quadratic function measures the mutual reinforcement between the image elements labeled as figure, and is obviously minimized when all labels are equal to 1. This trivial solution is avoided by adding the constraint term

$$E_{\text{constraint}}(p) = \left(\sum_i p_i \right)^2 \;,$$

which encourages small numbers of figure elements. In this way, noise is eliminated because primitives are penalized if they do not receive much "feedback" from other primitives, which indicates that they do not belong to some salient group. The parameter $\lambda \in \mathbb{R}_0^+$ in (2.8) can be interpreted as serving the purpose to adjust the signal-to-noise ratio [84].

In order to find good minimizers of (2.8), Hérault and Horaud investigated various annealing approaches [84]. In contrast to local minimization techniques [163, 138], which strongly depend on good initializations, these optimization methods are able to cope with the large number of local minima of the energy function and thus to find (a good approximation to) the global optimum [64]. However, the corresponding annealing schedules usually are impractically slow for real world applications, and require exact tuning of the artificial temperature parameter. Accordingly, in a recent comparison [192], the combinatorial complexity and the resulting computational cost have been considered as decisive disadvantages of using (2.8) as a saliency measure.

In Chapter 4, we will demonstrate that a good minimizer for (2.8) can yet be conveniently computed with our semidefinite relaxation approach. To this end, we derive a formulation of the energy (2.8) in terms of ± 1-variables by using the transformation $x = 2p - e$, which finally leads to the following minimization problem (up to constant terms):

$$\min_{x \in \{-1,+1\}^n} \frac{1}{4} x^\top (\lambda E - W) x + \frac{1}{2} e^\top (\lambda E - W) x \;, \tag{2.9}$$

with $E = ee^\top$ denoting the matrix of all ones. This formulation makes more explicit the role of the signal-to-noise ratio parameter λ which acts as a threshold value in a twofold way: two primitives i and j reinforce each other if their similarity value w_{ij} is larger than λ (first term), whereas a single primitive i is (additionally) favored if its *average* similarity $\frac{(We)_i}{n}$ is larger than λ (second term). The combination of both terms results in a meaningful global measure of "saliency" based on pairwise comparisons of locally computed primitives.

A closer look at (2.9) reveals a connection of this perceptual grouping approach to the graph cut problems presented in the last section: consider the graph with the primitives as vertices, and define edge-weights $\tilde{w}_{ij} := w_{ij} - \lambda$ (which leads to both positive and negative edge-weights). Minimizing the cut in this graph then is equivalent to minimizing the first term in (2.9), since $\frac{1}{4} x^\top (\lambda E - W) x = -\frac{1}{4} x^\top \tilde{W} x = \frac{1}{4} x^\top \tilde{L} x - const.$ In contrast to the partitioning problems presented in the last section, the trivial solution is now prevented by adding a weighted linear term $c^\top x$ with $c = \frac{1}{2}(\lambda E - W)e$ to the objective function instead of scaling by oppositional terms as in (2.3) or using an additional linear constraint as in (2.6). The additional minimization of the term $c^\top x$ now balances the number of foreground elements against the amount of background: as already stated above, negative values $c_i = \frac{1}{2}(n\lambda - (We)_i) = \frac{1}{2}\sum_j (\lambda - w_{ij})$ favor the corresponding primitive to belong to the foreground, and vice versa. Through this perspective the interpretation of λ as a signal-to-noise ratio also becomes more obvious: larger values of λ correspond to higher percentages of noise, and thus allow fewer foreground primitives.

In fact, a direct interpretation of (2.9) as a graph cut measure is possible if besides the vertices corresponding to the primitives one additional terminal vertex is defined which corresponds to the label $+1$. This terminal is connected to each vertex i with an edge weight of c_i. Finding a minimal cut in this graph then is equivalent to the problem (2.9). In [70, 24] a similar graph representation with two terminals is used to find exact minima for special energy functions.[1] However, as (2.9) does not conform with the assumptions made there (since the corresponding graph contains negative edge-weights), it cannot be solved by applying those techniques.

[1] Their graph representation contains two terminal vertices t_1 and t_2 corresponding to the labels $+1$ and -1, respectively. A vertex i is connected to t_1 if $c_i > 0$ and to t_2 if $c_i \leq 0$, in either case with an edge weight of $|c_i|$ [70]. Obviously, this leads to the same minimum cut problem as the one-terminal representation.

Figure 2.3: Binary Restoration Problem. A black and white map of Iceland has been degraded by adding binary salt and pepper noise.

2.3 Restoration and Supervised Classification

A fundamental issue in computer vision is *image restoration* (reconstruction), which is a special case of the broader problem of *supervised classification* (image labeling). In contrast to the unsupervised partitioning problem presented in Section 2.1, in this case some prototypical data is given which represents the groups to which the image elements should be assigned. This prototypical data is either known in advance (leading to a restoration problem, i.e. the image features are noisy measurements of a number of known prototypes), or can be estimated from training data. In the latter case, it is a common approach to assume a certain probability distribution of the image features for each group, and to calculate the corresponding parameters (e.g. mean value and covariance) from the training data to find a prototypical representation of each group.

As an illustrating example consider the restoration problem given in Figure 2.3: the original binary image (a map of Iceland) has been degraded by noise. Based on the observed values, we want to determine the true intensity for each pixel. To this end, a compromise between two competing forces should be found: on the one hand, we seek for classifications that best conform to the observed intensities, while on the other hand — assuming that natural images are mostly smooth, except for occasional region boundaries — spatially neighboring pixels should receive similar labels.

In order to find a labeling which captures this trade-off, we seek to minimize a global energy function which involves pairwise relationships among the objects. This kind of problem has a long history in the literature, in particular as it arises naturally from the well-studied theory of *Markov random fields* [64, 63, 193, 20], a statistical framework that builds the core of many image processing applications [16, 49, 114, 104].

Using the integer variables x_i to indicate the label of each image element i, this amounts in minimizing an energy functional of the following general form [105]:

$$E(x) = \sum_i C_i(x_i) + \sum_{\langle i,j \rangle} P_{ij} D(x_i, x_j) \;, \tag{2.10}$$

where the second term sums over all pairwise adjacent image elements. The energy (2.10) comprises two terms familiar from many regularization approaches [13]: a data-fitting term and a smoothness term modeling spatial context. In

more detail, the data-fitting term measures the assignment costs $C_i(x_i)$ of labeling elements i as x_i, while in the smoothness term, the separation costs consist of two factors for each related pair i, j of objects: the weight P_{ij} indicating the strength of the relation and the distance $D(x_i, x_j)$ of the two labels x_i and x_j.

Due to the integer constraint on x the optimization problem obtained from (2.10) is much more difficult than standard regularization problems. In fact, apart from a few special cases, it is in general NP-hard [24]. Different methods have been proposed to find good minimizers of (special instances of) the energy (2.10) efficiently, like the ICM-algorithm [16], the graduated non-convexity approach [17], flow-based local search heuristics [89, 24], or linear programming relaxations [105]. Of these approaches, we will present the ICM-algorithm as an exemplary local greedy technique in more detail in Section 3.3.

In this thesis, we will show how semidefinite relaxations can be used to approximate the minimal solution of the energy (2.10). To begin with, we first focus on a *binary* version that can be cast in the general form (2.1); the multiclass case will be considered in Section 6.1. To this end, assume that feature vectors g_i (comprising e.g. gray-values, color-values, or texture measures) were locally computed for each pixel i within the image plane. Furthermore, we suppose that g_i is known to originate from either of two given prototypical vectors u_1 or u_2. In practice, of course, g_i contains perturbed values due to measurement errors and noise. Using the integer variable $x \in \{-1, +1\}^n$ as class indicator in connection with defining assignment costs $C_i(x_i) = \frac{1}{4}\|x_i(u_2 - u_1) + u_2 + u_1 - 2g_i\|^2$ and separation costs $P_{ij}D(x_i, x_j) = \lambda\frac{1}{2}(x_i - x_j)^2$, we obtain the following energy functional:[2]

$$E_R(x) = \frac{1}{4}\sum_i \|x_i(u_2 - u_1) + u_2 + u_1 - 2g_i\|^2 + \frac{\lambda}{2}\sum_{\langle i,j\rangle}(x_i - x_j)^2 \,. \qquad (2.11)$$

In this case, the pairwise adjacent image elements correspond to horizontally and vertically neighboring pixels on the regular image grid. Like in (2.8), the parameter λ balances the trade-off between data similarity and smoothness of the result.

Up to constant terms, (2.11) leads to the following optimization problem:

$$\min_{x\in\{-1,+1\}^n} \; \frac{1}{4}x^\top L x + \frac{1}{2}d^\top x \,, \qquad (2.12)$$

with $d_i = (u_2 - u_1)^\top(u_2 + u_1 - 2g_i)$, and matrix entries $L_{ij} = -2\lambda$ for adjacent pixels i, j and $L_{ij} = 0$ otherwise. Note that in contrast to the problems introduced in the previous sections, in this case the problem matrix L is very sparse, which is advantageous from the computational point of view.

We are well aware that the special instance (2.12) of the restoration problem can be solved to optimality using the methods presented in [70, 89, 23], as the separation costs yield a metric Potts interaction penalty [23]. In fact, (2.12)

[2]Note that the assignment costs $C_i(x_i)$ are naturally obtained from the distance measure $\|z_i^\top(u_1, u_2) - g_i\|^2$, with $z_i \in \left\{\binom{1}{0}, \binom{0}{1}\right\}$ indicating the class membership of i, by substituting $z_{i1} = \frac{1}{2}(1 - x_i)$ and $z_{i2} = \frac{1}{2}(1 + x_i)$.

can be interpreted as a graph cut problem in the same manner as the perceptual grouping problem (cf. end of the last section): the corresponding graph for which a minimum cut is sought has edges weighted with 2λ for adjacent pixels and edges with the weight d_i from each pixel vertex to the terminal vertex. However, depending on the application considered, it might be useful to modify the terms in (2.11) to model properties of the imaging device (data-fitting term) or to take into consideration *a priori* knowledge about spatial regularities (smoothness term; see, e.g., [20, 193]). These modifications would lead to other entries for L and d, which could violate the assumptions on the energy functional made in [70, 89, 23], but would not affect the applicability of our semidefinite relaxation approach.

Chapter 3

Established Segmentation Methods

In this chapter we summarize (un-)supervised segmentation methods that have been applied successfully to various image partitioning tasks, and that will later be used for comparison with our semidefinite relaxation approach. We start with closely investigating different spectral relaxation techniques which can be applied to graph-based unsupervised partitioning problems (Section 3.1). These methods are based on computing certain extremal eigenvectors either of the similarity matrix of the graph or of another matrix derived from it. We will show how these methods are related by considering them in a common framework based on scaled cut-cost functions.

Moreover, Section 3.2 briefly recapitulates the most important facts about the mean shift algorithm, which is an alternative unsupervised clustering technique that works directly in the (Euclidean) feature space of the extracted image elements. We will later apply this method to reduce the size of image segmentation problems arising in practice. Section 3.3 is devoted to an established technique in the context of supervised segmentation problems: the iterated conditional modes (ICM) algorithm is a local greedy technique that is basically motivated in the framework of probabilistic estimation based on Markov random fields. In particular, we will show how the underlying idea is related to the energy functional proposed in Section 2.3 for image restoration problems.

3.1 Spectral Techniques for Unsupervised Partitioning

As already stated in Section 2.1, unsupervised image segmentation can be reformulated as a graph partitioning problem: if the image is represented by a graph $G(V, E)$ with locally extracted image elements as vertices V and pairwise similarity values as edge weights $w_{ij} \in \mathbb{R}_0^+$, segmenting the image is equivalent to finding "good" cuts (of low weight) through the graph. Since many of the optimization problems that arise in this context are NP-hard, different approaches for efficiently computing suboptimal solutions have been proposed in the litera-

ture. In this section, we will compare some of these methods which are based on *spectral decompositions* of matrices connected with the graph. In particular, we will show how the corresponding eigenvector computations can be interpreted as relaxations of partitioning problems related to different cut measures.

To this end, we first introduce a general framework to define suitable measures for the quality of a segmentation that is based on graph cuts (Section 3.1.1). A general spectral relaxation for these measures is presented in Section 3.1.2, from which the Fiedler vector (Section 3.1.3) and the normalized cut (Section 3.1.4) relaxations can be directly derived. Moreover, we consider the relation to an average association measure (Section 3.1.5), and give a brief experimental comparison of the different relaxations (Section 3.1.6).

In the following, the same notation as in Section 2.1 is used: $W \in \mathcal{S}^n$ denotes the symmetric similarity matrix, $D = \mathrm{Diag}(We)$ the corresponding diagonal degree matrix, and $L = D - W$ the Laplacian matrix of the graph G. A partition $S, \overline{S}$ is indicated by the vector $x \in \{-1, +1\}^n$ with $x_i = 1$ for $i \in S$ and $x_i = -1$ for $i \in \overline{S}$. The weight of a cut then is given by (2.2) and (2.4): $\mathrm{cut}(S, \overline{S}) = \frac{1}{4} x^\top L x$.

3.1.1 Cut Measures

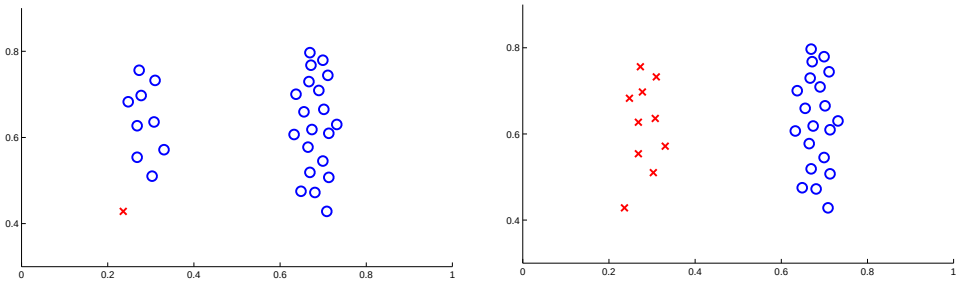
In order to identify a "good" binary partitioning based on the corresponding cut, we first have to define a suitable quality measure. The simplest idea certainly is to use the weight of the corresponding cut directly. This yields the following equivalent formulations of the *minimum cut* optimization problem:

$$\min_{S \subset V} \mathrm{cut}(S, \overline{S}) \;=\; \min_{x \in \{-1,+1\}^n} \; \frac{1}{4} x^\top L x \;=\; d(V) - \max_{x \in \{-1,+1\}^n} \; \frac{1}{4} x^\top W x \,, \quad (3.1)$$

where $d(V) = x^\top D x = e^\top W e$ denotes the sum of all edge weights. Since we assume the edge weights to be positive, finding the minimizer for (3.1) is easy: $S = V$ or equivalently $x = e$ will give a cut of weight 0. To derive more meaningful solutions, this trivial cut (or rather non-cut) is usually prohibited in practice, which leads to an optimization problem that can still be solved in polynomial time (see, e.g., [31] for an efficient algorithm).

A clustering method based on computing subsequent minimum cuts (3.1) is proposed in [198]. However, as the authors already notice in their work, the minimum cut criterion favors separating small sets of isolated vertices from the rest of the graph, which results in very unbalanced partitionings. The simple explanation for this fact is that for larger subsets S, the number of edges connecting S and $\overline{S}$ usually also increases (at least for regular graphs as they appear for image segmentation problems), which leads to higher cut-values. Figure 3.1 gives an illustrating example: assuming that all points are connected to each other with a weight inversely proportional to their Euclidean distance, cutting one point on the left from the rest of the points results in the best value for (3.1), whereas separating the points into a left and a right half gives a more reasonable clustering.

To avoid such unbalanced partitions, more suitable cut measures have been suggested in the literature. A popular approach is based on scaling the cut



Minimum cut Average/normalized cut

Figure 3.1: Example where the minimum cut criterion gives a bad partition: one point on the left is separated from the rest of the points, yielding a very unbalanced clustering. The average and normalized cut criteria produce a more reasonable clustering.

value: generally defining a positive weight $\omega_i > 0$ for each vertex i, the minimum of the following "scaled cut" objective function corresponds to a partition that simultaneously yields a small cut and gives large cluster weights for each part, which results in a balanced allocation of the vertices [45]:

$$f_{\mathrm{Gcut}}(S) := \frac{\mathrm{cut}(S,\overline{S})}{\omega(S)} + \frac{\mathrm{cut}(S,\overline{S})}{\omega(\overline{S})} \;=\; \omega(V)\frac{\mathrm{cut}(S,\overline{S})}{\omega(S)\;\omega(\overline{S})}\;, \tag{3.2}$$

where $\omega(S) := \sum_{i\in S}\omega_i$ denotes the sum of the vertex weights in S.

Depending on the choice of these vertex weights ω_i, we can distinguish two important cut measures that have attracted the interest of researchers:

- Using equal weights $\omega_i = 1$ for all vertices results in an objective function that favors similar cluster sizes $|S|$. The corresponding cut measure is called the *average cut* [159, 168] or the *ratio cut* [74, 29]:

$$f_{\mathrm{Acut}}(S) := \frac{\mathrm{cut}(S,\overline{S})}{|S|} + \frac{\mathrm{cut}(S,\overline{S})}{|\overline{S}|}\;. \tag{3.3}$$

- Shi and Malik [168] suggest to use the *degree* d_i of a vertex as its weight, $\omega_i = d_i := \sum_{j\in V} w_{ij}$, which leads to the *normalized cut* criterion:

$$f_{\mathrm{Ncut}}(S) := \frac{\mathrm{cut}(S,\overline{S})}{d(S)} + \frac{\mathrm{cut}(S,\overline{S})}{d(\overline{S})}\;, \tag{3.4}$$

  where $d(S) := \sum_{i\in S} d_i$ denotes the sum of the vertex degrees in S.

Hence, in comparison to the average cut, the normalized cut measure takes the edge weights within the subsets into account for balancing, instead of just the number of points contained in each part. The advantage of this idea becomes obvious when regarding the connection to the inner *association* $\mathrm{assoc}(S) := \sum_{i,j\in S} w_{ij}$ of the clusters, which measures the strength of the connections within

the subset S. As $\mathrm{cut}(S, \overline{S}) = d(S) - \mathrm{assoc}(S)$, the normalized cut becomes [168]

$$\begin{aligned} f_{\mathrm{Ncut}}(S) &= \frac{d(S) - \mathrm{assoc}(S)}{d(S)} + \frac{d(\overline{S}) - \mathrm{assoc}(\overline{S})}{d(\overline{S})} \\ &= 2 - \left(\frac{\mathrm{assoc}(S)}{d(S)} + \frac{\mathrm{assoc}(\overline{S})}{d(\overline{S})} \right) \\ &=: 2 - f_{\mathrm{Nassoc}}(S) \ . \end{aligned} \tag{3.5}$$

Thus minimizing the normalized cut is equivalent to maximizing the *normalized association* $f_{\mathrm{Nassoc}}(S)$, so that in fact two quality criteria are optimized simultaneously: on the one hand, the similarity between the two groups of the binary partition should be low, while on the other hand, the elements within each group should have strong associations on average.

In contrast to that, a relation like (3.5) does not hold (exactly) for the average cut criterion: defining the *average association* [168] analogously to the normalized association as

$$f_{\mathrm{Aassoc}}(S) := \frac{\mathrm{assoc}(S)}{|S|} + \frac{\mathrm{assoc}(\overline{S})}{|\overline{S}|} \ , \tag{3.6}$$

we only get

$$f_{\mathrm{Acut}}(S) = \frac{d(S)}{|S|} + \frac{d(\overline{S})}{|\overline{S}|} - f_{\mathrm{Aassoc}}(S) \ .$$

Hence, due to the non-constant first two terms, the optimization of both criteria is not equivalent. However, in case of similar vertex degrees ($d_i \approx \bar{d}$ for all i), minimizing the average cut closely approximates the maximum of the average association, as $\frac{d(S)}{|S|} + \frac{d(\overline{S})}{|\overline{S}|} \approx 2\bar{d}$ becomes nearly constant. Therefore, the average and normalized cut criteria can be expected to yield similar results unless the vertex degrees d_i vary considerably from each other. In this case, it can be argued in favor of the normalized cut measure that it better reflects the importance of the vertices by taking the corresponding degrees into account (cf. also [94]).[1]

Unfortunately, minimizing the scaled cut value (3.2) exactly is an NP-hard problem [61, 168]. However, approximate solutions can be found efficiently by reverting to eigenvector computations, as we will show in the next sections.

Finally, we note that other measures related to graph cuts were proposed in the literature recently, including a different ratio cut or mean cut, which scales the cut by the length of the boundary between the segments [189], contour-based ratio regions for object extraction [36], or typical cuts based on probability distributions over the set of possible segmentations [62, 165]. Since all these approaches result in optimization criteria which cannot be solved by spectral techniques, they are not considered here.

[1] Actually, the similarity measures used in image segmentation often yield varying degrees, which at least partly explains the recent success of the normalized cut criterion for applications in this field.

3.1.2 A General Relaxation

The following lemma provides a problem formulation which is equivalent to the minimization of the general scaled cut objective function f_{Gcut} from (3.2). This results in an optimization problem that can be solved approximately in an obvious way by calculating specific eigenvectors.

Lemma 3.1. *Assume that $S \neq \emptyset, \overline{S} \neq \emptyset$, and define $\beta := \sqrt{\frac{\omega(S)}{\omega(\overline{S})}}$. Moreover, let $\Omega = \text{Diag}(\omega)$ denote the diagonal matrix with the vertex weights ω_i on the diagonal. Minimizing the scaled cut* (3.2) *is equivalent to the following optimization problems:*

$$\begin{aligned} \min_x \quad & \frac{x^\top L x}{x^\top \Omega x}\, g(\beta) \\ s.t. \quad & x \in \{-1, +1\}^n \end{aligned} \tag{3.7}$$

with $g(\beta) := \frac{1}{4}(\beta^2 + \frac{1}{\beta^2} + 2)$, and

$$\begin{aligned} \min_y \quad & \frac{y^\top L y}{y^\top \Omega y} \\ s.t. \quad & y \in \{-\beta, \tfrac{1}{\beta}\}^n \\ & y^\top \Omega e = 0 \end{aligned} \tag{3.8}$$

with $y^\top \Omega y = \omega(V)$.[2]

Proof. Let $x \in \{-1, +1\}^n$ denote the indicator vector for the partition $S, \overline{S}$. Using the fact that $x^\top \Omega x = \omega(V) = \omega(S) + \omega(\overline{S})$, the scaled cut objective function (3.2) becomes

$$\begin{aligned} f_{\text{Gcut}}(S) = \frac{\text{cut}(S,\overline{S})}{\omega(S)} + \frac{\text{cut}(S,\overline{S})}{\omega(\overline{S})} &= (\omega(S) + \omega(\overline{S}))\frac{\frac{1}{4}x^\top L x}{\omega(S)\,\omega(\overline{S})} \\ &= \frac{(\omega(S) + \omega(\overline{S}))^2}{4\omega(S)\,\omega(\overline{S})}\frac{x^\top L x}{x^\top \Omega x} \\ &= \frac{x^\top L x}{x^\top \Omega x}\, g(\beta)\ , \end{aligned}$$

which proves the equivalence to (3.7). Observing that $e^\top L v = v^\top L e = 0$ for each $v \in \mathbb{R}^n$, we get (by expanding with $\frac{1}{\omega(S)^2}$)

$$\begin{aligned} f_{\text{Gcut}}(S) &= \frac{\left(\frac{\omega(V)}{2\omega(S)}x\right)^\top L \left(\frac{\omega(V)}{2\omega(S)}x\right)}{\frac{\omega(S)}{\omega(\overline{S})}x^\top \Omega x} + 0 \\ &= \frac{\left(\frac{1}{2}(1+\beta^2)x\right)^\top L \left(\frac{1}{2}(1+\beta^2)x\right)}{\beta^2 x^\top \Omega x} + \frac{\frac{1}{4}(1-\beta^2)^2 e^\top L e + \frac{1}{2}(1-\beta^4)e^\top L x}{\beta^2 x^\top \Omega x} \\ &= \frac{\left(\frac{1}{2\beta}\left((1+\beta^2)x + (1-\beta^2)e\right)\right)^\top L \left(\frac{1}{2\beta}\left((1+\beta^2)x + (1-\beta^2)e\right)\right)}{x^\top \Omega x}\ , \end{aligned}$$

[2] Note that Shi and Malik [168] obtain a slightly different problem formulation for the normalized cut: instead of $y \in \{-\beta, \frac{1}{\beta}\}^n$ they have $y \in \{-\beta^2, 1\}^n$. However, this is equivalent to (3.8), as can be seen easily by replacing y with βy.

which corresponds to the objective function $\frac{y^\top Ly}{y^\top \Omega y}$ in (3.8) by defining the generalized partition vector

$$y := \frac{1}{2}\left((\tfrac{1}{\beta}+\beta)x + (\tfrac{1}{\beta}-\beta)e\right) . \tag{3.9}$$

The equality of the denominators is derived by employing the facts that $e^\top \Omega x = \omega(S) - \omega(\overline{S})$ and $x^\top \Omega x = e^\top \Omega e = \omega(S) + \omega(\overline{S})$:

$$\begin{aligned} y^\top \Omega y &= \tfrac{1}{4}(\tfrac{1}{\beta}+\beta)^2 x^\top \Omega x + \tfrac{1}{2}(\tfrac{1}{\beta}+\beta)(\tfrac{1}{\beta}-\beta)e^\top \Omega x + \tfrac{1}{4}(\tfrac{1}{\beta}-\beta)^2 e^\top \Omega e \\ &= \tfrac{1}{2}(\tfrac{1}{\beta^2}+\beta^2)(\omega(S)+\omega(\overline{S})) + \tfrac{1}{2}(\tfrac{1}{\beta^2}-\beta^2)(\omega(S)-\omega(\overline{S})) \\ &= \frac{1}{\beta^2}\omega(S) + \beta^2\omega(\overline{S}) \\ &= \omega(\overline{S}) + \omega(S) \\ &= x^\top \Omega x . \end{aligned}$$

The definition (3.9) of y now directly implies $y_i \in \{-\beta, \frac{1}{\beta}\}$. Finally observe that

$$\begin{aligned} y^\top \Omega e &= \tfrac{1}{2}\big((\tfrac{1}{\beta}+\beta)x^\top \Omega e + (\tfrac{1}{\beta}-\beta)e^\top \Omega e\big) \\ &= \tfrac{1}{2}\big(2\tfrac{1}{\beta}\omega(S) - 2\beta\omega(\overline{S})\big) \\ &= \beta\left(\frac{\omega(\overline{S})}{\omega(S)}\omega(S) - \omega(\overline{S})\right) = 0 , \end{aligned}$$

which proves the equivalence to (3.8). □

As the weight quotient β is not known in advance, the optimization problem (3.8) still is intractable. However, the objective function now equals a *Rayleigh quotient* [69] of the generalized indicator vector y, which is related to the calculation of generalized eigenvectors for the matrix pair L and Ω. This observation suggests to relax (3.8) by dropping the intractable constraint on the entries of y. The resulting optimization problem can then be solved (in polynomial time) by using the following standard result from linear algebra (which is just another way of expressing the Courant-Fisher eigenvalue characterization, see Theorem A.3):

Theorem 3.2. *For a symmetric matrix L and a symmetric positive definite matrix Ω, finding the k-th smallest eigenvalue λ_k and a corresponding eigenvector v_k of the generalized eigenvalue problem*

$$Lv = \lambda \Omega v \tag{3.10}$$

is equivalent to solving the problem

$$\begin{aligned} \lambda_k = \min_{v \neq 0} \quad & \frac{v^\top Lv}{v^\top \Omega v} \\ \textit{s.t.} \quad & v^\top \Omega v_j = 0 \quad \textit{for } j = 1, \ldots, k-1 . \end{aligned} \tag{3.11}$$

In our case, Ω is positive definite as it corresponds to a diagonal matrix with positive entries only. Moreover, since $Le = 0$, e is an eigenvector of the generalized eigenvalue problem (3.10) corresponding to the eigenvalue $\lambda_1 = 0$. There are also no smaller eigenvalues: as the Laplacian L is positive semidefinite (because $v^\top Lv = \frac{1}{2}\sum_{i,j\in V} w_{ij}(v_i - v_j)^2 \geq 0$ for each $v \in \mathbb{R}^n$, cf. (2.4)), the Rayleigh quotient in (3.11) is always positive.

Hence, by virtue of Theorem 3.2, the relaxation of (3.8) becomes to compute the eigenvector v_2 corresponding to the *second smallest* eigenvalue λ_2 of the generalized eigenvalue problem (3.10). This results in the following bound on the scaled cut cost function:

$$\lambda_2 = \frac{v_2^\top L v_2}{{v_2}^\top \Omega v_2} \leq \min_{S\subset V} f_{\mathrm{Gcut}}(S) \; . \tag{3.12}$$

In the next sections, we will study this relaxation in more detail for the special cases of the average and the normalized cut measure, respectively. Beforehand however, we answer the important question of how a good *binary* solution can be obtained from the *continuous* eigenvector $v_2 \in \mathbb{R}^n$. The idea is quite simple: as the entries of the solution vector y of the unrelaxed problem formulation (3.8) take on either of two discrete values (which correspond to the ± 1-entries of the indicator vector x, but are not known beforehand), we get an integer solution x from v_2 by thresholding the eigenvector using some suitable splitting value t. The final partitioning thus is specified by the indicator vector x with entries $x_i = 1$ for $v_{2,i} > t$ and $x_i = -1$ for $v_{2,i} \leq t$, which induces the sets $S = \{i \in V \,|\, v_{2,i} > t\}$ and $\overline{S} = \{i \in V \,|\, v_{2,i} \leq t\}$. Several popular choices for the threshold value t arise naturally [174]:

- Set $t = 0$, which results in splitting according to the sign of the eigenvector entries. This criterion is motivated by the fact that before the relaxation, the entries of the generalized indicator vector y in (3.8) are either positive or negative.

- Set t equal to the median of the eigenvector entries $v_{2,i}$. This criterion results in an equipartition of the vertices ($|S| = |\overline{S}|$ or $|S| = |\overline{S}| - 1$).

- Set t to any value in the largest gap in the sorted list of eigenvector entries. This criterion is motivated by the interpretation that the eigenvector entries are noisy measurements of two different discrete values.

- Set t so that the corresponding indicator vector x gives the best value for the original objective function (3.2).

While the last choice obviously achieves the best value for the original partitioning criterion, it also is the most expensive threshold value to compute, as it requires sorting the entries of v_2 and calculating n objective function values. For large problems, a variation is therefore often used in practice: instead of testing each possible threshold, only a certain number of equally spaced splitting points is evaluated.

3.1.3 The Fiedler Vector

For the average cut measure (3.3), the vertex weights $\omega_i = 1$ result in the weight matrix $\Omega = I$. In this case, (3.8) yields the relaxation

$$\begin{aligned} \lambda_2(L) = \min_{z \in \mathbb{R}^n} \quad & \frac{z^\top L z}{z^\top z} \\ \text{s.t.} \quad & z^\top e = 0 \,, \end{aligned} \tag{3.13}$$

which can be solved by computing the eigenvector v_2 corresponding to the second smallest eigenvalue $\lambda_2(L)$ of the Laplacian of the graph. This directly leads to the following lower bound on the optimal average cut (3.3):

$$\lambda_2(L) \leq \min_{S \subset V} f_{\text{Acut}}(S) \,. \tag{3.14}$$

A corresponding suboptimal indicator vector x is finally obtained from v_2 by applying any of the thresholding methods mentioned at the end of the previous section.

Fiedler [51] was probably the first to analyze the properties of this special eigenvector in connection with graph partitionings by considering the components of v_2 as *characteristic valuations* of the corresponding vertices of the graph G. Therefore, the second smallest eigenvalue $\lambda_2(L)$ and the corresponding eigenvector v_2 are usually called the *Fiedler value* and the *Fiedler vector* of the Laplacian, respectively. Since this early work, the Fiedler vector has been the subject of extensive research, especially in connection with graph partitioning [47, 19, 146, 74, 29, 174, 73]. Concerning image segmentation, a successful application of the Fiedler vector approximation for perceptual organization problems was recently presented in [159].

Only in some of the mentioned literature, however, the Fiedler vector was considered in the way we derived it, namely as a relaxation to the average cut objective function (e.g. in [74, 29]). In fact, it was mainly used to approximate other optimization criteria. Accordingly, two further interpretations of the Fiedler vector are meaningful:

Equipartitioning. As already mentioned in Section 2.1, an alternative to scaling the cut value to prevent unbalanced solutions is to introduce an additional linear constraint $e^\top x = 0$ to get equally sized parts.[3] This results in the classical equipartition (or bisection) problem (2.5) [129]. Dropping the integer constraint on the indicator vector x, and instead using the fact that its norm is equal to n, we obtain the relaxation

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & z^\top L z \\ \text{s.t.} \quad & z^\top z = n \\ & z^\top e = 0 \,, \end{aligned}$$

[3] If the number n of vertices of the graph is not even, this constraint results in an infeasible problem. However, this case can be circumvented by adding an artificial, unconnected vertex $n + 1$ if necessary: this increases the problem size by one, without changing the objective function value.

which obviously is equivalent to (3.13) up to the constant factor $\sqrt{n}$ on z. Thus, a lower bound on the weight of the optimal equipartition (the so-called "bisection width") is given by $n\lambda_2(L)$. In Section 4.3, we will see how this bound can be improved by using a semidefinite relaxation approach.

In this sense, the Fiedler vector was successfully applied e.g. to find good orderings for the parallel factorization of matrices [146], to load balancing problems in parallel computation [9, 83], and for unsupervised learning [38]. Other eigenvalue-based lower bounds for equipartitioning were presented in [47].

Isoperimetric number. Closely related to the average cut criterion is the *isoperimetric number* i_G of a graph G (see, e.g., [129]), which is also NP-hard to determine [127]. If we define the *expansion* $\psi(S)$ of a cut as the quotient of the cut and the smaller of the two emerging parts,

$$\psi(S) := \frac{\mathrm{cut}(S, \overline{S})}{\min\{|S|, |\overline{S}|\}} ,$$

the isoperimetric number measures the minimal possible relative cut size for a graph in terms of the expansion:

$$\begin{aligned} i_G &:= \min_{S \subset V} \psi(S) \\ &= \min_{S \subset V,\, 0 < |S| \leq \frac{n}{2}} \frac{\mathrm{cut}(S, \overline{S})}{|S|} . \end{aligned} \tag{3.15}$$

Hence, i_G minimizes only the larger of the two summands of the average cut (3.3). A direct connection between the two measures is easy to derive, since for each $S \subset V$, we have

$$f_{\mathrm{Acut}}(S) \leq 2 \max \left\{ \frac{\mathrm{cut}(S, \overline{S})}{|S|}, \frac{\mathrm{cut}(S, \overline{S})}{|\overline{S}|} \right\} = 2\psi(S) ,$$

which results in $\min_{S \subset V} f_{\mathrm{Acut}}(S) \leq 2 i_G$. This shows that the Fiedler value can also be interpreted as an approximation to the isoperimetric number i_G.

The inequality (3.14) then directly yields the lower bound on the isoperimetric number given in the following theorem; the upper bound is proven in [127]:

Theorem 3.3. *Let G be a graph with $n \geq 3$, and denote by $\delta = \max_i d_i$ the maximal vertex degree in G. Then*

$$\tfrac{1}{2}\lambda_2(L) \leq i_G \leq \sqrt{(2\delta - \lambda_2(L))\lambda_2(L)} .$$

It can be shown [128, 73] that these inequalities also hold for the best approximation $\psi(S^*)$ to i_G that can be obtained by thresholding the eigenvector v_2; therefore, Theorem 3.3 also yields the following upper bound on the average cut value of the best partitioning $S^*, \overline{S^*}$ acquired from the Fiedler vector via thresholding:

$$\lambda_2(L) \leq f_{\mathrm{Acut}}(S^*) \leq 2\sqrt{(2\delta - \lambda_2(L))\lambda_2(L)} .$$

However, these bounds do not provide any information on how close to the real optimal value $\min_{S \subset V} f_{\text{Acut}}(S)$ this partitioning is.

While more theoretical results concerning the quality of partitionings obtained from the Fiedler vector seem to be missing for the average cut criterion, this topic has been studied in connection with the isoperimetric number i_G. The corresponding results indicate that the approximation quality depends on the underlying graph: on the one hand, Spielman and Teng [174] prove that for special graphs which often arise in practice (bounded degree planar graphs and finite element meshes), partitioning based on the Fiedler vector indeed gives good approximations to i_G. On the other hand, it can also be shown that for other graph types (which are less common in practice), spectral partitioning as approximation to i_G may perform poorly [73].

For more information on properties and applications of the Fiedler value and the Fiedler vector, we refer to [129, 39, 128].

3.1.4 Normalized Cut Relaxation

For the normalized cut measure (3.4), the vertex weights $\omega_i = d_i$ result in the weight matrix being equal to the degree matrix $\Omega = D$. Hence from (3.8) we obtain the relaxation

$$\begin{aligned} \lambda_2(L, D) = \min_{z \in \mathbb{R}^n} \quad & \frac{z^\top L z}{z^\top D z} \\ \text{s.t.} \quad & z^\top D e = 0 \,, \end{aligned} \tag{3.16}$$

which is solved by computing the second smallest eigenvalue $\lambda_2(L, D)$ and the corresponding eigenvector v_2 of the generalized eigenvalue problem

$$Lz = \lambda D z \,. \tag{3.17}$$

This special eigenvalue problem is equivalent to several other eigenvalue problems [190, 168, 125]. To derive the corresponding connections, define the *normalized Laplacian* L' of the graph G as

$$L' := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = I - W' \tag{3.18}$$

with the corresponding normalized similarity matrix W', and the (asymmetric) *stochastic similarity matrix* P as

$$P := D^{-1} W \,. \tag{3.19}$$

In this context, we assume that D is positive definite, so that the inverse D^{-1} and its square root $D^{-\frac{1}{2}}$ are well defined.

The following normalization lemma now summarizes how these matrices are linked to each other:

Lemma 3.4. *(a) If v denotes an eigenvector of the generalized eigenvalue problem* (3.17) *with the corresponding eigenvalue λ, then*

- *$u := D^{\frac{1}{2}} v$ is an eigenvector of L' with corresponding eigenvalue λ,*

- *u is an eigenvector of W' with corresponding eigenvalue $1-\lambda$,*
- *v is a (right) eigenvector of P with corresponding eigenvalue $1-\lambda$.*

(b) The matrix L' is positive semidefinite, and its eigenvalues lie in the interval $[0,2]$. The eigenvalues of W' and P lie in the interval $[-1,1]$, and the largest eigenvalue is $\lambda_n(W') = \lambda_n(P) = 1$.

Proof. (a) Using $v = D^{-\frac{1}{2}}u$, the propositions are shown by the following equivalence transformations:

$$\begin{aligned}
& Lv = \lambda Dv \\
\Leftrightarrow \quad & LD^{-\frac{1}{2}}u = \lambda DD^{-\frac{1}{2}}u \\
\Leftrightarrow \quad & L'u = D^{-\frac{1}{2}}LD^{-\frac{1}{2}}u = \lambda u \\
\Leftrightarrow \quad & Iu - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}u = \lambda u \\
\Leftrightarrow \quad & W'u = (1-\lambda)u \\
\Leftrightarrow \quad & D^{-\frac{1}{2}}W'D^{\frac{1}{2}}v = (1-\lambda)D^{-\frac{1}{2}}D^{\frac{1}{2}}v \\
\Leftrightarrow \quad & Pv = D^{-1}Wv = (1-\lambda)v\ .
\end{aligned}$$

(b) As the smallest eigenvalue of the generalized eigenvalue problem (3.17) is 0 (see Section 3.1.2), L' must be positive semidefinite. Furthermore, using (2.4) and the fact that $(v_i - v_j)^2 \leq 2(v_i^2 + v_j^2)$, we get for each eigenvalue λ of L' with the corresponding eigenvector $u = D^{\frac{1}{2}}v$:

$$\begin{aligned}
\lambda = \frac{v^\top Lv}{v^\top Dv} &= \frac{\frac{1}{2}\sum_{ij} w_{ij}(v_i - v_j)^2}{\sum_i d_i v_i^2} \\
&\leq \frac{\sum_{ij} w_{ij}(v_i^2 + v_j^2)}{\sum_i d_i v_i^2} \\
&= \frac{\sum_i d_i v_i^2 + \sum_j d_j v_j^2}{\sum_i d_i v_i^2} = 2\ .
\end{aligned}$$

The propositions on the eigenvalues of W' and P now directly follow from (a). □

This lemma shows that approximations to the normalized cut can conveniently be found by computing the eigenvector corresponding to the second smallest eigenvalue of L', or to the second largest eigenvalue of W' or P, respectively. If the normalized Laplacian L' or the similarity matrix W' are used, this eigenvector u_2 additionally has to be transformed by $v_2 = D^{-\frac{1}{2}}u_2$ to get the correct solution v_2 for the relaxation (3.16).[4] Thresholding v_2 with any of the methods mentioned at the end of Section 3.1.2 finally yields an approximate

[4] Weiss [190] points out that the vector v_2 can also be interpreted as component-wise ratio of the eigenvectors $u_1 = D^{\frac{1}{2}}e$ and u_2 corresponding to the two largest eigenvalues of W': $v_{2,i} = \frac{u_{2,i}}{u_{1,i}}$.

integer solution to the normalized cut criterion (3.4). As for the average cut, we directly obtain $\lambda_2(L')$ as a lower bound on the optimal normalized cut:

$$\lambda_2(L') \leq \min_{S \subset V} f_{\mathrm{Ncut}}(S) \; . \tag{3.20}$$

Via the stochastic matrix P, there is a strong relationship between the normalized cut and the theory of Markov random walks [125]: as each matrix entry P_{ij} can be interpreted as representing the probability of moving from vertex i to vertex j in one step, it follows that the normalized cut (3.4) measures the total probability of a random walk to transition from any vertex in S to any vertex in $\overline{S}$ or vice versa in one step. Thus, a small normalized cut corresponds to a partition such that a random walk tends to remain in each of the parts once it is in it [125].

In this context — similar to the connection between the average cut and the isoperimetric number — there are also strong analogies of the normalized cut with the (generalized) *Cheeger constant* h_G of a graph G [32, 94], which can be used to bound the mixing time of a Markov random walk [170]. In order to define h_G, consider the *conductance* $\phi(S)$ of a cut,

$$\phi(S) := \frac{\mathrm{cut}(S, \overline{S})}{\min\{d(S), d(\overline{S})\}} \; ,$$

which is a direct generalization of the expansion $\psi(S)$ introduced in the previous section. Analogously to the isoperimetric number, the Cheeger constant is then defined as

$$h_G = \min_{S \subset V} \; \phi(S) \; , \tag{3.21}$$

and thus measures the minimal possible relative cut size by scaling with the smaller degree of the corresponding parts.

As in the previous section, we easily get $\min_{S \subset V} f_{\mathrm{Ncut}}(S) \leq 2h_G$, which by virtue of (3.20) results in a lower bound on the Cheeger constant and indicates that the relaxation (3.16) can also be interpreted as approximation to h_G. The corresponding upper bound given in the following theorem is proven in [18] as a generalization of the proof in [32]:

Theorem 3.5. *For any connected graph G we have*

$$\tfrac{1}{2}\lambda_2(L') \leq h_G \leq \sqrt{(2 - \lambda_2(L'))\lambda_2(L')} \; .$$

In contrast to Theorem 3.3, however, it is not clear if this theorem also yields bounds on the partitioning obtained by optimally thresholding v_2 based on the conductance ϕ or the normalized cut measure [32]. Nevertheless, by analogy we reckon that similar bounds as in the previous section can be established, at least if the edge weights are known to be smaller than one.[5]

Since the pioneering work of Shi and Malik [168], who successfully applied the normalized cut criterion to image segmentation problems, there has been

[5] This assumption is motivated by Corollary 2.4 in [32], from which such a bound can be derived in this case.

great interest among researchers in this method: besides several variants that have been proposed [94, 118, 134, 54], cases where the relaxation works optimal in theory were presented in [125, 134]. Moreover, the normalized cut has been applied to such diverse fields as motion segmentation and tracking [167], object recognition [203], document clustering [45], analyzing and visualizing networks [162], and transductive learning [92]. For more information on the theory of the normalized Laplacian, we refer to [32].

3.1.5 Average Association Approximation

Lemma 3.4 shows that the relaxation of the normalized cut criterion can equivalently be solved by computing the eigenvector corresponding either to the second smallest eigenvalue of the normalized Laplacian L' or to the second largest eigenvalue of the normalized similarity matrix W'. This connection was already indicated by the relation (3.5), which comprises that minimizing the normalized cut measure f_{Ncut} is equivalent to maximizing the normalized association f_{Nassoc}. In fact, by virtue of Lemma 3.1 and (3.18), we get

$$f_{\text{Nassoc}}(S) = 2 - \frac{y^\top L' y}{y^\top y} = \frac{y^\top W' y}{y^\top y} + 1$$

(with y defined as in (3.9)), which reveals the direct connection between W' and the normalized association measure.

Unfortunately, a direct relation like (3.5) does not hold between the average cut measure f_{Acut} and the average association f_{Aassoc} (see Section 3.1.1), which opens up the question for an appropriate relaxation for the problem of maximizing the average association measure (3.6). Using a similar derivation as in Lemma 3.1, the following lemma establishes that this analogously to the normalized association f_{Nassoc} can be realized by resorting to the original weight matrix W.

Lemma 3.6. *Assume that $S \neq \emptyset, \overline{S} \neq \emptyset$, and define $\beta := \sqrt{\frac{|S|}{|\overline{S}|}}$. Maximizing the average association $f_{Aassoc}(S)$ is equivalent to the following optimization problem:*

$$\begin{aligned} \max_{y} \quad & \frac{y^\top W y}{y^\top y} \\ s.t. \quad & y \in \{-\beta, \tfrac{1}{\beta}\}^n \\ & y^\top e = 0 \end{aligned} \tag{3.22}$$

with $y^\top y = n$. The objective functions are related by

$$f_{Aassoc}(S) = \frac{y^\top W y}{y^\top y} + \frac{d(V)}{n} .$$

Proof. Let $x \in \{-1, +1\}^n$ denote the indicator vector for the partition $S, \overline{S}$, and define $y = \frac{1}{2}((\frac{1}{\beta} + \beta)x + (\frac{1}{\beta} - \beta)e)$ as in (3.9). By substituting $\Omega = I$ in

Lemma 3.1, we see that $y_i \in \{-\beta, \frac{1}{\beta}\}$, $y^\top e = 0$ and $y^\top y = n$. Using the fact that $|S| + |\overline{S}| = n$, the average association (3.6) becomes

$$\begin{aligned}
f_{\text{Aassoc}}(S) &= \frac{\text{assoc}(S)}{|S|} + \frac{\text{assoc}(\overline{S})}{|\overline{S}|} \\
&= \frac{1}{4}\left(\frac{1}{|S|}(x+e)^\top W(x+e) + \frac{1}{|\overline{S}|}(x-e)^\top W(x-e)\right) \\
&= \frac{1}{4n}\left((1+\tfrac{1}{\beta^2})(x+e)^\top W(x+e) + (1+\beta^2)(x-e)^\top W(x-e)\right) \\
&= \frac{1}{4n}\left((\tfrac{1}{\beta}+\beta)^2(x^\top Wx + e^\top We) + 2(\tfrac{1}{\beta^2}-\beta^2)e^\top Wx\right) \\
&= \frac{y^\top Wy}{y^\top y} + \frac{1}{4n}\left((\tfrac{1}{\beta}+\beta)^2 e^\top We - (\tfrac{1}{\beta}-\beta)^2 e^\top We\right) \\
&= \frac{y^\top Wy}{y^\top y} + \frac{4e^\top We}{4n} \\
&= \frac{y^\top Wy}{y^\top y} + \frac{d(V)}{n}\,.
\end{aligned}$$

Since $\frac{d(V)}{n}$ is constant, this proves the statement of the lemma. □

As for the average cut problem (3.13), the objective function in (3.22) equals a Rayleigh quotient of y, but this time for the matrix W. Hence, we can think of approximating the solution by neglecting the constraints on y, and computing the eigenvector v_n belonging to the largest eigenvalue of W. A corresponding partitioning is again obtained by applying any of the thresholding techniques described at the end of Section 3.1.2. However, in contrast to the relaxation (3.13) of the average cut problem, this relaxation does not automatically take into account the balancing constraint $y^\top e = 0$, as e is *no eigenvector* of W. Thus fewer original constraints than in (3.13) are adhered, which obviously leads to a weaker relaxation in terms of the quality of the corresponding solutions.

Nevertheless, the largest eigenvector v_n of W has been used for solving partitioning problems (though differently motivated), especially in the context of perceptual grouping [158, 142]. For instance, Sarkar and Boyer [158] define a measure of "cluster-cohesiveness" as $y^\top Wy$ for real vectors y from the unit sphere (i.e. with $y^\top y = 1$). To find maximally coherent clusters according to this measure, they compute the eigenvectors corresponding to the largest eigenvalues of W. Interpreting the entries of y to capture the participation of each graph vertex in a cluster, a so-called "eigencluster" is then derived from the dominant (largest magnitude) entries in these eigenvectors.

Perona and Freeman [142] follow a different idea: they compute the largest eigenvector v_n of the similarity matrix W to obtain the best rank one approximation to W in Frobenius norm: $W \approx v_n v_n^\top$. Similar to [158], this eigenvector is used as a saliency function of the vertices by observing that the entries of v_n which are larger than a certain threshold value (close to 0) usually belong to "foreground" objects. By comparing their approach with the normalized cut technique [168], they claim that v_n also approximates the minimum of an

asymmetric version of $f_{\mathrm{Ncut}}(S)$, namely the so-called "foreground cut" $\frac{\mathrm{cut}(S,\overline{S})}{\mathrm{assoc}(S)}$. However, this interpretation is not correct.[6] Instead, v_n rather approximates the maximum *average foreground association* $\frac{\mathrm{assoc}(S)}{|S|}$ [168], because

$$\max_{S\subset V} \frac{\mathrm{assoc}(S)}{|S|} = \max_{x\in\{0,1\}^n} \frac{x^\top W x}{x^\top x} \tag{3.23}$$

can be relaxed by taking $x \in \mathbb{R}^n$. This idea explains the success of using the eigenvector v_n for the perceptual grouping task: as only the foreground objects need to satisfy some coherency criterion (large association), the second term in the average association measure $f_{\mathrm{Aassoc}}(S)$ may be abandoned, thus rendering the balancing constraint $y^\top e = 0$ in (3.22) unnecessary.

However, to find partitionings with both parts being coherent according to the average association criterion (3.6), we have to find a way to incorporate the balancing constraint $y^\top e = 0$ from (3.22) into the relaxation appropriately. The following lemma shows that this can be done by *centering* the similarity matrix W:

Lemma 3.7. *Let $Q := I - \frac{1}{n}E$ denote the projection matrix onto the orthogonal complement of e, and define the* centered similarity matrix *$\tilde{W} := QWQ$. Furthermore, as in Lemma 3.6 assume that $S \neq \emptyset, \overline{S} \neq \emptyset$, and define $\beta := \sqrt{\frac{|S|}{|\overline{S}|}}$. Then maximizing the average association* (3.6) *resp.* (3.22) *is equivalent to:*

$$\begin{aligned} \max_{y} \quad & \frac{y^\top \tilde{W} y}{y^\top y} \\ s.t. \quad & y \in \{-\beta, \tfrac{1}{\beta}\}^n \end{aligned} \tag{3.24}$$

with $y^\top y = n$.

Proof. Observing that $Qe = (I - \frac{1}{n}E)e = e - \frac{1}{n}ne = 0$, we get for $y = \frac{1}{2}((\frac{1}{\beta} + \beta)x + (\frac{1}{\beta} - \beta)e)$ as in (3.9):

$$\begin{aligned} Qy &= \frac{1}{2}\left((\tfrac{1}{\beta}+\beta)Qx + (\tfrac{1}{\beta}-\beta)Qe\right) \\ &= \frac{1}{2}\left((\tfrac{1}{\beta}+\beta)x - (\tfrac{1}{\beta}+\beta)\tfrac{1}{n}Ex\right) \\ &= \frac{1}{2}\left((\tfrac{1}{\beta}+\beta)x - (\tfrac{1}{\beta}+\beta)\frac{|S|-|\overline{S}|}{n}e\right) \\ &= \frac{1}{2}\left((\tfrac{1}{\beta}+\beta)x - \beta\frac{|\overline{S}|+|S|}{|S|}\frac{|S|-|\overline{S}|}{|S|+|\overline{S}|}e\right) \\ &= \frac{1}{2}\left((\tfrac{1}{\beta}+\beta)x - \beta(1-\tfrac{1}{\beta^2})e\right) \\ &= y\ . \end{aligned}$$

[6] In their derivation, Perona and Freeman claim that $z^\top u$ subject to $\|z\| = 1$ is minimized by the vector $z = \pm(0,\ldots,0,1,0,\ldots,0)^\top$ with the 1 indicating the position of the largest entry of u. This is not true in general: as $\|u\| \leq \max_i |u_i|$, the vector $z = -\frac{u}{\|u\|}$ gives a smaller value. In fact, one can verify that the foreground cut criterion leads to the problem of finding the maximum eigenvector of the normalized similarity matrix W', which is equal to e and thus useless for partitioning.

Hence, without loss of generality we can replace y with Qy in (3.22). As now $(Qy)^\top e = y^\top Qe = 0$, the second constraint in (3.22) is automatically satisfied, and thus can be dropped from the optimization problem. □

This lemma shows that we can appropriately relax the problem of maximizing the average association (3.6) by computing the eigenvector v_n corresponding to the largest eigenvalue $\lambda_n(\tilde{W})$ of the centered similarity matrix $\tilde{W}$. Thus we get the upper bound:

$$\lambda_n(\tilde{W}) \geq \max_{S \subset V} f_{\mathrm{Aassoc}}(S) \; . \tag{3.25}$$

A corresponding binary solution is again obtained by applying one of the thresholding techniques described at the end of Section 3.1.2. In fact, with a different notion of the objective function, the bound (3.25) was already given in [129], based on a proof for unweighted graphs from [93].

The idea to compute clusterings based on the eigenvector corresponding to the largest eigenvalue of the centered similarity matrix $\tilde{W}$ also appears in the context of unsupervised learning [38], where the alignment of a fixed kernel (which is represented by W in our notation) with a set of labels (corresponding to the ± 1-entries of the indicator vector x in our notation) is approximately maximized in this way. Actually, the alignment measures how good a rank one matrix approximates the given kernel matrix, and is thus equivalent to the approach used in [142] for perceptual grouping. This explains why the use of the centered kernel matrix has become very popular in the learning community [124]: it results in better clusterings by effectively including a balancing constraint on the labels.

3.1.6 Experimental Results

All the relaxations presented in the previous sections result in the calculation of extremal eigenvectors of different matrices: either one corresponding to the largest (or second largest) eigenvalue of the (potentially centered or normalized) similarity matrix, or one corresponding to the second smallest eigenvalue of the (potentially normalized) Laplacian matrix (see Table 3.1). Such eigenvectors can be computed efficiently for large matrices with iterative techniques like the Lanczos method [69]. In particular, this numerical algorithm is able to exploit any kind of sparsity structure, a property that is often encountered for image partitioning problems (e.g. if only spatially neighboring pixels are connected). Moreover, the eigenvector entries are not required to be calculated with high accuracy, as they are subject to be thresholded afterwards. Putting all this together, the final approximative solution based on any of the measures presented can be obtained in running time $\mathcal{O}(n^{\frac{3}{2}})$ for typical image segmentation problems [168], if we only test a fixed number of possible threshold values. Hence, even for larger images with a few thousand pixels, the solution is calculated in less than one minute.[7]

[7] Unless stated otherwise, all the computation times stated in this work were measured on currently available 2 or 3 GHz Pentium IV Linux PCs.

Original criterion	**Relaxation**	**Solution**
Average cut $f_{\mathrm{Acut}}(S)$ (3.3)		
$\min \frac{\mathrm{cut}(S,\overline{S})}{\lvert S\rvert} + \frac{\mathrm{cut}(S,\overline{S})}{\lvert\overline{S}\rvert}$	$\min\limits_{z^\top e=0} \frac{z^\top L z}{z^\top z}$ (3.13)	$\lambda_2(L)$
Normalized cut $f_{\mathrm{Ncut}}(S)$ resp. **normalized association** $f_{\mathrm{Nassoc}}(S)$ (3.4)		
$\min \frac{\mathrm{cut}(S,\overline{S})}{d(S)} + \frac{\mathrm{cut}(S,\overline{S})}{d(\overline{S})}$ $= 2 - \max \frac{\mathrm{assoc}(S)}{d(S)} + \frac{\mathrm{assoc}(\overline{S})}{d(\overline{S})}$	$\min\limits_{z^\top De=0} \frac{z^\top L z}{z^\top D z} = \min\limits_{z^\top D^{\frac{1}{2}} e=0} \frac{z^\top L' z}{z^\top z}$ $= 1 - \max\limits_{z^\top D^{\frac{1}{2}} e=0} \frac{z^\top W' z}{z^\top z}$ (3.16)	$\lambda_2(L,D) = \lambda_2(L')$ $= 1 - \lambda_{n-1}(W')$
Average association $f_{\mathrm{Aassoc}}(S)$ (3.6)		
$\max \frac{\mathrm{assoc}(S)}{\lvert S\rvert} + \frac{\mathrm{assoc}(\overline{S})}{\lvert\overline{S}\rvert}$	$\max\limits_{z} \frac{z^\top \tilde{W} z}{z^\top z}$ (3.24)	$\lambda_n(\tilde{W})$
Average foreground association (3.23)		
$\max \frac{\mathrm{assoc}(S)}{\lvert S\rvert}$	$\max\limits_{z} \frac{z^\top W z}{z^\top z}$ (3.22)	$\lambda_n(W)$

Table 3.1: Overview of the spectral partitioning techniques presented in this section. While normalized cut and normalized association are equivalent, average cut and average association yield different relaxations.

Comparisons of spectral relaxation methods were presented by several authors, both in theory and concerning the application to image segmentation [190, 168, 173, 162]. In this work, we therefore provide only a few results which especially demonstrate the differences of the spectral techniques. Since the following examples are quite small, we always search for the threshold value on the eigenvector which results in the best value of the corresponding objective function.

Concerning the similarity measure, we compute the entries of W for a given problem from normalized distances $d(i,j) \in [0,1]$ between the extracted image features i and j as

$$w_{ij} = e^{-\left(\frac{d(i,j)}{\sigma}\right)^2},$$

where σ is usually set to a value between 5% and 30% of the maximal distance encountered in the problem [168]. More intricate similarity measures can of course be conceived [149]; see also Section 4.4.2. However, since the focus of this work is on analyzing different relaxations of hard problems from the optimization point of view, we do not elaborate on the issue of similarity measures here.

As a first result, Figure 3.2 shows the eigenvectors obtained for each of the spectral relaxation methods for the clustering problem depicted in Figure 3.1. In this case, the similarity matrix is calculated based on the normalized Euclidean distances between all the points, with $\sigma = 0.1$. While the approximating eigen-

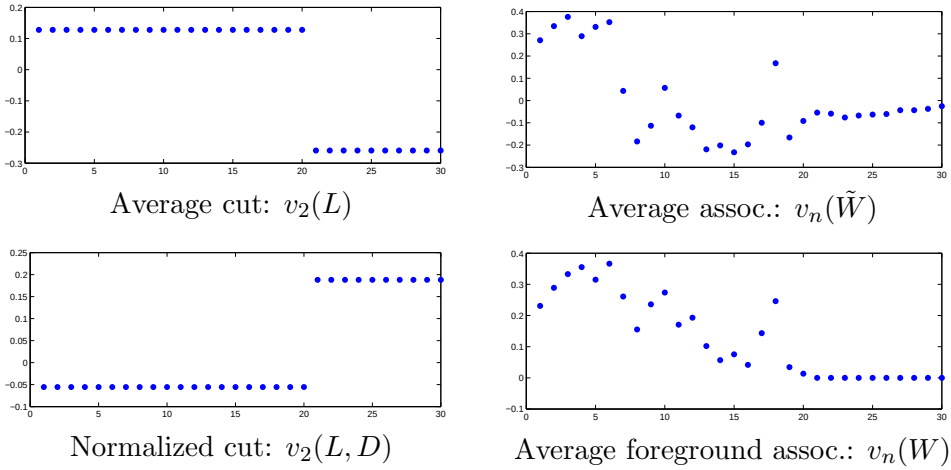


Figure 3.2: Eigenvectors used for clustering the point set from Figure 3.1, for similarities calculated with $\sigma = 0.1$. While the average and normalized cut relaxations result in eigenvectors that definitely yield the optimal clustering (left), the eigenvector corresponding to the largest eigenvalue of W does not give a clear result (bottom right). However, centering the similarity matrix completely fails in this case (top right), as only very close points get high similarity values.

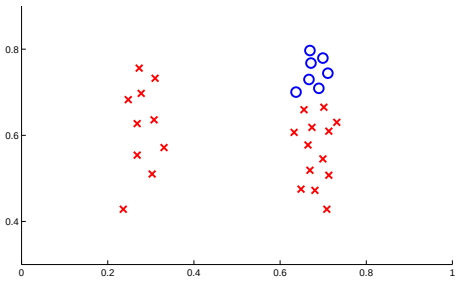


Figure 3.3: Segmentation obtained from the eigenvector $v_n(\tilde{W})$, given in Figure 3.2, top right. The average association criterion is not able to find the desired clusters.

vectors for the average and the normalized cut measures very clearly produce the desired segmentation given in Figure 3.1, right, the situation is more problematic for the average association measure: while the eigenvector corresponding to the largest eigenvalue of the original similarity matrix W does not give an obvious threshold value, the centered similarity matrix $\tilde{W}$ yields an eigenvector that is even worse. However, in this case the clustering obtained from $v_n(\tilde{W})$, which is depicted in Figure 3.3, really has a higher average association value ($f_{\text{Assoc}}(S) = 6.37$) than the segmentation from Figure 3.1 ($f_{\text{Assoc}}(S) = 5.87$) — apparently, the average association is not a suitable measure for the given clusters. Nevertheless, when we calculate the similarity matrix with $\sigma = 0.3$, all the eigenvectors produce the desired clustering (cf. Figure 3.4). This shows that the average (foreground) association criterion is more susceptible to different calculation of the similarity values than the cut criteria: if only very close

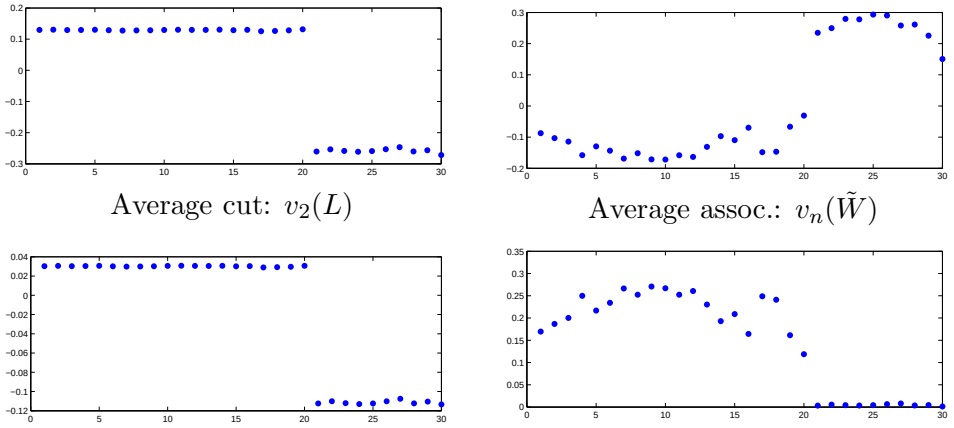


Figure 3.4: Eigenvectors used for clustering the point set from Figure 3.1, for similarities calculated with $\sigma = 0.3$. With this value for σ, the similarity values decrease, which results in all eigenvectors yielding the optimal clustering.

points have high similarity, it focuses on finding one small, very dense cluster as the associations between the other points are too small. We refer to [168] for further investigations on this topic of robustness to the similarity measure.

A different problem is given in Figure 3.5: this point set comprises a dense cluster in the middle and equally distributed background clutter, both containing 80 points. The similarity values are again computed based on normalized Euclidean distances, with $\sigma = 0.1$. For this example, only the eigenvector corresponding to the largest eigenvalue of W is able to separate the dense cluster from the background, which indicates that the average foreground association criterion is more appropriate in this case. The desired segmentation neither consists of two parts of similar inner association nor has a small cut value, and thus is not optimal for the other criteria. However, by resorting to a different distance measure (e.g. based on the point-density observed in the neighborhood of each point), this problem can possibly be solved.

Finally, Figure 3.6 depicts the results obtained for a small patch from a larger color image. In this case, the corresponding graph is locally connected by defining edges only between horizontally and vertically neighboring pixels. As distance measure $d(i, j)$ we compute the normalized color difference of two pixels in the perceptually uniform L*u*v* space, and use $\sigma = 0.3$ to derive the corresponding similarity values. As can be seen, both cut measures result in very similar segmentations, in which the hand is clearly separated from the ball. For this example, the likewise behavior of both measures may be attributed to the sparsity of the similarity matrix, which results in a degree vector d that is almost constant and hence in nearly equivalent optimization problems (cf. Section 3.1.1). As will be demonstrated in Section 4.4.3, however, the results may differ considerably if we revert to a dense similarity matrix. Note that for this example, the average association criterion once again separates a smaller patch of mostly consistent points from the rest of the image.

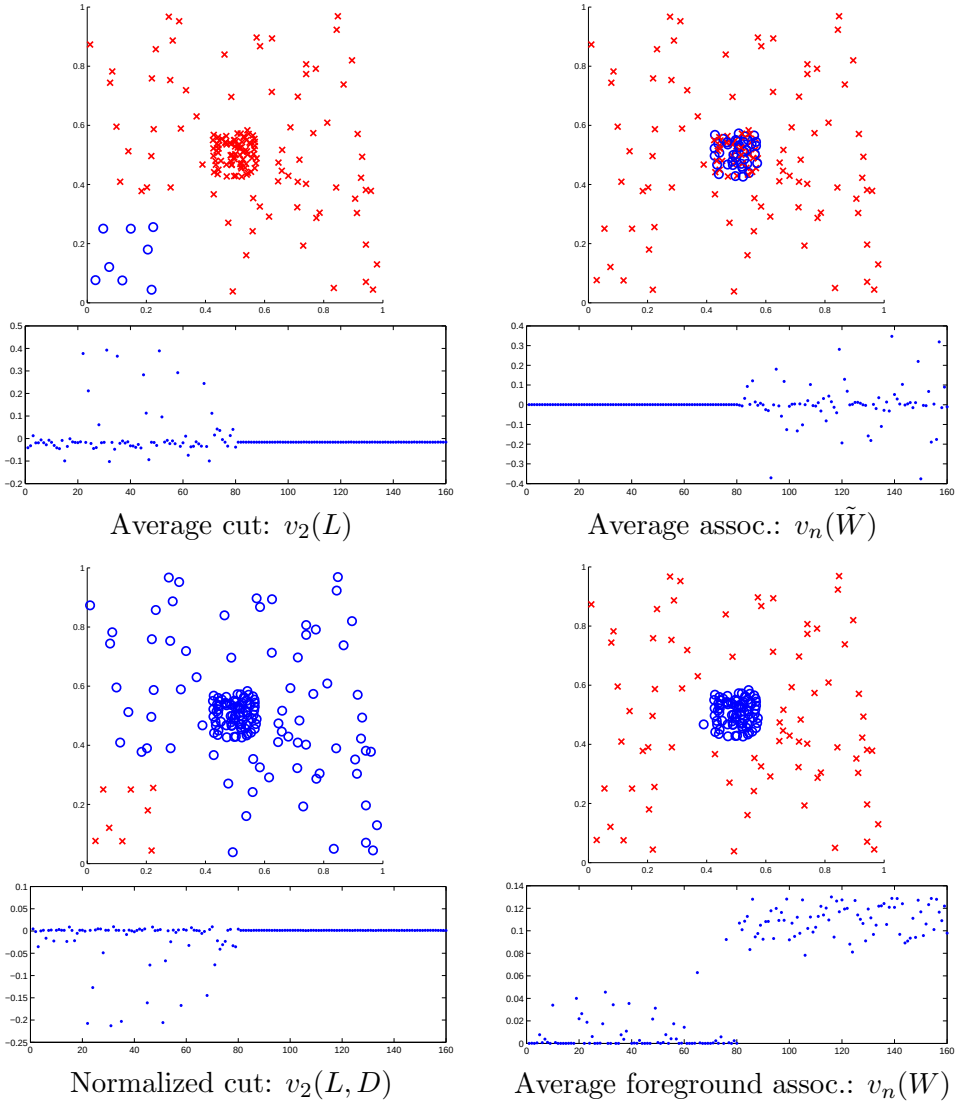


Figure 3.5: Eigenvectors and corresponding clusterings for a dot density problem. In this case, only the eigenvector corresponding to the largest eigenvalue of W yields the desired result (bottom right): the average foreground association criterion is more appropriate here, as it does not balance the parts of the segmentation.

As a first conclusion, these experiments indicate that in practice, the average and the normalized cut criteria may perform quite similarly. On the other hand, the average association is a much weaker measure of the perceptual impression of the scene, at least for the similarity values we used. Nevertheless, the example in Figure 3.4 demonstrates that reverting to similarity values of higher fluctuation (by using larger σ-values and/or smaller neighborhoods) may yield optimization problems where the average association measure is more appropriate. More experimental results for spectral partitioning techniques will be given throughout this thesis, when we compare them with our semidefinite relaxation approach.

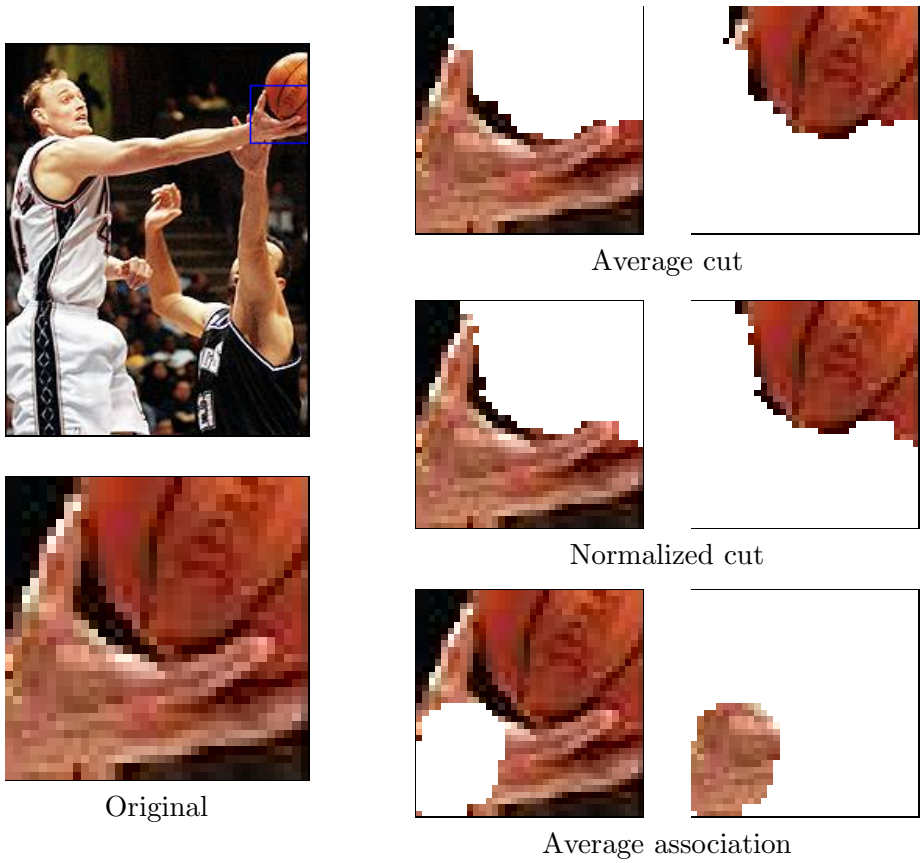


Figure 3.6: Segmentations for a color image of 36×36 pixels, obtained as a small patch of a larger image. While both cut measures yield satisfactory segmentations, the average association criterion does not give a valuable result.

3.2 Unsupervised Clustering in Euclidean Spaces: Mean Shift

For unsupervised partitioning tasks, clustering approaches that work directly in the feature space of the given image elements present an alternative to graph-theoretical approaches like the spectral techniques presented in Section 3.1. Representing each image element by a feature vector in Euclidean space, such methods form clusters efficiently by grouping together similar vectors based on (a variant of) a Euclidean distance measure. Hence, in contrast to graph-based methods, it is not necessary to compute the (dis-)similarity relations between all image elements in advance. This results in a clear advantage in terms of storage requirements and therefore allows dealing with much larger problem instances.

However, a common argument in favor of using (dis-)similarity relations is that they better capture signal variability in low-level vision [181]: in contrast to feature vector representations, they are not restricted to be calculated from Euclidean distances. Hence, (dis-)similarity relations may yield a more adequate description of the image in general situations. For this reason, they are

probably more suited for unsupervised segmentation, a claim that also appears to be supported by research on human perception [75].

Nevertheless, as an important representative method for unsupervised feature space partitioning we will briefly review the *mean shift technique* [33] in this section. In particular, since graph-based approaches are limited to smaller images due to the size of the similarity matrix, we will use this method later to reduce the number of variables for large images (cf. Sections 4.4.3 and 5.2).

Generally speaking, the mean shift technique is a nonparametric clustering approach based on density gradient estimation in a feature space that is equipped with a Euclidean metric. The associated procedure of iteratively seeking modes (i.e. local maxima) in the density distribution has already been developed by Fukunaga and Hostetler [59], but was only recently applied to unsupervised image segmentation [33]. The basic idea is to consider image pixels in the combined spatial-range domain, find the modes of the corresponding density distribution with the mean shift procedure and finally delineate the clusters associated with these modes by assigning the pixels appropriately.

In more detail, assuming that n points $x_i \in \mathbb{R}^d$ are given, the estimate of the density gradient for a point $x \in \mathbb{R}^d$ is obtained as the gradient of a multivariate kernel density estimation in x,

$$f(x) = \frac{1}{n\sigma^d} \sum_{i=1}^{n} K(\tfrac{x-x_i}{\sigma}) \ ,$$

by using the Epanechnikov kernel for K, which minimizes the asymptotic mean integrated square error of the density approximation [33]. This results in the density gradient estimate

$$\nabla f(x) = \frac{(d+2)|T_\sigma(x)|}{n\sigma^2 \operatorname{vol}(T_\sigma(x))} \frac{1}{|T_\sigma(x)|} \sum_{x_i \in T_\sigma(x)} (x_i - x) \ , \tag{3.26}$$

where $T_\sigma(x)$ denotes the hypersphere of radius σ around the center x, containing $|T_\sigma(x)|$ data points and having the volume $\operatorname{vol}(T_\sigma(x))$. Note that the only parameter needed in this estimation is the bandwidth $\sigma > 0$, which specifies the radius of the sphere the density is estimated in. Since in (3.26), the direction of the largest increase of the density is indicated by the *mean shift vector*

$$m_\sigma(x) = \frac{1}{|T_\sigma(x)|} \sum_{x_i \in T_\sigma(x)} (x_i - x) \ ,$$

a local density maximum (a *mode*) is found by iteratively following the path defined by the mean shift vectors. This leads to the *mean shift procedure* which computes a series of points

$$y^k = y^{k-1} + m_\sigma(y^{k-1}) = \frac{1}{|T_\sigma(y^{k-1})|} \sum_{x_i \in T_\sigma(y^{k-1})} x_i \quad \text{for } k = 1, 2, \dots \ , \tag{3.27}$$

and which is guaranteed to converge in a finite number of steps for discrete data sets [33].

Based on this procedure, the following algorithm computes a complete clustering of the given set of points $x_i, i = 1, \dots, n$:

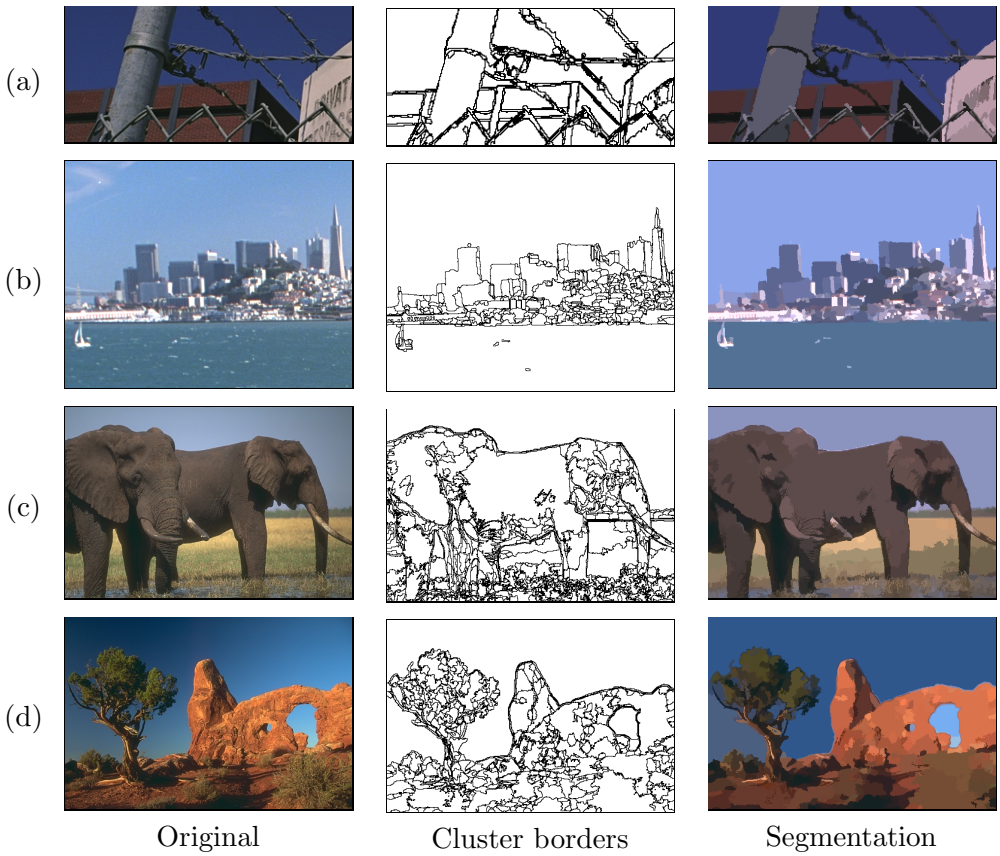


Figure 3.7: Clusterings obtained with mean shift. The bandwidth parameters and the minimum cluster size were adjusted to result in 200 – 500 clusters. The right column displays the resulting segmentations by replacing the color of each pixel with the average color of the corresponding cluster.

1. Apply the mean shift procedure (3.27) successively to each given point x_i by initializing $y_i^1 = x_i$, and find the corresponding convergence point z_i (which is a mode of the data distribution).

2. Delineate the clusters S_k, $k = 1, \ldots, m$, by grouping together all data points x_i which converged to similar modes: if for two points x_i and x_j the distance of the corresponding convergence points z_i and z_j in feature space is less than σ, $\|z_i - z_j\| < \sigma$, then both points are assigned to the same cluster S_k.

3. Optionally, eliminate small clusters containing less than M points by fusing them with the (spatially) closest larger cluster.

The final number m of clusters obtained with this algorithm besides by M is controlled by the bandwidth parameter σ: smaller values yield small hyperspheres as basis for the density estimation, which results in a higher number of clusters.

For image segmentation problems, the data points x_i are feature vectors comprising the position and the color in the perceptually uniform L*u*v* space

Image	Size	σ_s	σ_r	M	# Segments
(a)	298×141	4.0	8.0	20	211
(b)	512×404	5.0	8.0	25	404
(c)	481×321	5.0	3.5	50	460
(d)	481×321	5.0	9.0	50	366

Table 3.2: Size of the original image, parameter settings and number of segments obtained with the mean shift for the examples from Figure 3.7.

(or any other Euclidean feature data) of the corresponding pixel in the image. As each x_i thus originates from two different domains (the spatial and the range domain), its entries may vary considerably. Therefore, in practice two bandwidth parameters, σ_s (spatial domain) and σ_r (range domain), are used to scale the entries of the feature vectors appropriately. The mean shift procedure in the joint domain can then be executed very efficiently, since the search for the points in the hypersphere $T_\sigma(y^k)$ can be limited to a quadratic window of size $\sigma_s \times \sigma_s$ in the image.

Figure 3.7 shows some results obtained with the mean shift technique for diverse color images from the VisTex database [187] and the Berkeley segmentation dataset [121]. For these examples, we adjusted the bandwidth parameters σ_s and σ_r and the minimum cluster size M manually or semi-automatically, so that the final segmentation contains between 200 and 500 clusters. Since we will use the mean shift algorithm mainly as a preprocessing step, this number is adequate to obtain optimization problems of reasonable size (cf. Section 5.2). The semi-automatic adjustment affects the range bandwidth parameter σ_r, which is calculated by randomly picking a fixed number of pixels from the image, computing their maximum distance $d_{\max}$ in the L*u*v* color space, and setting σ_r to a certain fraction of $d_{\max}$. The parameter settings used for the images in Figure 3.7 can be found in Table 3.2. For these examples, the segmentations are then obtained in less than 10 seconds.

3.3 Supervised Segmentation with Markov Random Fields: Iterated Conditional Modes (ICM)

In Section 2.3, we stated that supervised segmentation problems can be approached by minimizing a global energy function $E(x)$ of the general form (2.10), which seeks a compromise between local agreement with the measured data and global smoothness of the labeling. Due to the integer constraint on x, however, this optimization problem is usually NP-hard. In this section, we will therefore briefly present Besag's ICM-algorithm [16] as an approximation method that greedily finds local minima of $E(x)$, and which has become quite popular because of its fast convergence properties and its general applicability.

Since the ICM-algorithm is basically motivated in the framework of probabilistic estimation based on Markov random fields [193, 114], we will start with explaining how the energy (2.10) arises in this context. To this end, the feature vectors g_i computed for the pixels i are assumed to depend on the 'true' image

x as realizations of some random variable Y, which is given by the conditional density function $f(g \mid x)$. Moreover, a *prior distribution* $P(x)$ on the space of images is assumed, which (independently of the given data) models prior expectations of the 'true' scene like smoothness constraints or regularity conditions. On this basis, the task of supervised segmentation can be defined as estimating the 'true' scene by seeking the labeling x that has maximum probability, given the observed feature vectors g_i. Using the Bayes theorem [193], this results in maximizing

$$\Pr(x \mid g) \propto f(g \mid x) P(x) , \tag{3.28}$$

which corresponds to finding the *maximum a posteriori (MAP)* estimate or the mode of the posterior distribution of the image.

If the prior distribution is represented in Gibbsian form

$$P(x) = \exp(-H(x)) , \tag{3.29}$$

with some real valued energy function $H(x)$, finding the MAP estimate is equivalent to minimizing the posterior energy function

$$E_{\mathrm{MAP}}(x \mid g) := -\ln f(g \mid x) + H(x) . \tag{3.30}$$

Besag [16] now makes two main assumptions:

1. The random variable Y is defined by conditionally independent random variables Y_i for each pixel i with identical density functions $p_{x_i}(g_i)$ that only depend on the label x_i. This results in the joint conditional density

$$f(g \mid x) = \prod_i p_{x_i}(g_i) . \tag{3.31}$$

   Modifications of this assumption for cases where it is not applicable are discussed in [16, 123].

2. The labeling x is a realization of a *locally dependent Markov random field*, i.e. the prior distribution $P(x)$ is defined through the local conditional probabilities

$$\Pr(x_i \mid x_{V \setminus \{i\}}) = P_i(x_i \mid x_{\delta i}) ,$$

   where $\delta i \subset V \setminus \{i\}$ denotes the set of *neighbors* of the pixel i. This means that the label of each point i only depends on the labels of its neighbors δi, and thus provides a way to formalize the idea that nearby pixels are likely to belong to the same class [123]. Most commonly used neighborhood systems in computer vision result from the symmetric lattice structure of an image: connecting each pixel to its four horizontal and vertical neighbors yields a first-order Markov random field, while additionally including the diagonally adjacent pixels defines a second-order field [16].

Furthermore, Besag [16] suggests to use a *pairwise interaction* Markov random field, which is described by (3.29) through the energy function

$$H(x) = \sum_i H_i(x_i) + \sum_{\langle i,j \rangle} H_{ij}(x_i, x_j)$$

with arbitrary functions H_i and H_{ij}. Based on this prior distribution of the image, and using the assumption (3.31), the posterior energy (3.30) becomes

$$E_{\mathrm{MAP}}(x \,|\, g) = -\sum_i \ln p_{x_i}(g_i) + \sum_i H_i(x_i) + \sum_{\langle i,j \rangle} H_{ij}(x_i, x_j) \; .$$

If we now define the point-wise conditional densities through the assignment costs $C'_i(x_i)$ as

$$p_{x_i}(g_i) \;\propto\; \exp\big(-C'_i(x_i)\big) \;\; ,$$

finding the MAP estimate becomes equivalent to minimizing the energy

$$E_{\mathrm{MAP}}(x \,|\, g) = \sum_i \big(C'_i(x_i) + H_i(x_i)\big) + \sum_{\langle i,j \rangle} H_{ij}(x_i, x_j) \; .$$

Obviously, this equals the energy functional (2.10) by setting $C_i(x_i) = C'_i(x_i) + H_i(x_i)$ and defining $H_{ij}(x_i, x_j) = P_{ij} D(x_i, x_j)$.

As the exact computation of the MAP estimate is generally intractable,[8] Besag [16] proposes the *iterated conditional modes (ICM)* algorithm, which seeks an approximative solution by individually maximizing the local conditional probabilities

$$\Pr(x_i \,|\, g_i, x_{V \setminus \{i\}}) \;\propto\; p_{x_i}(g_i) P_i(x_i \,|\, x_{\delta i}) \tag{3.32}$$

iteratively for each pixel i. As a first step, an initial estimate $\hat{x}$ of the classification needs to be calculated: ignoring the spatial relationships between the image pixels, this can be achieved by reverting to the maximum likelihood classifier, which simply chooses $\hat{x}_i$ to maximize $p_{x_i}(g_i)$ for each pixel i. Afterwards, in a single iteration the label $\hat{x}_i$ is updated for each pixel in turn by maximizing (3.32) based on the current labeling $\hat{x}_{\delta i}$ of its neighboring pixels. In terms of the energy functional (2.10), the ICM algorithm thus minimizes

$$E(x_i) = C_i(x_i) + \frac{1}{2} \sum_{j \in \delta i} P_{ij} D(x_i, \hat{x}_j) \tag{3.33}$$

with respect to x_i individually for each i. This procedure is applied until convergence or, in practice, for a predefined number of iterations to find the final classification x. In fact, convergence cannot be guaranteed for synchronous updating, i.e. when the new estimate $\hat{x}_i$ for each pixel is based on the labeling of the previous iteration, but only for serial updating [16].

Since the update of each pixel only requires the comparison of k different values (where k is the number of predefined classes), the ICM algorithm terminates very fast. In comparison to simulated annealing [64], which usually requires long computation times, ICM is equivalent to instantaneous freezing [16]. However, the final solution obviously merely corresponds to a *local* minimum of the posterior distribution of the image. As it critically depends on the

[8] One exception was already mentioned in Section 2.3: in case of binary labels and separation costs of Ising type (i.e. $H_{ij}(x_i, x_j) = P_{ij} x_i x_j$ with $x_i \in \{-1, +1\}$), the exact MAP solution can be computed by adopting the Ford-Fulkerson algorithm [70, 193].

initial estimate and the visiting scheme, the quality of this local minimum (in comparison to the MAP estimate as the global optimum) is difficult to analyze in general [193]. In fact, Besag [16] does not consider ICM as an approximation to the MAP estimator, but rather as an alternative estimator in its own right which has the advantage to ignore the large scale deficiencies of a Markov random field. Since for image restoration problems with low signal-to-noise ratio, the MAP estimator tends toward over-smoothing the image, this motivation is indeed justified; actually, it does not seem to be clear in which situations the MAP estimator is intrinsically desired for such problems (cf. [119]).

Several modifications of the ICM algorithm are proposed in [16, 123] which are useful in practice, like:

- The parameters P_{ij} in (3.33) indicating the strength of the pairwise relations can be increased during the iterations. This imposes a weaker random field for the first steps, which prevents that pixel labels are fixed too early.

- Instead of updating only one pixel in each step, the local conditional probabilities (3.32) can be maximized for small sets of pixels in parallel.

- Potentially unknown or uncertain parameter values in the assignment costs $C_i(x_i)$ could be (re-)estimated during the algorithm. This may be especially beneficial when only few training data is available initially.

- Instead of hard classification of each pixel, probabilistic labelings can be used during the iterations, which allow a pixel to have partial class membership. In that way, the influence of incorrectly labeled pixels becomes less critical.

Due to its low complexity, the ICM algorithm has found various applications, especially for segmentation tasks in the context of remote sensing (in which it also was originally proposed) [172, 34, 85, 88, 99]. For established classification methods in this field that are based on maximum likelihood estimation, it is a natural extension to incorporate spatial context. Moreover, it is capable to handle the enormous amounts of data emerging in remote sensing applications, and to compute corresponding segmentations in short time.

We finish this section by giving a few exemplary results for binary restoration problems as they were introduced in Section 2.3. Using a general notation that also includes multiclass restoration problems (cf. Section 6.1), we obtain for the local energy (3.33) that is iteratively minimized:

$$\begin{aligned} E(x_i) &= \|Ux_i - g_i\|^2 + \frac{1}{2}\lambda \sum_{j \in \delta i} \|x_i - \hat{x}_j\|^2 \\ &= \|Ux_i - g_i\|^2 - \lambda \hat{v}_{\delta i}(x_i) + const\ , \end{aligned} \tag{3.34}$$

where $U \in \mathbb{R}^{m \times k}$ contains the group prototypes as columns, $g_i \in \mathbb{R}^m$ is the measurement at point i, the indicator vectors $x_i \in \{e_1, \ldots, e_k\}$ take unit vectors from $\mathbb{R}^k$ as values, and $\hat{v}_{\delta i}(x_i)$ denotes the number of neighbors of i for which the

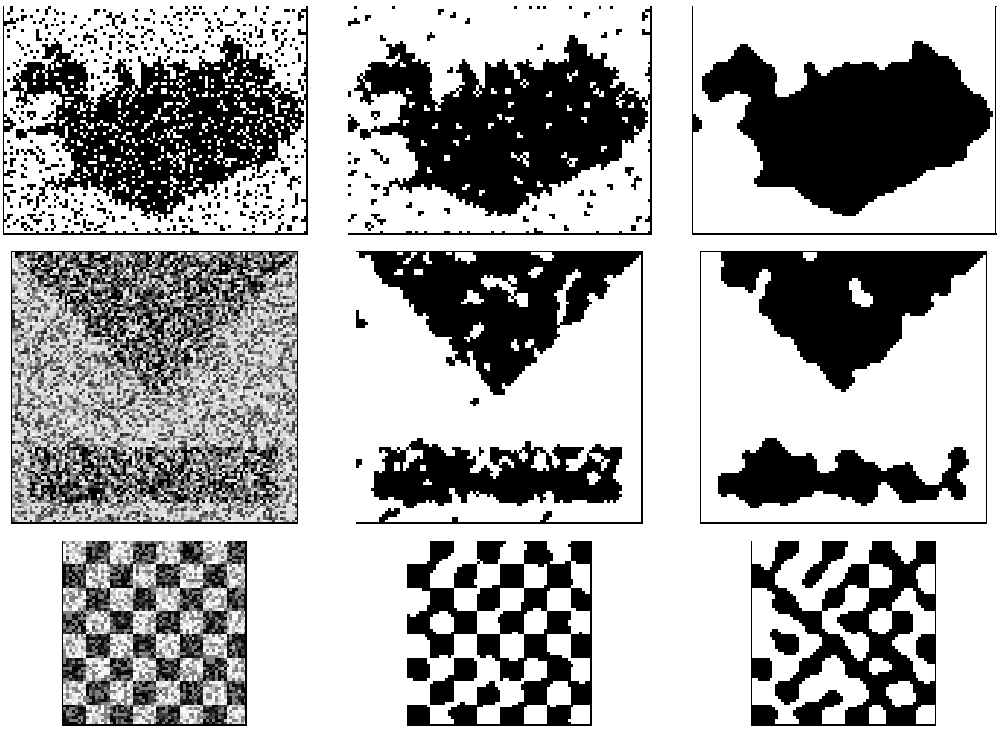


Figure 3.8: Restorations computed with the ICM algorithm. While the first image is difficult to restore due to its high noise level, better reconstructions are obtained for the other two images. The value of the parameter λ for which the best result is achieved depends on the spatial scales present in the image.

current label is x_i. Note that in this case, the probability distribution $P(x)$ of the corresponding Markov random field is equivalent to the Potts model, while the assignment costs can be interpreted as modeling Gaussian white noise with zero mean.

Figure 3.8 shows some noisy gray-value images and the corresponding restorations obtained with the ICM algorithm based on the energy function (3.34), for two different values for the smoothness parameter λ (the pixel values range between 0 and 1). The initial classification is calculated as the maximum likelihood estimate without spatial context, while in subsequent iterations, a second-order Markov random field is used. For the small images in Figure 3.8, the final segmentations were obtained after maximally 10 iterations in less than one second. The results reveal that ICM may find good reconstructions (first image with $\lambda = 0.6$), but also may fail to restore strongly degraded images in a smooth way (second image). In Section 4.4.5, we will see that a stronger relaxation of the MAP estimate will produce much better restorations.

Chapter 4

Semidefinite Relaxation of Binary Optimization Problems

One of the most important classes in mathematical optimization is given by *convex optimization problems* (e.g. [22]), which have the attractive property that due to the convexity of both the objective function and the feasible set, every local optimum is also a global one. This fact allows solving such problems reliably and efficiently, as there is no danger of getting trapped in undesirable local optima. Moreover, an elegant accompanying theory yields conceptual advantages that make convex optimization approaches convenient for many applications.

In this work, we pursue the concept of *convex relaxation* to deal with the combinatorial complexity of optimization problems. More specifically, this approach leads to *semidefinite programming (SDP) problems*, which are a special type of convex optimization problems. For this reason, we first give a general introduction to semidefinite programming in Section 4.1, which includes the main aspects of the corresponding duality theory, the geometry of the set of positive semidefinite matrices, and a brief overview of different methods that can be used to solve SDP problems.

In Section 4.2, we explain our SDP relaxation approach by performing Lagrangian relaxation, discuss geometry and feasibility issues of the obtained semidefinite program, present a randomized approximation method for obtaining a suboptimal solution of the original problem, and investigate the topic of performance bounds. The relation to spectral relaxation approaches is studied in Section 4.3. To this end, we provide a formulation of the SDP relaxation as an eigenvalue optimization problem, and compare it with spectral techniques. Finally, numerous application results are given in Section 4.4, including ground-truth experiments on binary restoration problems, and partitionings of real scenes for the different problem classes presented in Chapter 2. In this context, we also briefly discuss the issue of similarity measures and the computational complexity of the SDP relaxation method in practice.

4.1 Semidefinite Programming (SDP)

Semidefinite programs are a special type of convex optimization problems. In fact, SDP problems can be interpreted as a generalization of the established and thoroughly analyzed class of linear programming (LP) problems to the case of positive semidefinite matrix variables $X \succeq 0$. As we will see, most of the mathematically appealing properties of LP are inherited by SDP problems. With increasing computational capabilities, the interest in SDP is steadily growing. Besides approaching combinatorial optimization problems as in this work, applications to such diverse fields as signal processing [117], communication theory [178], or finance [68] have been presented recently. For a survey of more applications, we refer to [196, 40].

We consider the following standard formulation of a semidefinite program over symmetric matrix variables $X \in \mathcal{S}^n$:

$$\begin{aligned} f_p^* := \min_X \quad & C \bullet X \\ \text{s.t.} \quad & A_i \bullet X = b_i \quad \text{for } i = 1, \dots, m \\ & X \succeq 0 \,, \end{aligned} \tag{4.1}$$

with arbitrary symmetric matrices $C, A_i \in \mathcal{S}^n$, any vector $b = (b_1, \dots, b_m)^\top \in \mathbb{R}^m$, and the '$\bullet$'-operation denoting the standard matrix inner product $C \bullet X = \operatorname{Tr}(C^\top X) = \sum_{i,j} C_{ij} X_{ij}$. Note that the assumption of symmetry for C and the matrices A_i is no restriction: if e.g. C is not symmetric, we easily obtain an equivalent SDP instance by replacing it with $\frac{1}{2}(C^\top + C) \in \mathcal{S}^n$, since $C^\top \bullet X = C \bullet X$ [3]. Moreover, note that SDP problems with several matrix variables X_j can also be modeled in the standard form (4.1) by reverting to the matrix $\operatorname{Diag}(X_1, \dots, X_k)$ with the matrices X_j on its diagonal, since $X_j \succeq 0$ for $j = 1, \dots, k$ is equivalent to $\operatorname{Diag}(X_1, \dots, X_k) \succeq 0$ [78].

It is easy to see that SDP problems cover both linear programming and quadratic programming problems. For example, the linear program

$$\begin{aligned} \min_x \quad & c^\top x \\ \text{s.t.} \quad & a_i^\top x = b_i \quad \text{for } i = 1, \dots, m \\ & x \geq 0 \end{aligned}$$

is equivalent to the SDP problem (4.1) by setting $C = \operatorname{Diag}(c), A_i = \operatorname{Diag}(a_i)$ and restricting (4.1) to diagonal matrices $X = \operatorname{Diag}(x)$. The connection to quadratic programming can be established similarly (see [132, 78]).

4.1.1 Duality Theory

Analogous to linear programming, the (primal) SDP problem (4.1) is closely related to a corresponding dual problem, which can be obtained by a Lagrangian approach (e.g. [14]). To this end, we choose a vector of Lagrangian multipliers $y \in \mathbb{R}^m$ to lift the equality constraints of the primal problem (4.1) into the

objective function. This results in the following minimax-problem:

$$\begin{aligned} f_p^* &= \min_{X \succeq 0} \max_{y \in \mathbb{R}^m} C \bullet X + \sum_{i=1}^{m} y_i \left(b_i - A_i \bullet X\right) \\ &= \min_{X \succeq 0} \max_{y \in \mathbb{R}^m} \left(C - \sum_{i=1}^{m} y_i A_i\right) \bullet X + b^\top y \\ &\geq \max_{y \in \mathbb{R}^m} \min_{X \succeq 0} \left(C - \sum_{i=1}^{m} y_i A_i\right) \bullet X + b^\top y =: f_d^* \,. \end{aligned} \tag{4.2}$$

The interchange of min and max in the last step (4.2) corresponds to the so-called "minimax inequality" [152] and yields the dual of (4.1).[1] The inner minimization over $X \succeq 0$ in (4.2) now only becomes finite (which is implied by assuming that $f_p^* < \infty$) if the matrix $C - \sum_{i=1}^m y_i A_i$ is positive semidefinite, in which case $X = 0$ is optimal. Extracting this hidden semidefinite constraint, we obtain the following standard formulation of the *dual semidefinite program*:

$$\begin{aligned} f_d^* = \max_{y} \quad & b^\top y \\ \text{s.t.} \quad & C - \sum_{i=1}^{m} y_i A_i \succeq 0 \\ & y \in \mathbb{R}^m \,. \end{aligned} \tag{4.3}$$

Sometimes, the dual SDP is given in a slightly different way by introducing the slack variable $Z \in \mathcal{S}_+^n$ and replacing the first constraint in (4.3) by $\sum_{i=1}^m y_i A_i + Z = C$ and $Z \succeq 0$.

From the Lagrangian approach used above, we directly obtain the following *weak duality* property, which is identical to LP weak duality [199]:

Theorem 4.1 (Weak duality for SDP). *Let X and y denote feasible solutions of the primal SDP* (4.1) *and the dual SDP* (4.3)*, respectively. Then the gap between the solutions is*

$$C \bullet X - b^\top y = Z \bullet X \geq 0 \,. \tag{4.4}$$

However, unlike for LP, optimal solutions X^* and y^* may result in a nonzero *duality gap* (4.4) for SDP problems, as can be illustrated by simple examples provided e.g. in [183, 78, 179].

Nevertheless, a *strong duality* result for SDP can be derived if at least one of the problems (4.1) and (4.3) has a strictly interior point, which means that either a feasible, positive definite matrix $X \succ 0$ exists for (4.1), or a feasible y exists for (4.3) that yields a positive definite matrix $C - \sum_{i=1}^m y_i A_i \succ 0$. In general convex programming, this constraint is usually referred to as *Slater condition* [3, 14]. The corresponding theorem is provided e.g. in [3, 199]:

[1] In terms of linear operators, $\mathcal{A}^\top : \mathbb{R}^m \to \mathcal{S}^n$ with $\mathcal{A}^\top y := \sum_i y_i A_i$ is the adjoint operator of $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ with $\mathcal{A}X := (A_1 \bullet X, \ldots, A_m \bullet X)^\top$, as the corresponding inner products coincide: $(\mathcal{A}X)^\top y = X \bullet \mathcal{A}^\top y$ [78, 179].

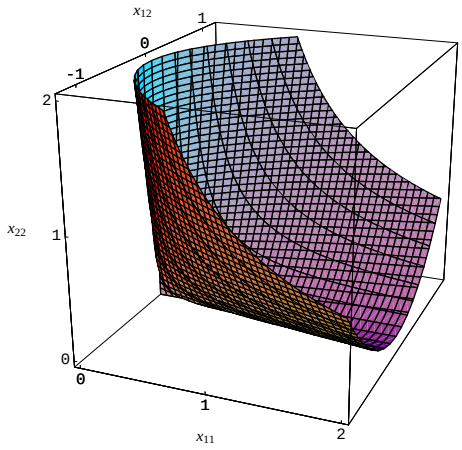


Figure 4.1: The cone of positive semidefinite matrices for $n = 2$ (the forth coordinate x_{21} is equal to x_{12}). In this case, the cone is rotational ("ice-cream cone") with vertex 0.

Theorem 4.2 (Strong duality for SDP). *If* (4.1) *and* (4.3) *both are feasible and there is a strictly interior point for the dual problem* (4.3) *(resp. for the primal problem* (4.1)*), then an optimal primal solution* X^* *(resp. an optimal dual solution* y^**) exists and the corresponding duality gap is zero:*

$$f_p^* - f_d^* = 0 \; .$$

If only one of the two problems is known to be feasible, then the existence of a strictly interior point in connection with a finite optimal value already guarantees the feasibility of the other problem and a zero duality gap. Yet, if one problem is unbounded, the other is automatically infeasible [199, 78]. More about the elegant duality theory for semidefinite programming can be found e.g. in [3, 196].

4.1.2 Geometry of SDP

The set of positive semidefinite matrices $\mathcal{S}_+^n$ over which is optimized in SDP is a special convex set, namely a *closed pointed cone*[2] in $\mathbb{R}^{\binom{n+1}{2}}$. Moreover, this cone is self-dual, i.e. it coincides with its dual (or polar) cone $(\mathcal{S}_+^n)^* = \{Y : X \bullet Y \geq 0,\ X \in \mathcal{S}_+^n\}$ [132]. Figure 4.1 illustrates the geometry of this pointed cone for the case $n = 2$.

The geometry of the semidefinite cone $\mathcal{S}_+^n$ has been studied extensively, especially in connection with semidefinite programming [110, 150, 139]. An important role in this context plays the facial structure of this cone. A *face* $F \subset \mathcal{S}_+^n$ is defined as a subset for which $X, Y \in F$ implies $Z = \alpha X + (1-\alpha)Y \in F$ for all $0 < \alpha < 1$. In general, the set of optimal solutions of an SDP problem always corresponds to a small face of the feasible set.

[2] A convex cone K is defined as a set that is closed under addition, $x + y \in K$ for $x, y \in K$, and multiplication with positive scalars, $cx \in K$ for $x \in K, c \geq 0$.

More specifically, consider the feasible set

$$K = \{X \in \mathcal{S}_+^n \mid A_i \bullet X = b_i \text{ for } i = 1, \ldots, m\}$$

of the SDP problem (4.1), which is called *spectrahedron* in [150]. As K is the intersection of an affine subspace with the semidefinite cone, the faces of K are given by the intersections of the faces of these two convex sets. Since for $A, B \in \mathcal{S}_+^n$, B is contained in the smallest face of $\mathcal{S}_+^n$ containing A if and only if $\ker(A) \subseteq \ker(B)$ [110], the minimal face containing an optimal solution X^* of the SDP problem (4.1) is given by

$$F_K(X^*) := \{X \in K \mid \ker(X^*) \subseteq \ker(X)\} .$$

The following lemma from [10, 139] shows that an optimal solution X^* of (4.1) is likely to have small rank:

Lemma 4.3. *(a) Let F be a face of dimension d of the feasible set K. Then $r = \operatorname{rank}(X)$ for $X \in F$ is bounded by*

$$\frac{1}{2}(r+1)r \leq m + d .$$

(b) If K contains extreme points (faces of dimension zero), then there exists an optimal solution X^* of (4.1) *with rank r^* for which*

$$\frac{1}{2}(r^*+1)r^* \leq m .$$

The second statement of this lemma immediately follows from the fact that the optimum of an SDP problem is always attained at an extreme point, since a linear function is minimized over a convex set.

For a detailed treatment of the geometry of SDP problems we refer to [196, Chapter 3].

4.1.3 SDP Solvers

To compute optimal primal and dual solutions X^*, y^* for the SDP problems (4.1) and (4.3), respectively, a wide range of different SDP algorithms can be used. In fact, the development of such solvers currently is one of the most active areas of research in optimization, and the number of reliable software to handle SDP is steadily growing. However, the underlying methods of these SDP algorithms are rather sophisticated, and it is beyond the scope of this work to present the details. Therefore, we will only briefly summarize the main ideas here; more information on this topic can be found in several books [132, 199, 196].

In general, each SDP can be solved as convex minimization problem *almost* exactly in polynomial time [132]. More precisely, an ϵ-approximation to the global optimum can be numerically determined for any fixed precision ϵ, e.g. by using the ellipsoid method [71]. However, since the running time of this method is prohibitively high in practice, other more efficient algorithms are usually used.

Most SDP solvers are based on iterative *interior point methods*, which originally have been developed for LP. The basic idea of such methods is to do a line search through the interior of the feasible set which converges to the solution. To this end, a weighted barrier term is added to the objective function which prevents the algorithm from leaving the interior of the feasible set. To be able to reach an optimal solution (which is usually located on the boundary), the weight of the barrier term, represented by a parameter $\mu > 0$, is successively reduced. For this modified objective, a corresponding sequence of minimizers $\{X_\mu, y_\mu\}$ depending on the parameter μ is computed, until the duality gap falls below some given threshold ϵ. This sequence defines a smooth curve, called the *central path*, which is guaranteed to converge to the global optimum. Typically, it is sufficient to approximate the minimizers $\{X_\mu, y_\mu\}$ by applying a few Newton steps. A remarkable result in [132] asserts that for the family of self-concordant[3] barrier functions, such methods converge in polynomial time, with the complexity depending on the number of variables n and the value of ϵ.

Various variants of interior point algorithms have been developed in recent years, like pure primal or dual methods [132, 3], combined primal-dual methods [80, 130, 133], or potential-reduction methods [183, 12]. Moreover, there have been major efforts to exploit the special structure of some SDP problems, especially in the context of combinatorial optimization problems for which extreme sparsity is often encountered [57, 12, 194]. These approaches are able to solve large-scale problems with up to 10,000 variables efficiently [11].

Besides interior point algorithms, other methods have been proposed for SDP which are especially suited for large-scale problems. One example is given by the *spectral bundle method* of Helmberg and Rendl [79], which is applicable for problems that can be cast as an eigenvalue optimization problem. At the cost of a poor convergence rate, it is dedicated for problems with a large number of constraints. Another example is the nonlinear programming approach of Burer and Monteiro [27], which tries to reduce the number of variables by exploring the property given in Lemma 4.3 via low-rank factorization of the primal solution matrix X. Similarly, Kočvara and Stingl [108] use an augmented Lagrangian method as the framework of a nonlinear programming approach to solve large-scale SDP problems.

Finally, we note that several of the mentioned SDP solvers were evaluated at the Seventh DIMACS Implementation Challenge on Semidefinite and Related Optimization Problems in 2000 [46]. A discussion and comparison of the participating algorithms based on the corresponding benchmark results is presented in [126].

4.2 Optimization via Semidefinite Relaxation

The rising interest in semidefinite programming in recent years has been initiated mainly by the fact that many combinatorial optimization problems can be

[3] A function is called self-concordant, if it is three times continuously differentiable, and satisfies a certain inequality constraint (see [132] for more details). This includes, e.g., all linear and quadratic functions.

solved approximately by applying a semidefinite relaxation approach. The basic ideas were introduced in the seminal work of Lovász and Schrijver [116], who derived bounds on combinatorial problems by constructing semidefinite relaxations based on lifting the problem variables into a higher-dimensional space. Goemans and Williamson [66] later decisively extended this idea by developing a randomized approximation algorithm with a strong performance guarantee for the max-cut problem. Since then, many applications of semidefinite relaxation in combinatorial optimization have been presented in the literature. Several surveys give an overview of the recent developments [3, 65, 152, 78, 195, 115, 112].

In this section, we introduce a semidefinite relaxation approach which can be used to approximately solve binary optimization problems from computer vision like those presented in Chapter 2. In this context, note again that all these partitioning problems can be cast as minimization problems of the general form (2.1) involving a quadratic objective function and binary decision variables. The missing linear constraint for the perceptual grouping problem (2.9) and the restoration problem (2.12) can easily be modeled by setting $c = 0$ and $\beta = 0$.

As a first step, we homogenize the objective function of (2.1) by increasing the dimension by one to obtain a purely quadratic functional:

$$x^\top Q x + 2d^\top x + const = \begin{pmatrix} x_0 \\ x \end{pmatrix}^\top L \begin{pmatrix} x_0 \\ x \end{pmatrix} ,$$
$$\text{with} \quad L = \begin{pmatrix} const & d^\top \\ d & Q \end{pmatrix} \text{ and } x_0 = 1 .$$

Since this is only necessary for the perceptual grouping and the restoration problems (which lack the linear constraint), the sign of x_0 does not influence the minimization of this functional: if $x_0 = -1$ for the optimal solution, we can simply replace x with $-x$ to obtain an equivalent solution that satisfies $x_0 = 1$. Therefore, we may require without loss of generality that $x' = (x_0, x)^\top \in \{-1, +1\}^{n+1}$, which leads to a problem of dimension $n + 1$ that is equivalent to (2.1).[4] Hence, with slight abuse of notation, we will consider the following homogeneous combinatorial optimization problem in this section:

$$\begin{aligned} z^* := \min_x \quad & x^\top L x \\ \text{s.t.} \quad & x \in \{-1, +1\}^n \\ & c^\top x = \beta , \end{aligned} \tag{4.5}$$

where we will only assume that L is symmetric (note that this is more general than the unsupervised partitioning problem (2.6), where L is required to be a Laplacian matrix). The findings then apply to all three problems presented in Chapter 2, unless it is stated otherwise by a special choice of the constraint variables c and β. Since (4.5) can also be interpreted as seeking a constrained minimum cut in a graph with (possibly negative) edge-weights defined by L,

[4] For problems of type (2.1) with both $d \neq 0$ and $c \neq 0$, the homogenization is only valid if the linear constraint is replaced by $|c^\top x| = |\beta|$, since switching the sign of the solution x may result in $c^\top x = -\beta$.

this confirms the observation from Sections 2.2 and 2.3 that both problems can be translated into graph cut problems.

The combinatorial optimization problem (4.5) is now solved approximately in two steps: first, the decision variables are lifted into a higher-dimensional matrix space, where the corresponding problem is relaxed to a semidefinite programming problem by *weakly* incorporating the combinatorial constraints on the variables (Section 4.2.1). This SDP problem can then be solved with any of the techniques presented in Section 4.1.3. In the second step, an integer solution is obtained from the relaxed solution by applying a randomized rounding procedure (Section 4.2.3), which was first suggested in [66]. Moreover, we also investigate the geometry and feasibility of the SDP relaxation (Section 4.2.2), and provide bounds on the quality of the solutions (Section 4.2.4).

4.2.1 Lagrangian Relaxation

In order to relax the discrete optimization problem (4.5), we pursue *Lagrangian relaxation* [113]. This fundamental technique not only allows some insights into the relaxation process itself, but also yields strong bounds on the optimal solution via Lagrangian duality, and is therefore often considered as "best" relaxation approach [195]. Moreover, we will see that the resulting convex optimization problem corresponds to a direct semidefinite relaxation of (4.5).

As a first step, we express the integer constraints on the entries of x by $x_i^2 = 1$, $i = 1, \dots, n$, and replace the linear constraint with the quadratic constraint $(c^\top x)^2 = \beta^2$. Note that this does not change the optimization problem: if the squared constraint results in a solution x with $c^\top x = -\beta$, it can simply be replaced by $-x$, which yields the same objective value and satisfies the original linear constraint. Denoting the Lagrangian multiplier variables with y_i, $i = 0, \dots, n$, the Lagrangian of (4.5) reads:

$$\begin{aligned} & x^\top L x - y_0 \left((c^\top x)^2 - \beta^2 \right) - \sum_{i=1}^{n} y_i (x_i^2 - 1) \\ = \; & x^\top \left(L - y_0 c c^\top - \operatorname{Diag}(y) \right) x + \beta^2 y_0 + e^\top y \; . \end{aligned}$$

Several other methods are possible to model a linear constraint before it is incorporated into the Lagrangian [81, 112]; however, using a squared representation simplifies the following analysis and is advantageous from the theoretical [145, 195] and the computational [81] point of view.

In terms of the corresponding minimax-problem, the Lagrangian relaxation of (4.5) now becomes (cf. [113] and eq. (4.2)):

$$z^* \geq \max_{y_0, y} \min_{x} x^\top \left(L - y_0 c c^\top - \operatorname{Diag}(y) \right) x + \beta^2 y_0 + e^\top y \; .$$

Since x is unconstrained now, the inner minimization is finite-valued if and only if $L - y_0 c c^\top - D(y)$ is positive semidefinite, in which case $x = 0$ is optimal. Using this *hidden semidefinite constraint*, we arrive at the relaxed problem:

$$\begin{aligned} s_d^* := \max_{y_0, y} \quad & \beta^2 y_0 + e^\top y \\ \text{s.t.} \quad & L - y_0 c c^\top - \operatorname{Diag}(y) \succeq 0 \; . \end{aligned} \tag{4.6}$$

The important point here is that (4.6) is a semidefinite program! In comparison to the general dual SDP formulation (4.3), we have $m = n + 1$, $b = (\beta^2, e)^\top$, $C = L$, and the constraint matrices $A_0 = cc^\top$ and $A_i = e_i e_i^\top$, $i = 1, \ldots, n$, where $e_i \in \mathbb{R}^n$ denotes the i-th unit vector. Hence, the corresponding primal semidefinite program can directly be obtained from the general SDP formulation (4.1):

$$\begin{aligned} s_p^* := \min_{X \succeq 0} \quad & L \bullet X \\ \text{s.t.} \quad & cc^\top \bullet X = \beta^2 \\ & \operatorname{diag}(X) = e \,. \end{aligned} \tag{4.7}$$

This final relaxation (4.7) can also be derived as a direct semidefinite relaxation of the original problem (4.5). To see this, we first rewrite the objective function of (4.5) as $x^\top L x = \operatorname{Tr}(Lxx^\top) = L \bullet xx^\top$ by exploiting the commutativity of the trace. Using the quadratic representation of the constraints, this yields the following problem formulation, which is equivalent to (4.5):

$$\begin{aligned} z^* := \min_{x} \quad & L \bullet xx^\top \\ \text{s.t.} \quad & cc^\top \bullet xx^\top = \beta^2 \\ & \operatorname{diag}(xx^\top) = e \,. \end{aligned}$$

Note that the matrix $xx^\top$ is positive semidefinite and has *rank one*. Thus the relaxation (4.7) consists in replacing $xx^\top$ by an arbitrary positive semidefinite matrix $X \in \mathcal{S}_+^n$, i.e. dropping the rank one condition. In fact, if we add the constraint $\operatorname{rank}(X) = 1$ to (4.7), the problem becomes equivalent to the original problem (4.5) [110, 78]. This already indicates the strength of the SDP relaxation: by lifting the problem into the higher-dimensional space $\mathcal{S}_+^n$, the integer constraint on the entries of x can be taken into account appropriately, and "only" a rank-constraint needs to be relaxed. This fundamental quality of the lifting procedure to allow a simpler representation of intricate constraints in higher-dimensional spaces is also a well-known fact in other fields like pattern recognition and statistical learning [35, 184, 43].

The Lagrangian relaxation approach provides the opportunity to derive even stronger relaxations: adding redundant constraints to the original problem formulation (4.5) may result in non-redundant conditions in the corresponding Lagrangian relaxation, which may lead to tighter bounds on the objective value. One important example of such redundant constraints is given by the well-known *triangle inequalities* which define the so-called metric polytope in $\mathcal{S}^n$ (cf. [80]):

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} X_{ij} \\ X_{jk} \\ X_{ik} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \geq 0 \,,$$

for all $1 \leq i < j < k \leq n$. These inequalities model the trivial observation that for each triple i, j, k of points either two or none of the connecting edges in the corresponding graph can be cut, which translates to the fact that not all pairwise

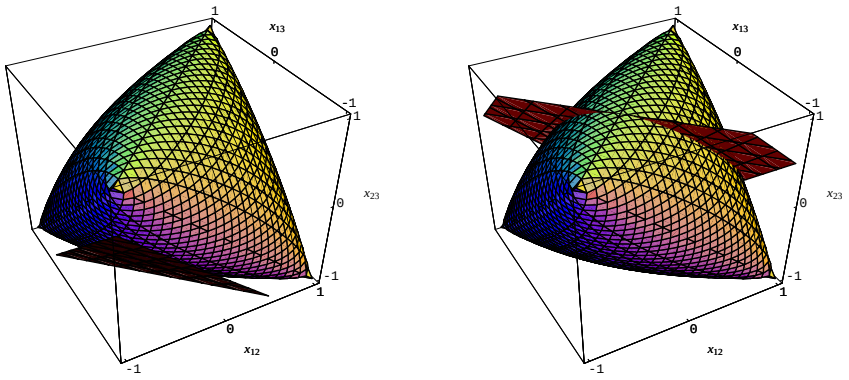


Figure 4.2: The elliptope $\mathcal{E}_3$ containing the feasible solutions for the convex problem relaxation (4.7) for $n = 3$. The subset of feasible *combinatorial* solutions only consists of the four vertices. To take into account the linear constraint, this set additionally has to be intersected with the hyperplane $cc^\top \bullet X = \beta^2$. This is illustrated for $c = e$ with $\beta = 0$ (left), yielding a tangential hyperplane, and for $\beta = 2$ (right), respectively.

products $X_{ij} = x_i x_j, X_{jk} = x_j x_k$, and $X_{ik} = x_i x_k$ can be -1 simultaneously. Including all triangle inequalities yields $4\binom{n}{3}$ additional constraints.

The inclusion of other redundant constraints has been suggested, like the more general clique inequalities [82], or quadratic constraints on the matrix entries based on the Hadamard product [195]. However, although adding such constraints yields better relaxation bounds, it has been shown that for instance for the max-cut problem, the performance of the randomized rounding technique (cf. Section 4.2.3) which gives the combinatorial solution does not improve [97]. Since preliminary results from practice support this statement for our problem formulation (4.5), and as such constraints do not fit exactly into the SDP framework presented in Section 4.1, we will not further consider them here.

4.2.2 Geometry and Feasibility

In order to illustrate how the semidefinite relaxation (4.7) approximates the *combinatorial, non-convex* problem (4.5), let us consider the case $n = 3$. The intersection of the convex set $\mathcal{S}^n_+$ with the hyperplanes defined by $\operatorname{diag}(X) = e$ yields the convex set $\mathcal{E}_n := \{X \in \mathcal{S}^n_+ \mid \operatorname{diag}(X) = e\}$, which is referred to as the set of *correlation matrices* [72] or the *elliptope* [110]. The structure of the elliptope has been studied extensively, see e.g. [44].

For $n = 3$, a matrix $X \in \mathcal{E}_3$ has three unknowns due to symmetry, corresponding to the upper (or lower) triangular part. The corresponding elliptope $\mathcal{E}_3$ (or rather its 3-dimensional projection) is shown in Figure 4.2. It looks like a polytope with four vertices (which correspond to the combinatorial solutions of the unrelaxed problem) but with non-linear faces. The set of feasible solutions for (4.7) is now obtained by additionally intersecting this set with the hyperplane $cc^\top \bullet X = \beta^2$ (see Figure 4.2), and thus may reduce to a single point on the surface of the elliptope (e.g. in the case $c = e$ and $\beta = 0$). This

also shows that the original combinatorial problem (4.5) only has a feasible solution if at least one of the vertices of the elliptope $\mathcal{E}_n$ lies on this hyperplane. Nevertheless, as long as the hyperplane hits the elliptope, the solution X^* of the *relaxed* problem can always be determined by numerically minimizing the linear functional $L \bullet X$ over the feasible set. The nearest vertex to X^* then corresponds to a suboptimal solution of (4.5), or at least to a combinatorial solution which closely approximates the linear constraint $c^\top x = \beta$.

This simple example already indicates that it is possible that no feasible solution for the primal SDP relaxation (4.7) exists if the constraint variables c or β are chosen inappropriately. Another example illustrates this situation more clearly: for the case $n = 2$, the constraints in (4.7) yield $X = \left(\begin{smallmatrix} 1 & x_{12} \\ x_{12} & 1 \end{smallmatrix}\right)$ with $x_{12} = \frac{\beta^2 - c_1^2 - c_2^2}{2c_1c_2}$ (assuming that both c_1 and c_2 are positive). Additionally, since X is positive semidefinite, $-1 \leq x_{12} \leq 1$ has to hold, which results in the condition $(c_1 - c_2)^2 \leq \beta^2 \leq (c_1 + c_2)^2$. For example for $\beta = 0$, this is only valid for $c_1 = c_2$; all other choices of c make the primal SDP problem (4.7) infeasible.

Fortunately, it is possible to exactly characterize the situations when the primal problem (4.7) has a feasible solution. The following result is mainly based on a theorem given in [41]:

Theorem 4.4. *The primal SDP problem* (4.7) *is feasible for* $c \in \mathbb{R}^n, \beta \in \mathbb{R}$ *if and only if the vector* $\tilde{c} := \left(\begin{smallmatrix} c \\ \beta \end{smallmatrix}\right)$ *is balanced, i.e.*

$$|\tilde{c}_i| \leq \sum_{j \neq i} |\tilde{c}_j| \quad \textit{for all } i = 1, \ldots, n+1 \; .$$

Proof. Consider the following result from [41], which is stated in a more convenient way in [111]: for a given vector $\tilde{c} \in \mathbb{R}^{n+1}$, there exists a matrix $\tilde{X} \in \mathcal{E}_{n+1}$ with $\tilde{X}\tilde{c} = 0$ if and only if $\tilde{c}$ is balanced. Using this result, it remains to show that the primal SDP problem (4.7) is feasible if and only if a matrix $\tilde{X} \in \mathcal{E}_{n+1}$ with $\tilde{X}\tilde{c} = 0$ exists.

The proof is based on the decomposition

$$\tilde{X} = \begin{pmatrix} X & z \\ z^\top & 1 \end{pmatrix} \tag{4.8}$$

with $z \in \mathbb{R}^n$, which directly gives the equivalence of $\operatorname{diag}(X) = e$ and $\operatorname{diag}(\tilde{X}) = e$. Substituting (4.8) into $\tilde{X}\tilde{c} = 0$ yields

$$\begin{aligned} Xc &= -\beta z \\ z^\top c &= -\beta \; . \end{aligned}$$

Multiplication of the first equation with $c^\top$ results in $c^\top Xc = -\beta c^\top z = \beta^2$ via the second equation. Thus requiring $\tilde{X}\tilde{c} = 0$ is equivalent to the constraint $cc^\top \bullet X = c^\top Xc = \beta^2$ in the primal problem (4.7) in combination with choosing z to satisfy $Xc = -\beta z$.

Finally, it remains to prove the equivalence of $\tilde{X} \succeq 0$ and $X \succeq 0$. To this end, we use the Schur complement (see Theorem A.8) of $\tilde{X}$, which gives $\tilde{X} \succeq 0 \;\Leftrightarrow\; X - zz^\top \succeq 0$. We immediately see that the last statement yields

$X \succeq zz^\top \succeq 0$, since the rank one matrix $zz^\top$ is always positive semidefinite (see Lemma A.9). For the other implication, observe that for each $v \in \mathbb{R}^n$

$$\begin{aligned} v^\top(X - zz^\top)v &= v^\top Xv - (v^\top z)^2 \\ &= v^\top Xv - \tfrac{1}{\beta^2}(v^\top Xc)^2 \end{aligned}$$

by assuming $\beta \neq 0$ and substituting $z = -\frac{1}{\beta}Xc$ (see above). As $X \succeq 0$ (assumption), we can apply the general Cauchy-Schwarz inequality $|v^\top Xc|^2 \leq (v^\top Xv)(c^\top Xc)$ (see Theorem A.10) to obtain

$$\begin{aligned} v^\top(X - zz^\top)v &\geq v^\top Xv - \tfrac{1}{\beta^2}(v^\top Xv)(c^\top Xc) \\ &= v^\top Xv - \tfrac{1}{\beta^2}(v^\top Xv)\beta^2 = 0 \,, \end{aligned}$$

which shows the desired positive semidefiniteness of $X - zz^\top$.

For $\beta = 0$, setting $z = 0$ yields the equivalences directly. □

Due to this result, we will only investigate examples in the following sections where the combination of the constraint variables c and β is balanced as stated in Theorem 4.4. In this case, the corresponding semidefinite relaxation is known to be "well-behaved" according to the strong duality Theorem 4.2, since besides the guaranteed feasibility of the primal problem (4.7), a strictly interior point for the dual problem (4.6) can always be found by setting $y_0 = 0$ and $y = -\alpha e$ with α large enough. Hence, an optimal primal solution X^* of the SDP relaxation (4.7) exists which yields no duality gap:

$$s_p^* - s_d^* = L \bullet X^* - s_d^* = 0 \,.$$

In contrast to this, note that the optimum for the dual SDP problem (4.6) may not be attained (i.e. no feasible $\binom{y_0}{y}$ for (4.6) yields the optimal objective value s_d^*) if no strictly interior point for the primal problem exists — a case that occurs for instance for $\beta = 0, c \neq 0$, since this requires the smallest eigenvalue of X to be zero. However, this is not relevant for the optimization problems considered in this thesis, since we only need the optimal *primal* solution X^* to approximate the combinatorial solution of (4.5).

To compute this optimal solution X^*, any of the SDP algorithms sketched in Section 4.1.3 can be used. Yet, as the primal problem may have no strictly interior point in some cases, interior point methods based on the primal problem are often ineligible. Due to this observation, we resorted to SDP solvers in practice which either do not use interior point methods [108], or which are based only on the dual problem, like the dual-scaling algorithm of Benson et al. [12]. The latter algorithm also is especially suited for large-scale problems, and capable to exploit the sparsity structure which is encountered in some of the relaxations of the combinatorial problems investigated here.

4.2.3 Randomized Approximation

Once the solution matrix X^* of the primal SDP problem (4.7) has been computed, we wish to find a corresponding combinatorial solution x to the original

problem (4.5). For this purpose, we use the *randomized hyperplane technique* proposed by Goemans and Williamson [66] for the max-cut problem. To describe this method, the following geometric interpretation of the SDP relaxation (4.7) is more convenient:

Since each feasible matrix X is positive semidefinite, it can be factorized into $X = VV^\top$ with $V \in \mathbb{R}^{n\times n}$, using e.g. the Cholesky decomposition (cf. Theorem A.6). If we write $V = (v_1, \ldots, v_n)^\top$, i.e. the rows of V are identified by v_i, the constraint $\operatorname{diag}(X) = e$ in (4.7) corresponds to $\|v_i\|^2 = v_i^\top v_i = 1$ for all $i = 1, \ldots, n$. Moreover, the first constraint can be rewritten as $\beta^2 = cc^\top \bullet (VV^\top) = \operatorname{Tr}(c^\top VV^\top c) = \|V^\top c\|^2$. Exploiting the permutation property for the arguments of the trace also for the objective function, the primal SDP relaxation (4.7) is equivalent to the problem

$$\begin{aligned} \min_{V\in\mathbb{R}^{n\times n}} \quad & \operatorname{Tr}(V^\top LV) \\ \text{s.t.} \quad & \|V^\top c\|^2 = \beta^2 \\ & \|v_i\|^2 = 1 \quad \text{for } i = 1, \ldots, n \,, \end{aligned} \tag{4.9}$$

as the matrix $VV^\top$ is always positive semidefinite.

In comparison to the original combinatorial problem (4.5), the relaxation step may thus be interpreted as associating the binary variables $x_i \in \{-1, +1\}$ with vectors $v_i \in \mathbb{R}^n$ from the unit sphere in a high-dimensional space. Accordingly, each matrix entry $(xx^\top)_{ij} = x_i x_j$ is replaced by the matrix entry $X_{ij} = v_i^\top v_j$ which represents the cosine of the angle between the vectors v_i and v_j. Minimizing the objective function then corresponds to finding an embedding of the points on the unit sphere such that for large positive matrix entries L_{ij} the corresponding points $i \neq j$ are placed far apart. Using this interpretation, it is easy to see that the above problem becomes equivalent to the original combinatorial problem (4.5) if we furthermore demand that $|v_i^\top v_j| = 1$ for all $i, j = 1, \ldots, n$ [195].

Based on the factorization $X^* = VV^\top$ of the optimal solution of the SDP relaxation (4.7), the idea of the randomized hyperplane algorithm is straightforward: randomly select a hyperplane through the origin, and group all points v_i on the same side of this hyperplane together, thus partitioning the points into two sets. In mathematical terms, this is achieved by representing the hyperplane by its normal vector r from the unit sphere, $\|r\| = 1$, and setting each entry of the binary vector x to

$$x_i = \operatorname{sgn}(v_i^\top r) = \begin{cases} +1 \text{ for } v_i^\top r \geq 0 \\ -1 \text{ for } v_i^\top r < 0 \end{cases} .$$

For the max-cut problem with positive edge weights, Goemans and Williamson [66] prove that this randomized hyperplane technique yields a high approximation ratio (in terms of the expected objective value). In the next section, we will show how their result can be extended to yield performance bounds for the more general problem formulation (4.5). To obtain good results in practice, the randomized rounding step is repeated several times for different random

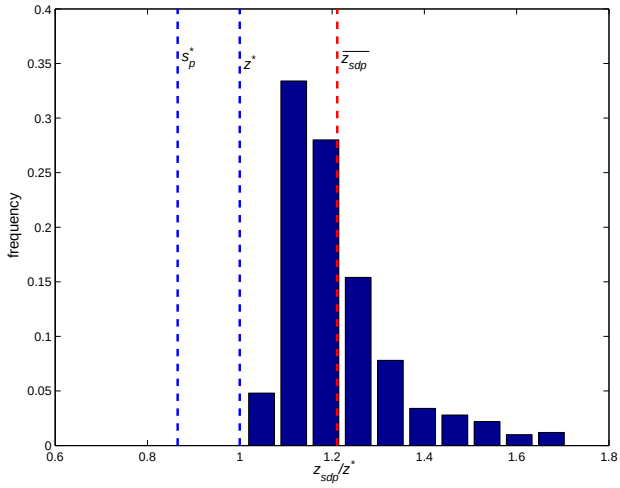


Figure 4.3: Performance of the randomized hyperplane technique. For a reconstruction problem considered in Section 4.4.1, the solution of the SDP relaxation is rounded to an integer solution with 500 different hyperplanes. The histogram comprises the corresponding objective values: while the mean objective value $\overline{z_{sdp}}$ has a relative error about 21% in comparison to the optimal solution z^* of the combinatorial problem, the best integer solution found in this way results in a relative error of only 1.2%. The objective value of the SDP relaxation yields a lower bound s_p^* of about 86.5% of the optimal objective value.

vectors r. The final binary solution x_{sdp} is then selected as the one that yields the minimum value for the original objective function $x^\top Lx$.

An alternative rounding procedure which yields the same result in terms of its expected objective value is given by Bertsimas and Ye [15]: they interpret the solution X^* as covariance matrix of a multivariate normal distribution $N(0, X^*)$ with mean 0. Generating a vector z from this distribution, a binary vector x is then obtained by directly setting $x_i = \operatorname{sgn}(z_i)$ for each i.

Note that since these randomized rounding techniques do not take into account the linear constraint $c^\top x = \beta$, the resulting binary solution x_{sdp} is not necessarily feasible for the original problem (4.5) — this is only guaranteed for $c = 0, \beta = 0$. In fact, the corresponding objective value $z_{sdp} = x_{sdp}^\top Lx_{sdp}$ may be even smaller than the optimal value s_p^* of the semidefinite relaxation (4.7). For this reason, different modifications of the randomized hyperplane technique have been proposed in the literature [55, 209, 201] which are able to find feasible solutions for bisection problems (i.e. $c = e, \beta = 0$) with a guaranteed performance ratio. This is achieved e.g. by greedily swapping certain points from the larger to the smaller part [55], or rearranging the vectors v_i on the sphere by outward rotations prior to the randomized rounding procedure [209, 201]. A general framework for such approximation algorithms is given in [76].

However, we stick to the original randomized hyperplane technique even when a linear constraint is present. The main reason for this decision is that for the applications investigated in this work, it is usually not mandatory to

find an exactly feasible solution: all we demand is that the final solution is balanced reasonably. Hence, the balancing constraint $c^\top x = \beta$ in (4.5) rather serves as a strong bias to guide the search to a meaningful solution than as a strict requirement. For this reason, we can apply the SDP relaxation even in cases where no feasible combinatorial solution exists (see Section 4.4).

Finally, Figure 4.3 indicates the performance of the randomized hyperplane technique in practice. For a special reconstruction problem considered in Section 4.4.1 (cf. Figure 4.5), the solution of the corresponding SDP relaxation is computed and rounded with 500 different hyperplanes. The frequency of the objective values obtained in this way is depicted by the histogram in Figure 4.3: while most solutions are within 20% of the optimal solution z^*, the best objective value z_{sdp} found by randomized rounding has a relative error of merely 1.2%. This shows that in practice, it is sufficient to compute only a limited number of randomized hyperplanes (cf. [131]). In general, we will apply the rounding procedure n times in our experiments in order to find the final binary solution $x_{sdp} \in \{-1, +1\}^n$.

4.2.4 Performance Bounds

If no balancing constraint is given (i.e. $c = 0, \beta = 0$) — as for the perceptual grouping (Section 2.2) and the restoration problem (Section 2.3) — the combinatorial solution x_{sdp} obtained with the randomized hyperplane technique is feasible. In this case, it makes sense to compare the corresponding objective value $z_{sdp} = x_{sdp}^\top L x_{sdp}$ with the objective value $s_p^* = L \bullet X^*$ of the SDP relaxation (4.7) and the objective value $z^* := x^{*\top} L x^*$ of the optimal combinatorial solution x^*. Note that the following relations between these values are always valid:

$$s_p^* \leq z^* \leq z_{sdp} \leq E[z] \ ,$$

where $E[z]$ denotes the *expected* objective value produced by the randomized hyperplane technique.[5] Based on the results of Goemans and Williamson [66], we obtain the following suboptimality bound on $E[z]$:

Theorem 4.5. *The expected value $E[z]$ of the objective function $z = x^\top L x$ for a combinatorial solution x calculated with the randomized hyperplane technique is bounded by*

$$E[z] \leq \alpha s_p^* + (1 - \alpha) \sum_{i,j} |L_{ij}| \ ,$$

where

$$\alpha = \min_{0 \leq \gamma \leq \pi} \frac{2}{\pi} \frac{\gamma}{1 - \cos \gamma} \approx 0.87856 \ .$$

Proof. Using the notation of the previous section, let $V = (v_1, \ldots, v_n)^\top$ denote a factor of the optimal solution $X^* = VV^\top$ of the SDP relaxation (4.7). Then

[5] In contrast to this, note that if a balancing constraint is present (i.e. $c \neq 0$), the combinatorial solution x_{sdp} obtained by randomized approximation may yield an objective value $z_{sdp} \leq s_p^*$, since x_{sdp} is not required to satisfy the balancing constraint.

for the combinatorial solution x obtained with the hyperplane represented by the normal vector r, we have:

$$z = \sum_{i,j} L_{ij} x_i x_j = \sum_{i,j} L_{ij} \operatorname{sgn}(v_i^\top r) \operatorname{sgn}(v_j^\top r) \; .$$

The following propositions are easy to prove:

- $\Pr[\operatorname{sgn}(v_i^\top r) \neq \operatorname{sgn}(v_j^\top r)] = \frac{1}{\pi} \arccos(v_i^\top v_j)$ [66, Lemma 3.2],
- $\frac{\arccos(t)}{\pi} \geq \frac{1}{2}\alpha(1-t)$ for $-1 \leq t \leq 1$ [66, Lemma 3.4],
- $1 - \frac{\arccos(t)}{\pi} \geq \frac{1}{2}\alpha(1+t)$ for $-1 \leq t \leq 1$ [66, Lemma 3.2.2].

Using these facts and the linearity of the expectation, we deduce

$$\begin{aligned}
E[z] &= \sum_{i,j} L_{ij} E[\operatorname{sgn}(v_i^\top r) \operatorname{sgn}(v_j^\top r)] \\
&= \sum_{i,j} L_{ij} \Big(\Pr[\operatorname{sgn}(v_i^\top r) = \operatorname{sgn}(v_j^\top r)] - \Pr[\operatorname{sgn}(v_i^\top r) \neq \operatorname{sgn}(v_j^\top r)] \Big) \\
&= \sum_{i,j} L_{ij} \Big(1 - 2 \Pr[\operatorname{sgn}(v_i^\top r) \neq \operatorname{sgn}(v_j^\top r)] \Big) \\
&= \sum_{i,j} L_{ij} - \sum_{i,j} L_{ij} \frac{2}{\pi} \arccos(v_i^\top v_j) \\
&\leq \sum_{i,j} L_{ij} - \sum_{L_{ij}>0} L_{ij} \alpha (1 - v_i^\top v_j) - \sum_{L_{ij} \leq 0} L_{ij} \Big(2 - \alpha(1 + v_i^\top v_j) \Big) \\
&= \alpha \sum_{i,j} L_{ij} v_i^\top v_j + \sum_{i,j} L_{ij} - \alpha \sum_{i,j} |L_{ij}| + 2 \sum_{L_{ij} \leq 0} |L_{ij}| \\
&= \alpha s_p^* + (1-\alpha) \sum_{i,j} |L_{ij}| \; ,
\end{aligned}$$

which is the desired result. □

Note that in contrast to the result given in [66] for the max-cut problem, the bound in Theorem 4.5 contains the problem-dependent constant $\sum_{i,j} |L_{ij}|$. However, since we do not impose any restrictions on the entries of L, it is not possible to derive a more general bound in this form.

A different, weaker *relative* bound (for maximization problems) is presented by Nesterov [131], who extends the results of Goemans and Williamson to general problem matrices L that are allowed to contain positive and negative entries. Reformulating his results for minimization problems of the form (4.5) (without the balancing constraint) yields the following theorem:

Theorem 4.6. *Let z^* and $z^*_{\max}$ denote the minimum and the maximum value, respectively, of the objective function $z = x^\top L x$ subject to the integer constraint $x \in \{-1, +1\}^n$. Using the randomized hyperplane technique based on the SDP relaxation* (4.7)*, we obtain for the expected objective value $E[z]$:*

$$\frac{E[z] - z^*}{z^*_{\max} - z^*} \leq \frac{\pi}{2} - 1 < \frac{4}{7} \; .$$

Proof. The result can be derived directly from the corresponding results in [131] or [200]. For completeness, we include the main part of the proof here. We need the following propositions (which are easy to prove):

- $\arccos(t) + \arcsin(t) = \frac{\pi}{2}$ for $|t| \leq 1$.
- The dual of the semidefinite relaxation of the problem to find $z^*_{\max}$ is

$$\begin{aligned} s^*_{\max} := \min_{y \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} y_i \\ \text{s.t.} \quad & \operatorname{Diag}(y) - L \succeq 0 \; . \end{aligned} \tag{4.10}$$

- Let $X \succeq 0$ with $X_{ii} \leq 1$ for all i. Then $\arcsin X - X \succeq 0$, where $(\arcsin X)_{ij} = \arcsin X_{ij}$ is taken element-wise [131, Corollary 3.2]. Using Fejer's Theorem (cf. Theorem A.5), this directly yields $A \bullet \arcsin X \geq A \bullet X$ for all $A \succeq 0$.

Let X^* denote the optimal solution of the primal SDP relaxation (4.7), and define $y \in \mathbb{R}^n$ to be a feasible solution of (4.10). Then we can deduce, starting with a fact from the proof of Theorem 4.5:

$$\begin{aligned} E[z] &= \sum_{i,j} L_{ij} \left(1 - \frac{2}{\pi} \arccos X^*_{ij}\right) \\ &= \sum_{i,j} L_{ij} \frac{2}{\pi} \arcsin X^*_{ij} \\ &= -\frac{2}{\pi} (\operatorname{Diag}(y) - L) \bullet \arcsin X^* + \frac{2}{\pi} \operatorname{Diag}(y) \bullet \arcsin X^* \\ &\leq -\frac{2}{\pi} (\operatorname{Diag}(y) - L) \bullet X^* + \frac{2}{\pi} \sum_i \tfrac{\pi}{2} y_i \\ &= \frac{2}{\pi} L \bullet X^* - \frac{2}{\pi} \sum_i y_i + \sum_i y_i \\ &= \frac{2}{\pi} s^*_p + \left(1 - \frac{2}{\pi}\right) \sum_i y_i \; . \end{aligned}$$

Since this is valid for every feasible y, it especially holds for the optimal solution y^* of (4.10), and we can conclude

$$z^* \leq E[z] \leq \tfrac{2}{\pi} s^*_p + (1 - \tfrac{2}{\pi}) s^*_{\max} \; .$$

Analogously, we get

$$\tfrac{2}{\pi} s^*_{\max} + (1 - \tfrac{2}{\pi}) s^*_p \leq z^*_{\max} \; .$$

Combining these two equations, it follows that

$$\begin{aligned} E[z] &\leq \tfrac{2}{\pi} s^*_p + (1 - \tfrac{2}{\pi}) \left(\tfrac{\pi}{2} z^*_{\max} - (\tfrac{\pi}{2} - 1) s^*_p\right) \\ &= (\tfrac{\pi}{2} - 1) z^*_{\max} + (2 - \tfrac{\pi}{2}) s^*_p \\ &\leq (\tfrac{\pi}{2} - 1) z^*_{\max} + (2 - \tfrac{\pi}{2}) z^* \; , \end{aligned}$$

which yields the desired result by direct transformation. □

Unfortunately, this bound is quite weak: the expected objective value $E[z]$ can only be guaranteed to lie within the best $\frac{4}{7}$ of the total range of possible objective values $z^*_{\max} - z^*$ (which in addition depends on the problem instance). In practice, however, we do not seek $E[z]$ but the best possible value z_{sdp} of the objective function by applying the randomized hyperplane technique several times. As we will confirm in Section 4.4.1, this results in a much better performance than is indicated by the bounds presented above (also cf. Figure 4.3). Moreover, note that for most alternative optimization approaches applicable to the general problem class considered here, performance bounds are lacking completely.

4.3 Relation to Spectral Relaxation

In this section we will compare the convex relaxation approach with spectral relaxation approaches. Besides presenting a general formulation of the SDP relaxation as an eigenvalue optimization problem, we will show that for unsupervised partitioning problems, the SDP relaxation approach always compares favorably with the computation of the Fiedler vector (cf. Section 3.1.3).

4.3.1 Spectral Formulation of the SDP Relaxation

The idea to reformulate the semidefinite relaxation of a combinatorial problem as an eigenvalue optimization problem dates back to Delorme and Poljak [42], who examine the max-cut problem in this way. In this section, we extend their idea to the SDP relaxation of the more general combinatorial problem (4.5).

Starting with the dual problem formulation (4.6), we first parameterize y as $y = \alpha e - v$, where $e^\top v = 0$. Then the positive semidefiniteness constraint $0 \preceq L - y_0 cc^\top - \mathrm{Diag}(y) = L - y_0 cc^\top + \mathrm{Diag}(v) - \alpha I$ is equivalent to $\lambda_{\min}(L - y_0 cc^\top + \mathrm{Diag}(v)) \geq \alpha$, since the subtraction of αI reduces each eigenvalue of a matrix by α. This leads to the following spectral representation of (4.6):

$$\begin{aligned} s^*_d &= \max_{\substack{y_0, y \\ L - y_0 cc^\top - \mathrm{Diag}(y) \succeq 0}} \beta^2 y_0 + e^\top y \\ &= \max_{\substack{y_0, \alpha, e^\top v = 0 \\ \lambda_{\min}(L - y_0 cc^\top + \mathrm{Diag}(v)) \geq \alpha}} \beta^2 y_0 + n\alpha \\ &= \max_{y_0, e^\top v = 0} \beta^2 y_0 + n\lambda_{\min}\Big(L - y_0 cc^\top + \mathrm{Diag}(v)\Big) \\ &= \max_{y_0, e^\top v = 0} n\lambda_{\min}\Big(L - y_0(cc^\top - \tfrac{\beta^2}{n} I) + \mathrm{Diag}(v)\Big) \ . \end{aligned} \tag{4.11}$$

Next, we want to compare the spectral bound (4.11) with another spectral bound which is derived by a different relaxation of the original problem (4.5). The basic idea is to add the redundant constraint $x^\top x = n$, but to perform Lagrangian relaxation only on the integer constraints $x_i^2 = 1$ (cf. Section 4.2.1), which results in

$$z^* \geq z^*_{SR} := \max_y \min_{\substack{c^\top x = \beta \\ x^\top x = n}} x^\top \big(L - \mathrm{Diag}(y)\big) x + e^\top y \ .$$

Substituting $y = \alpha e - v$ as above, we obtain

$$\begin{aligned} z^*_{SR} &= \max_{e^\top v = 0} \min_{\substack{c^\top x = \beta \\ x^\top x = n}} x^\top \big(L + \mathrm{Diag}(v) - \alpha I\big) x + \alpha n - e^\top v \\ &= \max_{e^\top v = 0} \min_{\substack{c^\top x = \beta \\ x^\top x = n}} x^\top \big(L + \mathrm{Diag}(v)\big) x \\ &= \max_{e^\top v = 0} \; n \min_{\substack{c^\top x = \frac{\beta}{\sqrt{n}} \\ x^\top x = 1}} x^\top \big(L + \mathrm{Diag}(v)\big) x \;, \end{aligned} \tag{4.12}$$

where the last equation follows (with slight abuse of notation) by substituting x with $\sqrt{n}x$. For the special case $\beta = 0$, we can reformulate this relaxation as an eigenvalue bound by projecting onto the orthogonal complement $c^\perp$ of c:

$$z^*_{SR,0} := \max_{e^\top v = 0} n \lambda_{\min} \Big(V^\top \big(L + \mathrm{Diag}(v)\big) V \Big) \;, \tag{4.13}$$

where $V \in \mathbb{R}^{n \times (n-1)}$ contains an orthonormal basis of $c^\perp$ as columns, i.e. $V^\top c = 0$, $V^\top V = I$.

This spectral bound is a straightforward generalization of the constant constraint vector case ($c = e$), for which it was first provided by Boppana [19] (with $\beta = 0$) and Rendl and Wolkowicz [153] (in the form (4.12), for general $\beta \neq 0$), independently. For this special case of $c = e$, Poljak and Rendl [144] show the equivalence of the relaxation (4.12) and the semidefinite relaxation (4.11) by investigating general graph bisection problems, i.e. for L being a Laplacian matrix. The following theorem extends this result to the more general case of arbitrary problem matrices $L \in \mathcal{S}^n$ and balancing constraint vectors $c \neq e$:

Theorem 4.7. *Assume that the primal SDP relaxation is feasible (see Theorem 4.4). Then the dual SDP relaxation* (4.6) *resp.* (4.11) *yields the same lower bound on the optimal solution of the combinatorial problem* (4.5) *as the (spectral) relaxation* (4.12) *(or* (4.13) *for* $\beta = 0$*):*

$$s^*_d = z^*_{SR} \quad \big(= z^*_{SR,0} \quad \textit{for } \beta = 0 \big) \;.$$

Proof. We mainly follow the argumentation of Ye and Zhang [202], who recently proved a similar result for the minimization of a homogeneous quadratic function subject to two homogeneous quadratic constraints. First, for ease of notation, we define $L(v) = L + \mathrm{Diag}(v)$. Consider the following subproblem of (4.12) for fixed $v \in \mathbb{R}^n$ (due to symmetry, we can replace the linear constraint $c^\top x = \beta$ with the quadratic constraint $(c^\top x)^2 = \beta^2$):

$$\begin{aligned} z^*_{SR}(v) := \min_{x \in \mathbb{R}^n} \quad & x^\top L(v) x \\ \text{s.t.} \quad & x^\top c c^\top x = \beta^2 \\ & x^\top I x = n \;. \end{aligned} \tag{4.14}$$

Using Lagrangian relaxation, we obtain the following primal-dual pair of semidefinite programs:

$$\begin{aligned} s_p^*(v) := \min_{X \succeq 0} \quad & L(v) \bullet X \\ \text{s.t.} \quad & cc^\top \bullet X = \beta^2 \\ & I \bullet X = n \end{aligned} \qquad \begin{aligned} s_d^*(v) := \max_{y_0, y_1} \quad & \beta^2 y_0 + n y_1 \\ \text{s.t.} \quad & Z = L(v) - y_0 cc^\top - y_1 I \\ & Z \succeq 0 \,. \end{aligned} \tag{4.15}$$

Note that the dual of this relaxation corresponds to (4.11), if the maximization over v is included (and y_1 is substituted by α). We now consider two cases:

Case $\beta \neq 0$ (unbalanced partitioning). Observe that for (4.15), both the primal optimal solution (as the dual is strictly feasible) and the dual optimal solution are attained: for the dual, this is a consequence of the fact that the objective function $s_d(v, y_0) := \beta^2 y_0 + n\lambda_{\min}(L(v) - y_0 cc^\top)$ is continuous in y_0 and may reach values larger than $s_d(v, 0) = n\lambda_{\min}(L(v))$ only on a bounded interval, namely for $y_0 \in \left[-\frac{n(\lambda_{\max} - \lambda_{\min})}{\beta^2}, \frac{n(\lambda_{\max} - \lambda_{\min})}{n\|c\|^2 - \beta^2}\right]$, with $\lambda_{\min}$ and $\lambda_{\max}$ denoting the smallest and largest eigenvalue of $L(v)$, respectively. This is easily seen by considering the Rayleigh quotient $\frac{x^\top (L(v) - y_0 cc^\top) x}{x^\top x}$ for $x = c$ (upper bound) and $x = v$ with $v^\top c = 0$ (lower bound) (cf. [144, Lemma 2.5]).

Therefore, a pair of complementary optimal solutions X^* and (y_0^*, y_1^*, Z^*) with zero duality gap exists for (4.15). We now apply the following proposition, which is proven in [177, 202]:

> Assume that the positive semidefinite matrix $X^* \in \mathcal{S}_+^n$ has rank r, and let $G \in \mathcal{S}^n$ be a symmetric matrix with $G \bullet X^* = 0$. Then X^* can be decomposed into rank-one matrices $x_i x_i^\top$ such that
>
> $$X^* = \sum_{i=1}^{r} x_i x_i^\top$$
>
> and $x_i^\top G x_i = 0$ for all $i = 1, \ldots, r$.

Since in our case, the constraints of the primal relaxation yield $G \bullet X^* = 0$ with $G = \frac{1}{\beta^2} cc^\top - \frac{1}{n} I$, we have $x_i^\top (\frac{1}{\beta^2} cc^\top - \frac{1}{n} I) x_i = 0$ for all $i = 1, \ldots, r$. As $I \bullet X^* = \sum_i x_i^\top I x_i = n$ (second constraint of the primal relaxation), there exists an x_j, $1 \leq j \leq r$, with $x_j^\top x_j =: \tau > 0$. Setting $x^* := \sqrt{\frac{n}{\tau}} x_j$ then yields a feasible solution $x^* x^{*\top}$ of rank one for the primal relaxation. In order to show that this solution is also optimal, first observe that due to Fejer's Theorem (cf. Theorem A.5), we get $Z^* \bullet x_i x_i^\top \geq 0$ for all $i = 1, \ldots, r$. Using the linearity of the trace and the complementary slackness of Z^* and X^* (cf. [196]), this results in

$$0 \leq Z^* x^* x^{*\top} = \frac{n}{\tau} Z^* \bullet x_j x_j^\top \leq \frac{n}{\tau} Z^* \bullet X^* = 0 \,,$$

which shows that the complementary slackness condition is also satisfied for $x^* x^{*\top}$. Hence, the relaxation is exact, i.e. x^* is optimal for (4.14), and maximization over $e^\top v = 0$ yields the desired result.

Case $\beta = 0$ (equipartitioning). In this case, we may assume without loss of generality that the dual optimal solution of (4.15) is not attained — otherwise,

we can use a similar argumentation as for the case $\beta \neq 0$ to prove the theorem (by setting $G = cc^\top$). Due to the feasibility of the SDP relaxation (4.6), the optimal objective value $s_d^*(v)$ of the dual is finite. Since the dual objective function is continuous in y_0, this means that $s_d^*(v)$ can only be approached for $|y_0|$ approaching infinity. In this case, the inner minimization in the dual objective function will only be finite if $c^\top x$ becomes 0:

$$\begin{aligned} s_d^*(v) &= n \max_{y_0} \lambda_{\min}\left(L(v) - y_0 cc^\top\right) \\ &= n \lim_{y_0 \to -\infty} \min_{\|x\|=1} x^\top L(v) x - y_0 (c^\top x)^2 \\ &= n \min_{\substack{\|x\|=1 \\ c^\top x = 0}} x^\top L(v) x \;=\; z_{SR}^*(v) \; . \end{aligned}$$

Maximizing over $e^\top v = 0$ completes the proof. □

4.3.2 Comparison with Spectral Relaxation Techniques

In this section, we compare the SDP relaxation approach with the spectral relaxation techniques presented for unsupervised partitioning tasks in Section 3.1. To this end, first take a closer look at the following *weaker* spectral relaxation of (4.5) for the special case $\beta = 0$:

$$\begin{aligned} z_{SR2}^* := \min_{x \in \mathbb{R}^n} \quad & x^\top L x \\ \text{s.t.} \quad & x^\top x = n \\ & c^\top x = 0 \; , \end{aligned} \tag{4.16}$$

i.e. the combinatorial constraint $x \in \{-1, +1\}^n$ is directly relaxed to $\|x\|^2 = n$. For this relaxation, the following lemma holds:

Lemma 4.8. *Let $V \in \mathbb{R}^{n \times (n-1)}$ denote the matrix which contains an orthonormal basis of $c^\perp$ (cf. last section). Then*

$$z_{SR2}^* = n \lambda_{\min}(V^\top L V) \; , \tag{4.17}$$

and the solution of (4.16) *is given by $x^* = \sqrt{n} V w_0$, where w_0 is the eigenvector of norm one corresponding to the smallest eigenvalue of $V^\top L V$.*

Proof. Define the orthonormal matrix $P := \left(\frac{c}{\|c\|}, V\right) \in \mathbb{R}^{n \times n}$, and let $w := \frac{1}{\sqrt{n}} V^\top x$. Then we have

$$P^\top x = \begin{pmatrix} \frac{c^\top x}{\|c\|} \\ V^\top x \end{pmatrix} = \begin{pmatrix} 0 \\ \sqrt{n} w \end{pmatrix} \; ,$$

which gives $x = PP^\top x = \sqrt{n} V w$. This results in $x^\top x = n w^\top V^\top V w = n w^\top w$ and $c^\top x = \sqrt{n} c^\top V w = 0$, and substitution into (4.16) yields

$$z_{SR2}^* = \min_{\substack{x^\top x = n \\ c^\top x = 0}} x^\top L x = \min_{w^\top w = 1} \; n w^\top V^\top L V w = n \lambda_{\min}(V^\top L V) \; .$$

The optimal solution is thus given by the eigenvector w_0 corresponding to $\lambda_{\min}(V^\top L V)$, which directly yields $x^* = \sqrt{n} V w_0$. □

For unsupervised partitioning tasks, the problem matrix L is the Lagrangian matrix of a graph. In this case, we immediately see that the spectral relaxation (4.16) corresponds to the computation of the Fiedler vector, if the equipartition constraint vector $c = e$ is used (cf. Section 3.1.3): as e is the eigenvector corresponding to the smallest eigenvalue $\lambda_1(L) = 0$, the solution of (4.16) becomes $z^*_{SR2} = n\lambda_2(L)$. Comparing the results of Lemma 4.8 and Theorem 4.7 now directly reveals that (4.17) is a special case (for $v = 0$) of (4.13). This implies the following lemma, which shows the superiority of the SDP relaxation approach for the unsupervised equipartition problem:

Lemma 4.9. *For the unsupervised equipartition problem* (2.5)*, the SDP relaxation* (4.7) *yields a better lower bound on the optimal objective value than spectral relaxation based on the Fiedler vector* (3.13)*:*

$$z^*_{SR2} \leq z^*_{SR} = s^*_d \,.$$

Apart from this fact of being less tight concerning the value of the objective function, the spectral relaxation approach has another disadvantage: to obtain the corresponding combinatorial solution x of (2.6), the solution x^* of (4.16) has to be thresholded suitably (see Section 3.1.2). However, the appropriate choice of the splitting value is not obvious and can be time consuming (if all possible splitting points are tested); actually, as we will see in the next section, an unsupervised choice of the threshold value may fail completely. In contrast to this, the SDP relaxation uses a *randomized* algorithm to compute the combinatorial solution, which does not depend on a critical parameter setting and even allows the derivation of probabilistic performance bounds (cf. Section 4.2.4).

On the other hand, the computational effort for solving the spectral relaxation with the Fiedler vector is smaller than for solving the SDP relaxation, as the solution is an n-dimensional vector opposed to an $n \times n$-dimensional matrix. This fact permits handling larger problem instances with the spectral relaxation approach.

Finally, note that a direct comparison of the SDP relaxation with spectral relaxation based on the normalized cut criterion (see Section 3.1.4) is not adequate, since the normalized cut approach employs a different, i.e. normalized, objective function. Applying Lagrangian relaxation to the normalized cut criterion directly, however, is to no avail either: as in the problem formulation (3.8), the binary constraint on the entries of $x \in \{-\beta, \frac{1}{\beta}\}^n$ depends on the preliminarily unknown variable β, it cannot be included in the relaxation (in contrast to the $(-1, +1)$-constraint in (4.5)). Yet if it is dropped completely, the resulting SDP relaxation yields the same optimum as the computation of the second smallest eigenvalue of the normalized Laplacian matrix $L' = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. This can be seen by using the same argumentation as in the proof of Theorem 4.7, or by direct investigation of the eigenvalues of the involved matrices: first note that the normalized cut criterion (3.16) can be reformulated as

$$\begin{aligned} \lambda_2(L') = \min_{x \in \mathbb{R}^n} \quad & x^\top L' x \\ \text{s.t.} \quad & x^\top x = 1 \\ & d^\top x = 0 \,, \end{aligned}$$

where $d := D^{\frac{1}{2}}e$. It is easy to verify that Lagrangian relaxation of this problem results in the dual SDP

$$\max_{y_0} \lambda_{\min}(L' - y_0 dd^\top) \ .$$

Using the eigenvalue decomposition $L' = \sum_i \lambda_i v_i v_i^\top$ (cf. Theorem A.1) with $\lambda_1 = 0$ and $v_1 = d$ denoting the smallest eigenvalue of L' and the corresponding eigenvector, we see that

$$L' - y_0 dd^\top = \sum_{i=2}^{n} \lambda_i v_i v_i^\top + (\lambda_1 - y_0) dd^\top$$

has the eigenvalues $\lambda_2, \ldots, \lambda_n$ and $-y_0$. Thus maximizing the smallest eigenvalue of $L' - y_0 dd^\top$ over y_0 in the SDP relaxation also results in $\lambda_2(L')$.

4.4 Experimental Results

In this section, we investigate the performance of the SDP relaxation approach experimentally. We start with reporting statistical results for ground-truth experiments based on restoration problems for noisy one-dimensional signals (Section 4.4.1). The application of the SDP relaxation method to different real scenes from all problem types that were presented in Chapter 2 is the topic of Sections 4.4.3–4.4.5. Furthermore, we give a brief discussion of similarity measures in Section 4.4.2, and study the computational complexity of the SDP relaxation in practice in Section 4.4.6.

4.4.1 Ground-Truth Experiments

In order to be able to analyze the performance of the SDP relaxation method described in Section 4.2 on a statistical basis, the approximation obtained for the problem under consideration needs to be compared with the corresponding global optimum (the *ground-truth data*) of the original functional (4.5). To this end, we investigate the restoration of noisy one-dimensional signals based on the functional (2.11), as in this case the global optimum can be easily and quickly computed e.g. by using dynamic programming (or any other of the methods mentioned at the end of Section 2.3). Moreover, we can also compare the reconstructions with the "true" original signal, which yields valuable results concerning the significance of the restoration functional and the quality of the approximative solutions.

Our experiments are based on the one-dimensional synthetic signal x' depicted in Figure 4.4 which involves transitions at multiple spatial scales. This signal is superimposed with Gaussian white noise with zero mean and a standard deviation of $\sigma = 1.0$, resulting in noisy signals g like the one shown in Figure 4.5, top. Using the restoration functional (2.11) with each signal element being connected only to its two neighbors, we compute the combinatorial solution x_{sdp} from the corresponding SDP relaxation (4.7), and compare it with the global

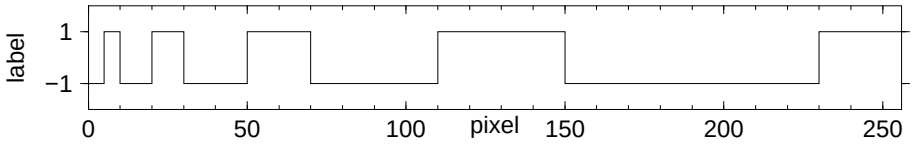


Figure 4.4: A one-dimensional signal x' comprising multiple spatial scales.

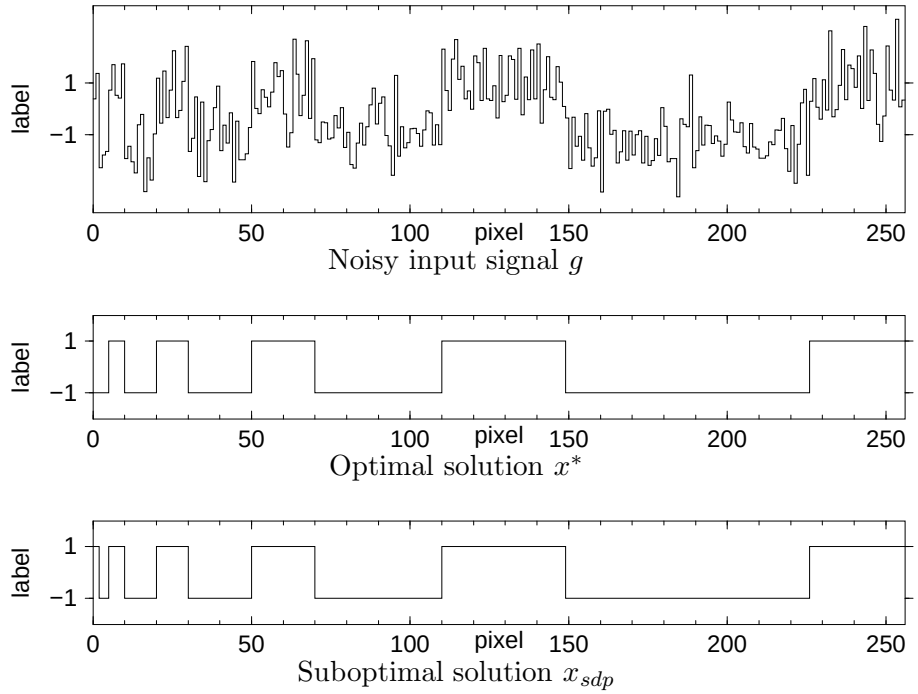


Figure 4.5: A representative example illustrating the statistics shown in Figure 4.6. Note that the SDP reconstruction differs from the optimal solution only in the first two elements of the signal.

optimum x^* of (2.11) and the original signal x', respectively. A representative example of such a restoration is given in Figure 4.5, bottom.

For varying values of the parameter λ controlling the desired smoothness, this experiment is repeated 1000 times with different noisy inputs in order to derive some significant statistics. Based on the results, we then calculate the following quantities for each λ-value:

$\overline{\Delta z}$**:** the sample mean of the relative gap $\Delta z = \frac{z_{sdp}-z^*}{z^*}$ with respect to the objective function values of the suboptimal solution x_{sdp} and the optimal solution x^*.

$\sigma_{\Delta z}$**:** the sample standard deviation of the gap Δz.

$\overline{\Delta z'}$**:** the sample mean of the relative gap $\Delta z' = \frac{|z_{sdp}-z(x')|}{z(x')}$ with respect to the objective function values of the suboptimal solution x_{sdp} and the synthetic original signal x'.

$\sigma_{\Delta z'}$**:** the sample standard deviation of the gap $\Delta z'$.

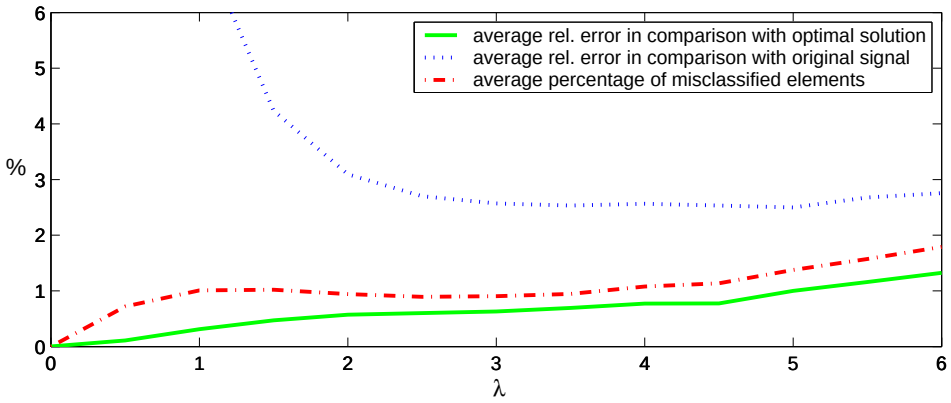


Figure 4.6: Statistics for the ground-truth experiment. For different values of the scale parameter λ, the figure shows the average relative errors $\overline{\Delta z}$ (comparison to the optimal solutions x^*) and $\overline{\Delta z'}$ (comparison to the original signal x') in the objective function for the solutions x_{sdp} obtained from the SDP relaxation. Also depicted is the average relative Hamming distance (percentage of misclassified elements) between the suboptimal solutions x_{sdp} and the optimal solutions x^*. Note that for $\lambda > 2$, the relative error is smaller than 3% for all measures.

Moreover, we also calculate the sample mean of the percentage of misclassified elements in x_{sdp} in comparison with the optimal solution x^* (relative *Hamming distance*).

The results are depicted in Figure 4.6. They reveal the accuracy of the suboptimal solutions obtained with the SDP relaxation: in comparison to the optimal solutions, both the average percentage of misclassified elements and the average relative error $\overline{\Delta z}$ of the objective function are below 2%, with the corresponding standard deviation ranging between $0.12\% \leq \sigma_{\Delta z} \leq 1.33\%$. This result is confirmed by the example in Figure 4.5, for which the optimal solution and the SDP solution only differ in two elements. This shows that in practice, the performance of the SDP relaxation approach can be much better than the bounds presented in Section 4.2.4.

Concerning the restoration of the original signal x', the quality of the SDP solutions also is remarkably good, at least for scale parameter values $\lambda \geq 2$: in this case, the average relative error $\overline{\Delta z'}$ of the objective function is below 3%, with the corresponding standard deviation ranging between $1.95\% \leq \sigma_{\Delta z'} \leq 2.69\%$ (cf. Figure 4.6). The high error rates for $\lambda < 2$ are caused by the dominating larger spatial scales in the signal x', which are not taken into account correctly by the restoration functional (2.11) for small λ-values. Moreover, note that x' in general does not coincide with the best solution x^* of this functional, which is revealed by the fact that the corresponding error rates are higher: $\overline{\Delta z'} > \overline{\Delta z}$. This is mainly due to the different spatial scales that occur in x', which cannot be covered completely by a single scale parameter λ. This indicates that more appropriate criteria should be used for the restoration of such signals, e.g. by incorporating suitable priors with respect to x' (cf. [20]). However, the derivation of such criteria is not the objective of this thesis.

4.4.2 Similarity Measures

Before we present the results of the SDP relaxation approach for real world images, we have to spend a few words on the derivation of suitable similarity measures for the unsupervised partitioning task presented in Section 2.1. Recall that the objective in unsupervised partitioning is to split a graph with some extracted image features as vertices into two coherent groups. Hence, the results depend on the edge-weights w_{ij} which encode the similarities between two extracted image features i and j. In this work, we only consider similarities that are derived from distances $d(i,j)$ between the image features as

$$w_{ij} = e^{-\left(\frac{d(i,j)}{\sigma}\right)^2} ,$$

where the normalization parameter σ is chosen application dependent (usually between 5% and 30% of the maximal distance; cf. Section 3.1.6). Note that such an exponential decay of the similarity values with some power of the distance measure is in accordance with results from psychophysical studies [166].

Concerning the distance measure $d(i,j)$, we use two different calculation methods in this section:

(i) Compute $d(i,j)$ for each feature pair (i,j) directly, e.g. by using the Euclidean distance in the feature space. Since for images, we do not include the location of the points as additional feature, this measure does not incorporate information on the underlying spatial neighborhood structure of the data.

(ii) Compute $d(i,j)$ only for spatially neighboring points, and derive the other distances between the points by calculating the shortest paths connecting them. This results in a similarity measure which favors spatially coherent structures (also cf. [52]).[6]

Since both methods result in completely connected graphs, the corresponding similarity matrices are dense.

As a motivation for the second method, consider Figure 4.7, which shows a set of points arranged in two spirally shaped regions. It was critically observed in [62] that spectral methods based on Euclidean distances fail to partition such "skewed" coherent groups. In fact, a direct pairwise comparison as in method (i) will always find points from the other group at a smaller distance than most of the points from the same group. This is illustrated by Figure 4.7, right: in the Delaunay graph corresponding to this point set, the shortest Euclidean path between two points of the same group traverses the other group. However, this also shows that the spatial context introduced by the Delaunay graph is not appropriate for a successful application of the second method. The main problem in this situation is that both the neighborhood structure and the edge

[6] An alternative modification to include spatial information is to additionally use a cut-off radius, i.e. to set the distance to infinity if two points are spatially too far apart (cf. [168]). In the extreme case, this corresponds to defining edges only between points that are direct neighbors. However, we do not use any cut-off radius in this section.

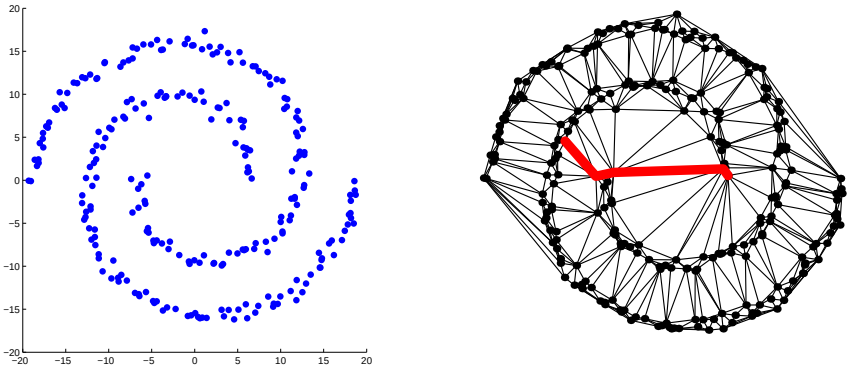


Figure 4.7: A skewed data distribution with two spiral-shaped groups. The right figure shows that within the corresponding Delaunay graph, the shortest *Euclidean path* between two points of the same group may traverse the other group.

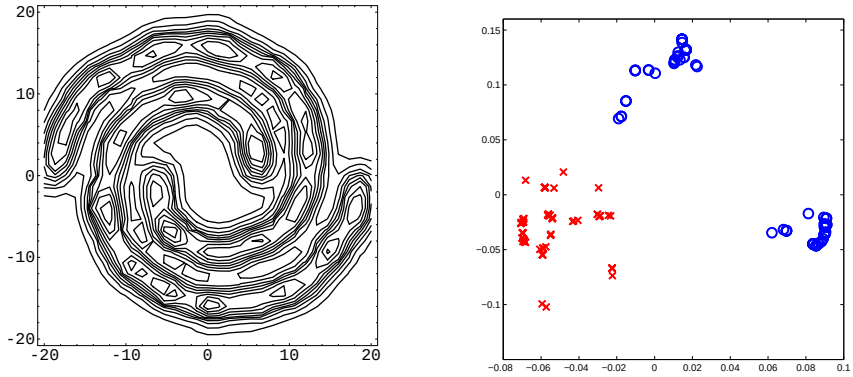


Figure 4.8: Left: Level lines of a Parzen estimate of the data distribution in Figure 4.7. This attribute yields a more appropriate description of the spatial context. **Right:** Visualization of the corresponding weighted-path distances (method (ii)), obtained by applying a metric scaling technique and projecting on the first two principal components: the two spiral-shaped groups, depicted by crosses and rings, now form more compact clusters.

weights are based on the same source, namely the Euclidean distances between the points. Typically, attributes differing from location (like color, texture, etc.) are used to define the pairwise distances. If this is the case, method (ii) provides a simple technique to appropriately exploit spatial coherency: calculating weighted paths as the distance measure based on the given neighborhood structure results in shorter distances *within* a spatially connected group, and in longer paths between weakly connected points (cf. [52]).

For the example shown in Figure 4.7, we simulate such an additional attribute by a Parzen density estimate [58] of the spatial data distribution (see Figure 4.8, left). The distances between neighboring points in the Delaunay graph are then obtained by estimating the energy which is needed to join the points along their direct connection. In this way, the given spatial neighborhood structure is represented by the distances more appropriately. The effect

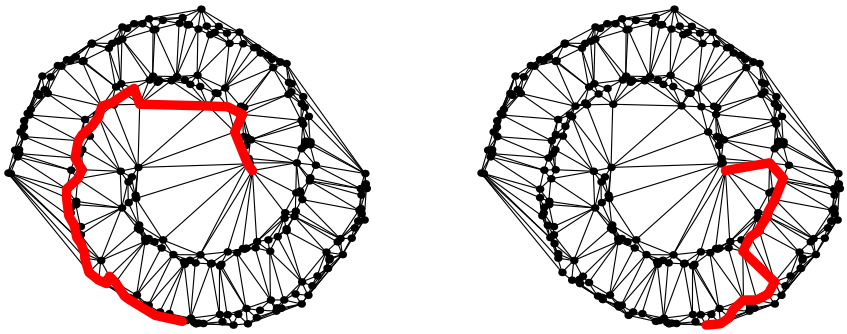


Figure 4.9: Weighted shortest paths based on the Parzen estimate. Compared to the Euclidean distance (Figure 4.7, right), within-group paths have become shorter (left), but may still be outperformed by paths traversing the other group (right).

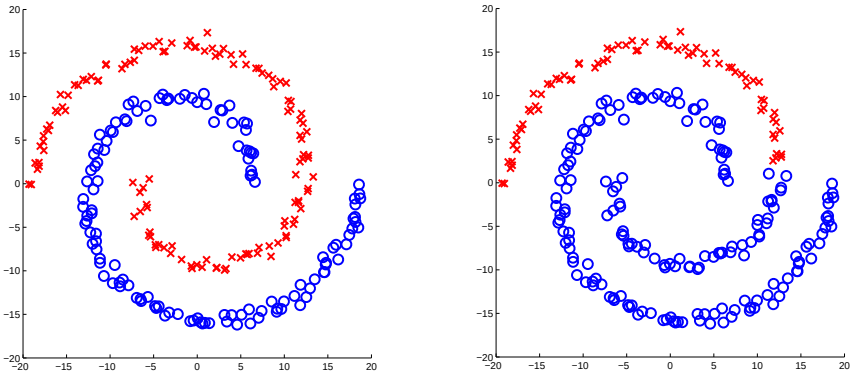
of applying method (ii) is visualized in Figure 4.8, right: approximating the resulting weighted path distances with Euclidean distances within 2D-space by using a classical metric scaling technique [37], it shows that the points within each spiral-shaped group (depicted by crosses and rings, respectively) have become more similar to each other. Accordingly, the partition task is now better defined, yet without becoming trivial: whereas weighted paths within a group have been shortened, the shortest paths between two points of the same group may still traverse the other group (see Figure 4.9).

Finally, we note that numerous other (dis)similarity measures for different computer vision tasks have been proposed in the literature, see e.g. [149, 156]. However, since the focus of this thesis is on studying the results of the SDP relaxation approach from an optimization point of view, we did not work on more elaborate computations of the similarity measures.

4.4.3 Binary Unsupervised Partitioning

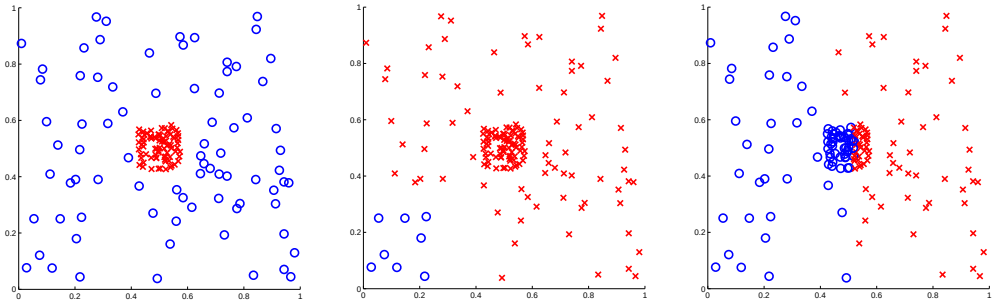
In this section, we present numerous results obtained by applying the SDP relaxation approach to artificial and real world unsupervised partitioning problems (cf. Section 2.1), and compare them to the corresponding results obtained with spectral relaxation. To this end, we decided to utilize the average cut criterion based on the Fiedler vector (see Section 3.1.3), since it resembles the SDP relaxation approach most (cf. Section 4.3.2). However, the experiments reveal that the best average cut threshold may result in very unbalanced partitionings for similarity values obtained with method (ii); in this case, we revert to the median threshold instead. This unfavorable behavior of the average cut is due to the fact that the vertex degrees d_i often vary considerably for the similarities acquired with method (ii). Therefore, the normalized cut criterion may be superior in this context, since it takes the d_i-values into account by normalizing the similarity matrix (see Section 3.1.1).

Moreover, unless stated otherwise, we use (4.5) with the equipartition constraint ($c = e$ and $\beta = 0$) for balancing in the experiments in this section.



SDP relaxation | Fiedler vector (best average cut)

Figure 4.10: Point set clustering for Figure 4.7, using distances based on weighted paths (method (ii)). Whereas the SDP relaxation produces the correct partitioning (left), spectral relaxation based on the average cut criterion (right) is not capable to separate the groups successfully in this case due to a less appropriate objective function (cf. Figure 4.8, right).



SDP relaxation | Fiedler vector (best average cut) | Fiedler vector (median threshold)

Figure 4.11: Point set clustering (cf. Figure 3.5) based on Euclidean distances (method(i), with $\sigma = 0.1$). SDP relaxation finds the correct partitioning (left), while spectral relaxation fails for both the best average cut (middle) and the median threshold (right).

Point Sets

As a first result, Figure 4.10 shows the partitioning of the two-spirals example from the previous section (Figure 4.7) obtained with the SDP relaxation (left) and with spectral relaxation (right), respectively. Although the similarity weights w_{ij} are calculated using the weighted-path metric from method (ii), the average cut relaxation still fails to compute the correct cut, whereas SDP relaxation partitions the spirals successfully. A look at Figure 4.8, right, indicates the reason: the average cut criterion favors to separate the dense cluster on the right from the rest of the points, and therefore is less appropriate in this case. However, it should be mentioned that the Fiedler vector produces the correct partitioning if we switch to the median threshold. Yet note in this context,

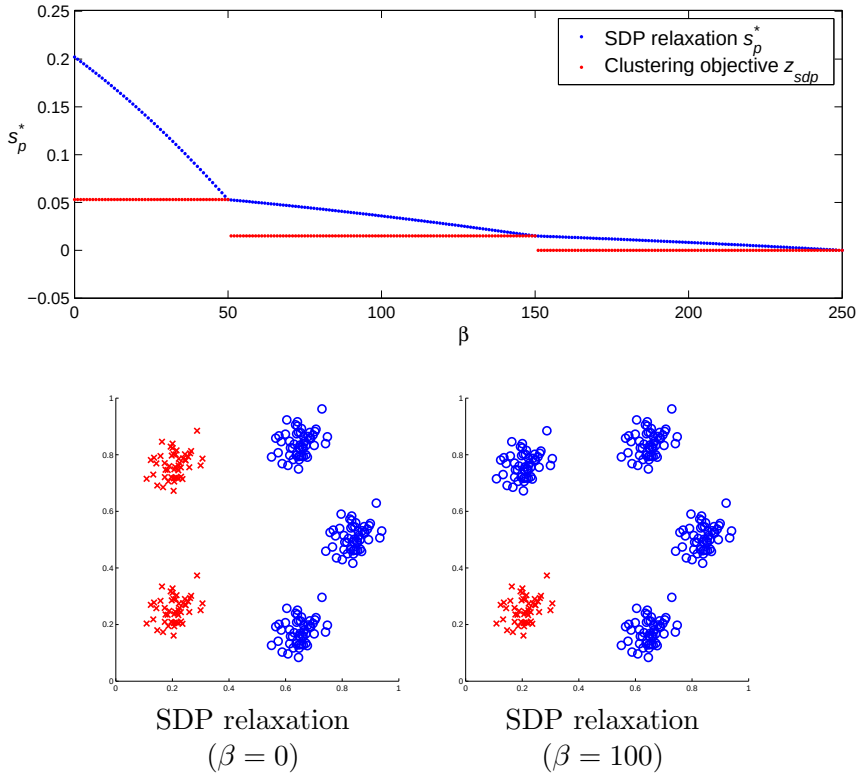


Figure 4.12: Influence of the balancing parameter β on the clustering result. While the optimal objective value s_p^* of the SDP relaxation continuously decreases towards 0 for increasing β, the corresponding objective values z_{sdp} of the clusterings obtained with the randomized hyperplane technique remain constant (top). This indicates the robustness of the SDP relaxation approach: even when β is chosen too small, the data set is divided into two reasonable components in accordance with the parameter value (bottom).

that the SDP relaxation works completely unsupervised, and does not depend on any thresholding heuristic.

Figure 4.11 depicts another situation, that was already examined in Section 3.1.6: a point set consisting of a dense cluster within equally distributed background clutter. The results for this example reflect the theoretical results of Section 4.3.2, showing the superiority of the SDP relaxation approach: although the similarities between the points are computed directly based on their Euclidean distances with method (i), the SDP relaxation successfully separates the dense cluster from the background — in contrast to spectral relaxation, which only achieves an unsatisfactory partitioning, even for different threshold values. The reason for this failure is due to the computed eigenvector, which does not give a clear cut value (cf. Figure 3.5). Furthermore, note that the balancing constraint $e^\top x = 0$ is not enforced for the solution obtained from the SDP relaxation: in accordance with the visual impression, the two parts contain 79 and 81 points, respectively.

Finally, Figure 4.12 demonstrates the influence of the parameter β on the clustering result. For this simple example consisting of five identical clusters

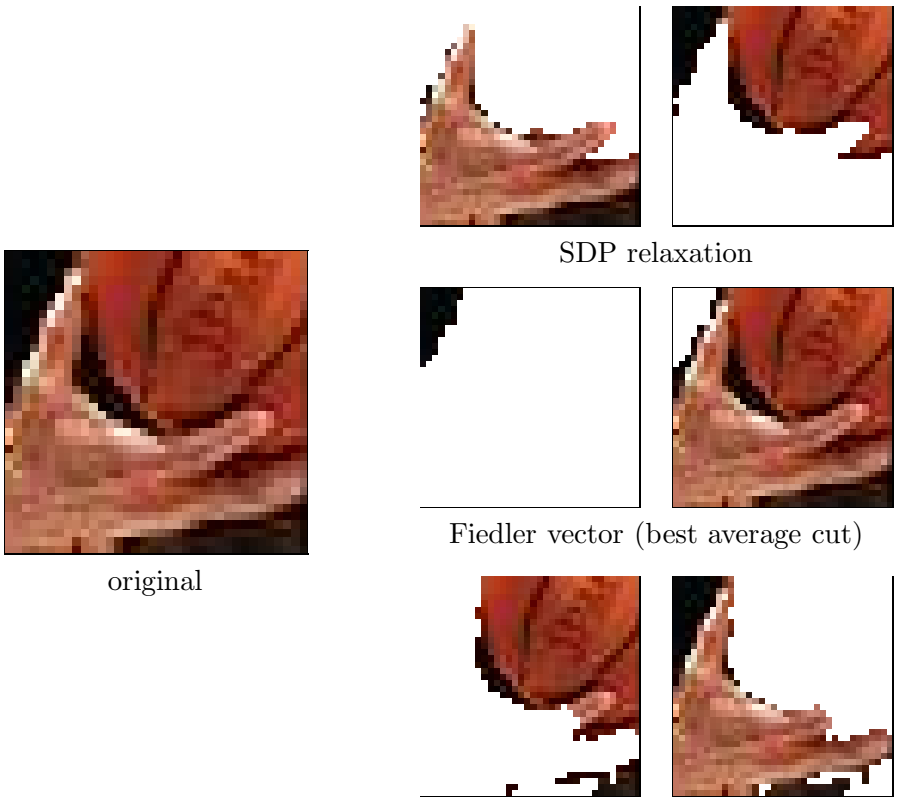


Figure 4.13: Color image partitioning. With similarities obtained with method (ii), a small patch (36×36 pixels) of a larger image (cf. Figure 3.6) is segmented on a pixel basis. While the SDP relaxation successfully separates the hand from the ball, spectral relaxation based on the Fiedler vector only yields unsatisfactory results for both the best average cut and the median threshold.

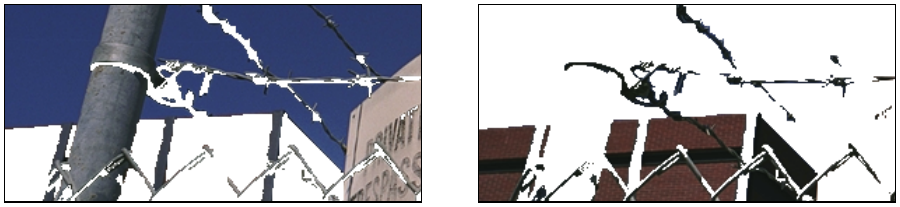
with each containing 50 points, the similarities are obtained with method (i), and β varies between 0 (which gives an equipartition constraint) and the maximum number of points. The results reveal that while the objective value s_p^* of the SDP relaxation continuously decreases towards 0 for increasing β-values, the randomized hyperplane technique over longer intervals finds constant clusterings that are in reasonable accordance with β. Hence, the balancing constraint acts as designated: a strong bias that guides the search to a meaningful solution and not as a strict requirement.

In this context, it should be mentioned that by introducing a nonzero value for the parameter β, the segmentation problem no longer remains unsupervised in the strict sense. In Section 5.2.3 we will suggest how the balancing constraint can be adjusted automatically.

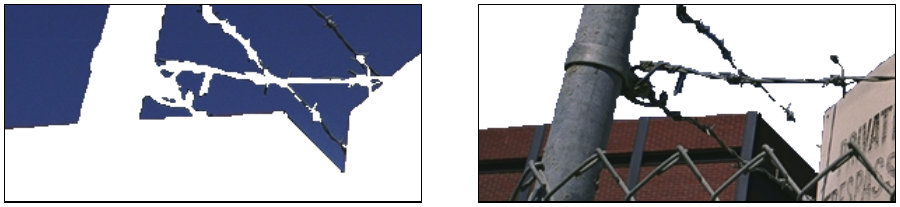
Color Images

As a first idea to study the partitioning of color images, we create the underlying neighborhood graph on a pixel basis by connecting horizontally and

Original image



SDP relaxation



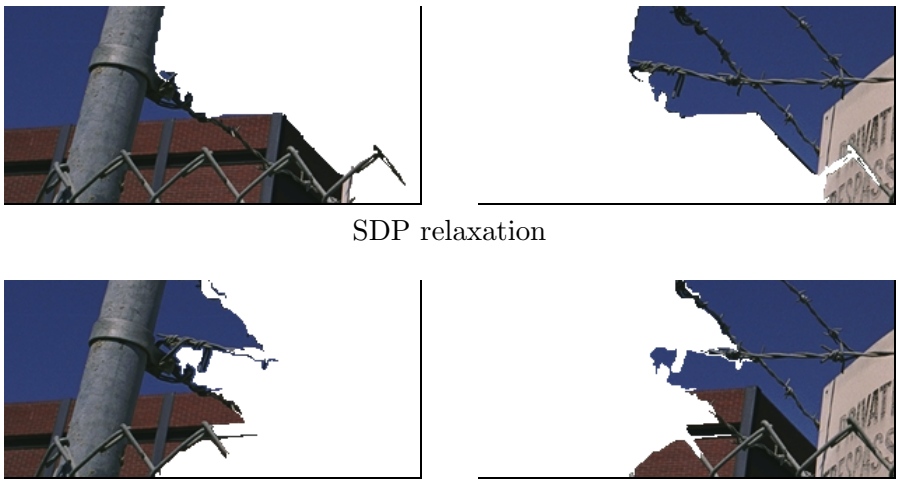
Fiedler vector (best average cut)

Figure 4.14: Color image partitioning, using method (i). Preprocessing the input image (298×141 pixels) yields 211 patches (cf. Figure 3.7). In the segmentations obtained with both SDP and spectral relaxation, patches of similar color are grouped together.

vertically adjacent pixels, and calculate the corresponding edge-weights based on the color differences in the perceptually uniform L*u*v* space. The remaining similarities are then obtained by applying method (ii) to favor spatially coherent structures.

The result for a small patch of a larger image (cf. Figure 3.6) is shown in Figure 4.13. For this example, the Fiedler vector only yields unsatisfactory partitionings: using the best average cut just separates the small black corner from the rest of the image, whereas the median threshold results in an incoherent segmentation (due to the same size constraint). In contrast to that, the SDP relaxation method successfully partitions the image by clearly separating the hand from the ball. Once more, note that the two groups arising from the SDP relaxation do not have the same size: they contain 641 and 655 pixels, respectively. However, better results may be obtained for spectral relaxation by reverting to another similarity measure (see Figure 3.6, where similarities are calculated only for neighboring pixels).

This first example already indicates the drawback of the SDP relaxation approach: the problem size soon becomes intractable when larger images are examined on the basis of pixels as input data. To be able to study the partitioning of large real world images, we therefore first compute an over-segmentation



SDP relaxation

Fiedler vector (median threshold)

Figure 4.15: Color image partitioning for the image from Figure 4.14, using method (ii). SDP relaxation as well as spectral relaxation based on the Fiedler vector result in segmentations that favor spatially coherent structures. However, the spectral relaxation result is influenced negatively by the requirement that both parts must have the same size.

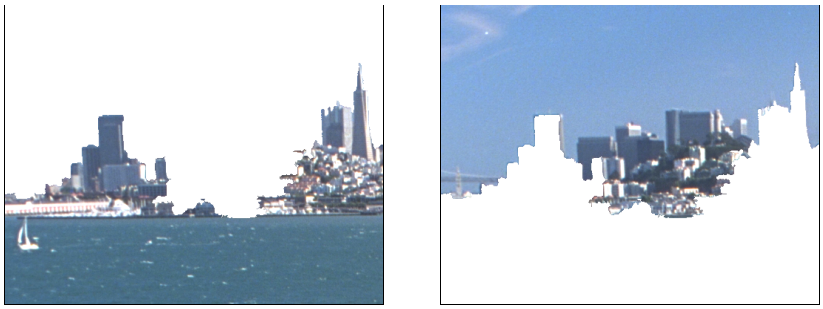
by applying the mean shift technique (see Section 3.2) at a fine spatial scale. This preprocessing step drastically reduces the size of the input data, without destroying any perceptually significant structure. Instead of having to deal with thousands of pixels, the graph vertices are now formed by the obtained image patches, and the corresponding similarities are computed based on the mean color difference between the patches in the perceptually uniform L*u*v* space.

We apply both methods (i) and (ii) to the color image shown at the top of Figure 4.14. The results approve the wide range of applicability and the success of the SDP relaxation method: whereas in the partitioning based on method (i), image patches of similar color are grouped together (see Figure 4.14), method (ii) yields a segmentation into two reasonable, spatially coherent parts (see Figure 4.15). For this example, the results obtained with spectral relaxation are also quite reasonable (see bottom of Figures 4.14 and 4.15). However, since the best average cut just separates one patch from the rest of the image, we need to revert to the median threshold for method (ii): this requirement influences the result negatively. As already mentioned, the normalized cut criterion may be more appropriate in this case. In this context, note again that the SDP relaxation approach works without any threshold.

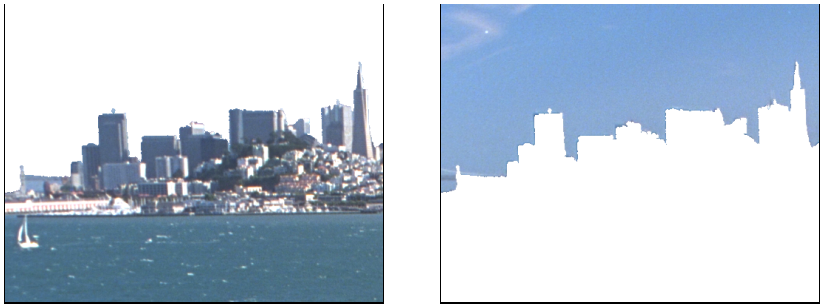
Choice of the Balancing Vector $\boldsymbol{c}$

So far, the size of the patches obtained from preprocessing the image with the mean shift technique was not considered by the segmentation process. This may yield unsatisfactory separation results. Figure 4.16 gives an example: here the sky accounts for nearly half of the image (approx. 43%), but the preprocessing step groups all of its pixels into one patch (cf. Figure 3.7). Since for $c = e$,

Original image



SDP relaxation (with $c = e$, $\beta = 0$)



SDP relaxation (with $c = m$, $\beta = 0$)

Figure 4.16: Color image partitioning for different balancing vectors c. Preprocessing the input image (512×404 pixels) yields 404 patches (cf. Figure 3.7), for which the similarities are computed based on method (ii). While for $c = e$, SDP relaxation segments the image into two reasonably coherent parts, using the number of pixels m_i contained in each patch i as constraint vector entries c_i gives a more balanced partitioning by separating the largest patch from the rest of the image.

all image patches are of equal importance no matter how large they are, the SDP relaxation method segments this image into two parts (both containing approximately the same number of patches) by cutting the city, which contains many small patches (see Figure 4.16, center).

To derive a segmentation which takes into account different patch sizes, the balancing constraint can be changed in the following way: calculate the number of pixels m_i contained in each patch i and set the constraint vector entries to $c_i = m_i$ instead of using $c_i = 1$. Thus we now search for a segmentation which

Fiedler vector (median threshold)

Figure 4.17: Segmentation for the image from Figure 4.16, top, based on the Fiedler vector: the median threshold yields a result similar to the segmentation obtained from the SDP relaxation.

partitions the image into two coherent parts, each containing approximately the same number of pixels instead of the same number of patches. The result depicted in Figure 4.16, bottom, approves the validity of this approach: now the sky is separated from the rest of the image, giving a segmentation in accordance with our new balancing constraint (but without enforcing it exactly). Note that for this example, the Fiedler vector also yields a quite meaningful partitioning, if it is thresholded at the median (see Figure 4.17); the best average cut again only separates one patch from the rest of the image.

Texture Images

The final experiment for binary partitioning tasks deals with grayscale images comprising some natural textures. An example is shown at the top of Figure 4.18. To derive a texture measure for this image, we subdivide it into 24×24-pixel windows, and calculate local histograms for two texture features within these windows. The corresponding graph is then obtained by identifying each window with a graph vertex, and by computing the similarity values w_{ij} based on the χ^2-distance of the histograms for all window pairs (i, j) directly, thus using method (i). Considering the simplicity of this texture measure, the segmentation result obtained with the SDP relaxation is excellent (see Figure 4.18, center). Based on the best average cut threshold, the Fiedler vector yields a different, but also satisfactory solution for this example (Figure 4.18, bottom).

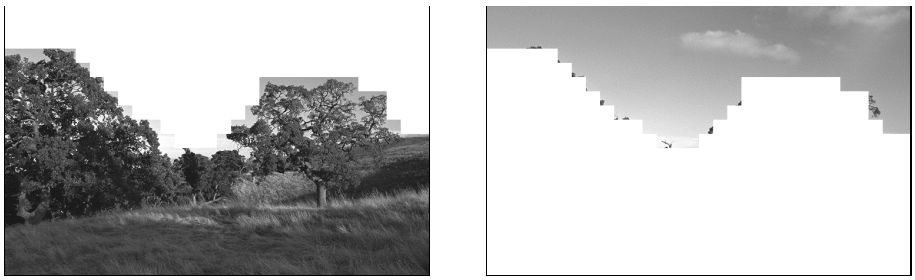
4.4.4 Perceptual Grouping

In this section, we study the application of the SDP relaxation approach to perceptual grouping problems (see Section 2.2). In this context, the corresponding version of the general combinatorial problem (4.5) does not have a balancing constraint, i.e. $c = 0$ and $\beta = 0$.

A first artificial grouping problem is depicted in Figure 4.19: several copies of a triangular object are hidden in a cluttered background. According to our knowledge about the relative angles between the edges of the object, we require two image primitives i and j to reinforce each other (by defining a high

Original image

SDP relaxation



Fiedler vector (best average cut)

Figure 4.18: Grayscale-texture partitioning. The input image (720×456 pixels) is subdivided into 570 windows containing 24×24 pixels each, and the similarity weights are computed based on direct texture comparison (method (i)). Both SDP and spectral relaxation result in convenient segmentations.

similarity value w_{ij}) if their enclosed angle is close to a multiple of $\frac{1}{3}\pi$. The suboptimal solution of the energy functional (2.8) computed for this example shows the success of the SDP relaxation: the structure is clearly separated from the background (Figure 4.19, right). Note that the presence of a small number of extra foreground primitives is not caused by the optimization approach; since these elements are consistent with the chosen similarity measure, they cannot be labeled as dissimilar.

Figure 4.20 shows an example of a perceptual grouping problem obtained from a real scene. In a first step, the input image is decomposed into a few hundred line fragments by applying an edge detector [50]. We again discriminate between figure and ground by defining a suitable similarity measure w_{ij}: two image primitives (line fragments) i and j are similar if the enclosed relative angle is close to a multiple of $\frac{\pi}{2}$, i.e. the lines are either orthogonal or parallel

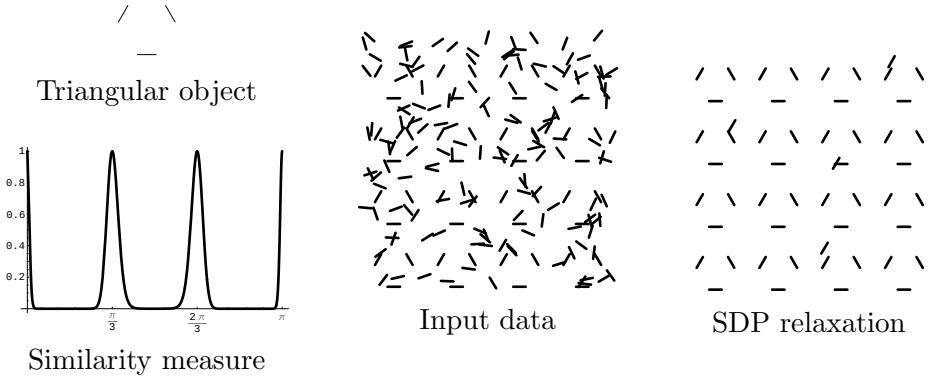


Figure 4.19: Perceptual grouping. An artificial triangular object (top left) is encoded by a similarity measure w_{ij} which favors relative angles between line fragments that are multiples of $\frac{1}{3}\pi$ (bottom left). Several copies of the object are superimposed by noise (center). The SDP relaxation of the functional (2.9) for $\lambda = 0.9$ successfully finds the objects as the largest coherent group of image primitives (right).

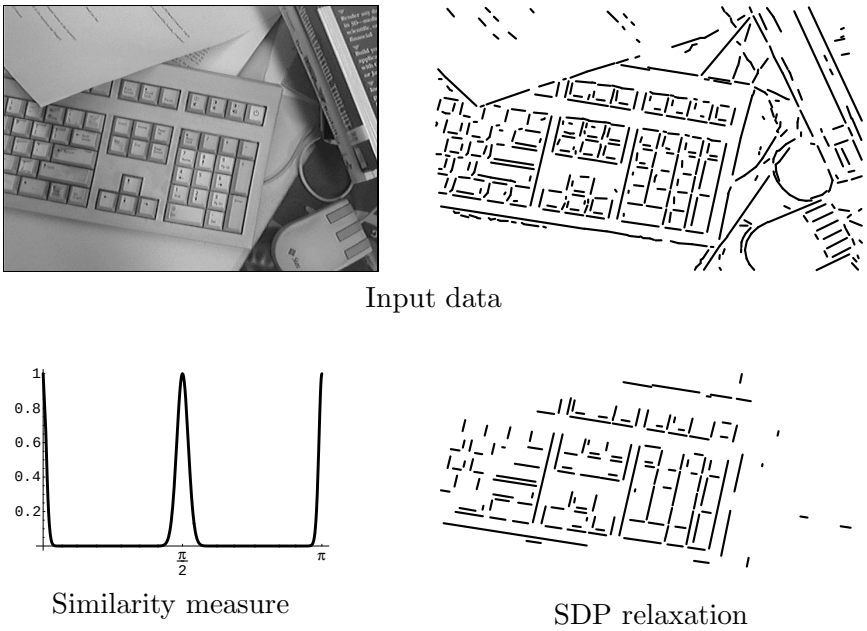
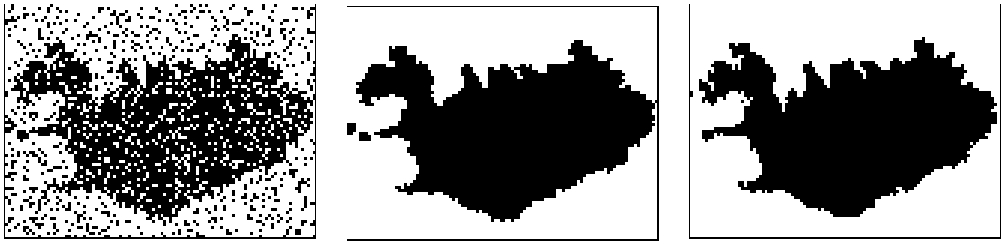
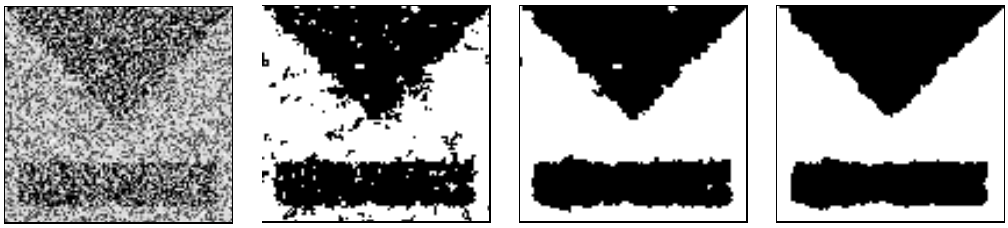


Figure 4.20: Perceptual grouping for a real world image. Line fragments within the input image are obtained by applying an edge detector (top). The similarity measure w_{ij} encodes preferred configurations as a function of the angle between two line fragments (bottom left): two fragments are most similar if they are (nearly) orthogonal or parallel to each other. Although several coherent groups of different cardinality exist in the input image, the SDP relaxation approach determines the largest coherent group of line fragments and suppresses the other groups (bottom right).



Noisy input image | SDP relaxation | Original image

Figure 4.21: Binary restoration. The original black and white map of Iceland (right, 104×78 pixels) has been degraded by adding binary salt and pepper noise (left). The restoration obtained based on the SDP relaxation of (2.12) for $\lambda = 2.0$ is fairly good (middle).



Noisy input image | SDP relaxation ($\lambda = 0.8$) | SDP relaxation ($\lambda = 1.5$) | SDP relaxation ($\lambda = 3.0$)

Figure 4.22: Binary restoration. Considering the poor signal-to-noise ratio of the input image (left, 98×93 pixels), the quality of the reconstruction for an appropriate choice of the smoothness parameter λ is very good (right).

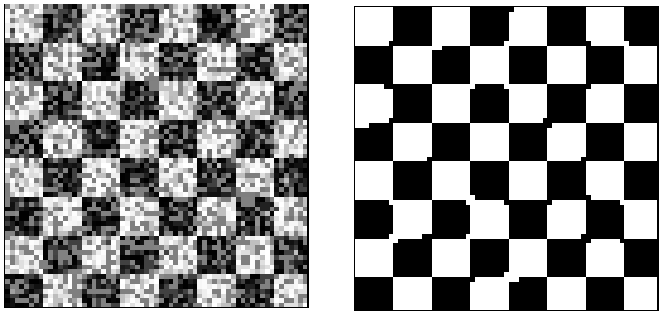
(cf. the graph of w in Figure 4.20, bottom left). We refer to [84] for more elaborate similarity measures which, however, are not essential for testing the SDP relaxation approach from the optimization point of view.

Note that according to the similarity measure w, *several* coherent groups exist for the example in Figure 4.20. Approximately minimizing (2.9) with the SDP relaxation method determines the keyboard as the "most coherent" group, as expected from a visual inspection of the scene (see Figure 4.20, bottom right).

4.4.5 Restoration

In Section 4.4.1, we already presented the results of the SDP relaxation approach with respect to the restoration of noisy one-dimensional signals. Figures 4.21–4.23 now depict reconstruction results for three different two-dimensional binary images that were degraded by noise. The pixel values for these examples are scaled to the interval $[-1, 1]$, so that the restoration prototypes correspond to $u_1 = -1$ (black) and $u_2 = +1$ (white), respectively. Only horizontally and vertically neighboring pixels are considered as adjacent (first-order neighborhood). Note that like for the grouping problem from the previous section, no balancing constraint is present, i.e. $c = 0$ and $\beta = 0$ in (4.5).

The overall quality of the restorations obtained from the SDP relaxation of the global objective function (2.11) is encouraging: for an appropriate choice



Noisy input image SDP relaxation

Figure 4.23: Binary restoration. For this checkerboard image (63×63 pixels), small errors in the reconstruction ($\lambda = 1.5$) only occur at corners where the local structure resembles noise.

of the parameter λ, we get smooth reconstructions despite a poor signal-to-noise ratio of the input image (Figure 4.22), and although the desired objects comprise structures at both large and small spatial scales (Figure 4.21). As expected, small errors may occur at sharp corners (Figure 4.23), which are difficult to distinguish from noise without prior knowledge. The influence of the scale parameter λ is illustrated in Figure 4.22: as desired, the restoration becomes smoother with increasing λ.

Finally, comparing these results with the restorations computed by the greedy ICM algorithm (see Figure 3.8 in Section 3.3) reveals the power of the global approach the SDP relaxation is based on: the stronger relaxation of the MAP estimate produces much better reconstructions.

4.4.6 Computational Complexity

The experimental results presented in the previous sections demonstrate that the SDP relaxation approach is a versatile tool for solving a broad range of difficult combinatorial problems conveniently. However, the price we have to pay for the favorable properties of this optimization approach (convexity, polynomial time solvability, no threshold; cf. Section 1.1) is the squared number of variables of the semidefinite programming relaxation. Although SDP solvers like the dual-scaling algorithm of Benson et al. [11, 12] are able to exploit a sparse problem structure (as it is encountered for the restoration problems), the memory requirements and the computation times quickly grow with the number of variables (see Table 4.1). While this is not a problem for perceptual grouping tasks with a couple of hundred primitives, it prevents at present the application to large-scale problems with several ten thousands of variables as they appear in image restoration or unsupervised image segmentation on a pixel basis. Especially the latter task soon becomes intractable since the matrices involved in the computations are usually dense.

In this context, several approaches from both the computational and the problem formulation point of view seem to be promising to mitigate the problem of dimensionality in the near future:

Image	n	time (in sec.)
1D signal restoration (Fig. 4.5)	256	3
spiral point sets (Fig. 4.10)	256	5
density point set (Fig. 4.11)	160	2
five clusters (Fig. 4.12)	250	4
hand/ball color image (Fig. 4.13)	1296	726
fence color image (Fig. 4.14)	211	3
city color image (Fig. 4.16)	404	17
texture image (Fig. 4.18)	570	59
triangle grouping (Fig. 4.19)	198	1
keyboard grouping (Fig. 4.20)	466	25
Iceland restoration (Fig. 4.21)	8113	13,320
arrow/bar restoration (Fig. 4.22)	9115	20,034
checkerboard restoration (Fig. 4.23)	3970	1,188

Table 4.1: Sizes and computation times for the different problems considered in this section (obtained with the dual-scaling algorithm from [11, 12] on a 3 GHz Pentium IV PC).

- Using Lemma 4.3, the maximal rank r of the primal solution matrix X of the SDP relaxation (4.7) can be bounded by $r < \sqrt{2n}$ for large problem sizes n (as the number of constraints equals $m = n + 1$ in (4.7)). This means that *in principle*, the number of n^2 problem variables in the primal of the SDP relaxation can be drastically reduced by setting $n - r$ rows of the matrix V in the decomposition $X = VV^\top$ to zero (cf. Section 4.2.3). In fact, first algorithms that exploit this property have already been published (e.g. [27]).

- For unsupervised partitioning tasks, a dense similarity matrix W can be made sparser by using a *cut-off radius* R, i.e. by setting the similarity of two points to zero if they are spatially farther apart than R units [168, 118]. In the extreme case, this results in defining edges only between directly neighboring pixels.

- For spectral relaxation methods, the problem size has been successfully reduced by applying *sampling methods* that only select a restricted number of pixels to represent the complete problem [48, 118, 54]. We will consider this possibility for the SDP relaxation approach in Section 5.3.

- Another alternative to reduce the size of unsupervised partitioning problems has already been applied in Section 4.4.3: the input is preprocessed to find image elements of larger scale which then replace the pixels as vertices in the corresponding graph representation. In the context of image segmentation, there has been an increasing interest in this idea recently [189, 151, 60]; therefore, we will also analyze this approach in more detail in Section 5.2.

Chapter 5

Efficient Unsupervised Segmentation

The results presented in the previous chapter indicate the usefulness of the SDP relaxation approach for image partitioning tasks. Concerning the problem size, however, they also reveal the limitations of this technique in practice. For this reason, we will examine methods in this chapter that enable the application of our SDP relaxation method to larger problem instances as they arise for real world partitioning tasks.

Specifically, we will investigate graph-based unsupervised image segmentation problems as they were introduced in Section 2.1. A general problem in this context concerns the size of the corresponding similarity matrix W. If the graph vertices represent the pixels of the image, the size of W increases quadratically with the number of pixels, and thus soon does no longer fit into memory completely (e.g. for an image of 481×321 pixels — the size of the images from the Berkeley segmentation dataset [121] — the similarity matrix contains $154401^2 \approx 23.8$ billion entries). Hence, any graph-based partitioning technique becomes computationally demanding (or even intractable) for larger problem instances.

A common approach to handle this issue is to revert to sparse similarity matrices by connecting pixels only within a certain neighborhood [168, 62, 204]. While spectral methods profit from this idea (since eigenvectors of sparse matrices can be calculated more efficiently), this is of no avail for our SDP relaxation: the solution matrix X of (4.7) usually is still dense even when the problem matrix L is sparse [12].

In this chapter, we therefore present two other methods which immensely reduce the size of unsupervised partitioning problems and thus make them feasible for the SDP relaxation approach. The basic idea of the first method has already been used in Section 4.4.3: based on computing an over-segmentation of the image in a preprocessing step, we abandon pixels as the basic image elements and use the obtained patches instead to form the graph representation (Section 5.2). In contrast to that, the second method reduces the problem size with a completely different technique that is based on a probabilistic sampling approach (Section 5.3). To derive more meaningful results, we first introduce

an obvious extension of the binary SDP relaxation method (Section 5.1): image segmentations into multiple parts can be obtained naturally by computing partitions recursively in a hierarchical way.

5.1 Hierarchical Segmentation

Up to now, we have only considered *binary* image partitioning problems. In practice, however, a segmentation into more than two parts may be more meaningful and is therefore often desired. For unsupervised clustering tasks, a corresponding straightforward extension consists in a *hierarchical application* of the binary approach: by recursively computing two-way partitions, we obtain a segmentation into multiple parts (see e.g. [98]). In the context of graph-based image segmentation, this idea leads to consecutive calculation of minimum cuts, for graphs of decreasing size. A general description of the hierarchical framework for computing a multipart segmentation is given by the following algorithm:

initialize: number of segments $k = 1$; first segment $S_1 = V$;
WHILE stopping criteria are not met DO
    select segment S_i ($i \in \{1, \ldots, k\}$) to split next;
    compute binary partitioning $S_i = S_{i,1} \cup S_{i,2}$ (using minimum cuts);
    $k = k + 1$; $S_i = S_{i,1}$; $S_k = S_{i,2}$;
END

For spectral partitioning techniques, successful applications of such a hierarchical framework have been presented e.g. in [198, 118]. In the subsequent sections of this chapter, we will also employ it to obtain multiclass segmentations based on our binary SDP relaxation approach. Yet first, we will briefly discuss the pros and cons of the hierarchical framework in the context of unsupervised image segmentation (Section 5.1.1), and make appropriate suggestions for the decision parameters involved in the partitioning process (Sections 5.1.2 and 5.1.3).

5.1.1 Hierarchical Partitioning vs. Direct Multiclass Clustering

Some authors argue against recursive bisection techniques, and instead favor partitioning a given graph into multiple clusters directly [29, 169, 173]. One mentioned reason is that a hierarchical approach requires the recalculation of the Laplacian and the minimization of the objective function for each step [29]. While this may be a drawback for real-time applications, it is only of minor interest for many image segmentation tasks, especially since the computational effort decreases with each level of the hierarchy.

The more important argument given against recursive bipartitioning is that the resulting clustering into k groups may not correspond to the optimal k-way segmentation [169, 173]. In fact, it is easy to provide examples that confirm this statement (see Figure 5.1). Yet this reasoning is slightly biased, since the hierarchical approach is not specifically designed for finding a "flat" partitioning

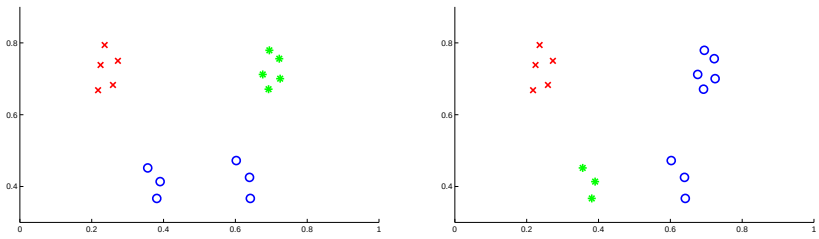


Figure 5.1: Example where recursive bipartitioning fails to find the optimal 3-way cut depicted in the left image. Hierarchical application e.g. of the binary average cut criterion produces a different solution for this simple example by first splitting the point set in the middle, and then separating top and bottom of one of the parts (right).

into a fixed number of groups, but rather tries to reveal information about the structural relationship between the different segments. If we know the number k of parts the image should be segmented into in advance, a direct k-way optimization criterion is certainly more suited to model the desired objective. In Section 6.2, we will consider according direct multiclass partitioning techniques.

For *unsupervised* image segmentation problems, however, the number of parts present in the image is usually *unknown*. In this case, the recursive bipartitioning framework represents the better alternative, since it allows the selection of a suitable number of segments during the partitioning process. Moreover, the subsequent splitting of segments yields a complete coarse-to-fine hierarchy of similar segmentations. This contrasts a direct multipartitioning approach, which may result in completely different segmentations for varying cluster-numbers k. Depending on the desired granularity of the final segmentation, we can therefore easily switch between different levels of the hierarchy. Furthermore, such a hierarchical representation of the image seems to be closer to human perceptual organization [121].

Besides these advantageous attributes, recursive image segmentation also introduces two additional decision parameters (cf. above algorithm): as no global objective function is optimized, criteria have to be defined in order to select the appropriate segment which should be partitioned next in each step, along with a suitable condition of when to stop the recursion. In the next two sections, we will discuss how such criteria can be derived based on the size of the current segments and the weights of the next potential cuts.

5.1.2 Which Segment to Split Next?

After each binary partitioning step, the question arises which of the segments obtained so far should be split next. Since the goal of unsupervised image segmentation tasks mainly consists in capturing the global impression of the scene, large parts of coherent structure are usually preferred to finer details. To obtain such a coarse-to-fine segmentation of the image, higher levels of the partition hierarchy have to be examined first. To this end, we in general select the largest existing segment (the one that contains most pixels) as the next candidate to be split.

Recalling that the binary SDP relaxation method is originally based on the graph-bisection criterion (2.6), which seeks a minimum cut through the graph subject to a balancing constraint, we may allow some exceptions to the above general selection rule:

- If the computed bisection for the candidate segment results in a high cut-value z_{sdp}, this indicates that the structure of this segment is already quite coherent. In this case, however, it should not be split any further and maintained as it is instead. To decide when a cut-value is too expensive, we compare it against the sum $d(V) = \sum_{i,j} w_{ij}$ of all edge-weights of the complete graph (which is an upper bound on the cut-value). Then splitting the candidate segment is only accepted if the corresponding cut-value z_{sdp} is smaller than a certain fraction $0 < a < 1$ of $d(V)$: $z_{sdp} < a\, d(V)$.

- Since it is not prohibited, the bipartition obtained from the SDP relaxation by randomized rounding may be the trivial solution of the underlying optimization problem, i.e. the non-cut solution that groups all points into one segment again. In this case, of course, this segment should no longer be considered to be split.

- Another exception may be defined in the case that the graph representation of the image is not based on the pixels directly, but on other image elements corresponding e.g. to patches obtained from an over-segmentation (see Section 5.2): if the largest (concerning the number of pixels) segment contains less than a certain number $p_{\min}$ of image elements, it should not be split any more. In this way, we prevent large patches from always being separated completely from the rest of the image.

5.1.3 Stopping Criteria

The probably most difficult question in connection to unsupervised image segmentation concerns the number of parts the image consists of, or alternatively, when to stop the hierarchical partitioning process. When asked to segment an image, every human is likely to give a different answer to this question. Hence, one can even argue that without defining the desired granularity, image segmentation becomes an ill-posed problem.

In this thesis, we consider two different stopping criteria for the hierarchical application of the SDP relaxation method that are both based on the desired granularity: whereas the first one directly defines a maximum number k of parts for the final segmentation, the second one is more sophisticated. Note that building the sum $z_{sum}(m) := \sum_{i=1}^{m} z_{sdp,i}$ of the cut-values $z_{sdp,i}$ calculated in each step i of the partitioning process results in an increasing function depending on the step number m, which is bounded above by the sum of all edge-weights $d(V)$ (cf. last section). Therefore, an additional stopping criterion is introduced by limiting the total cut-value $z_{sum}(m)$ to a certain fraction $q < 1$ of $d(V)$: $z_{sum}(m) < q\, d(V)$. We can then control the granularity of the final segmentation by adjusting the value of q appropriately.

5.2 Over-Segmentation with Mean Shift

One straightforward remedy to reduce the size of a graph-based image segmentation problem is to abandon pixels as graph vertices and to resort to other image representing elements instead. For example, the perceptual grouping task (Section 2.2) can be interpreted in this way: by using edge elements as basic image descriptors, a smaller graph representation is derived. In this section, we closer investigate an idea that was already introduced in Section 4.4.3: an over-segmentation of the image is computed in a preprocessing step by applying a clustering technique at a fine spatial scale. Instead of having to deal with thousands of pixels, the graph representation of the image is then based on the obtained image patches (or "superpixels") of coherent structure. Actually, this is also a more natural image representation, since the real world does not consist of pixels — those are merely the result of the digital image discretization process.

While different preprocessing methods have been proposed in the literature in this context [198, 118, 189, 8, 151], we suggest to use the mean shift technique presented in Section 3.2 for this purpose. After briefly reviewing the main aspects of the corresponding preprocessing step (Section 5.2.1), we describe how to obtain an appropriate graph representation based on the computed image patches (Section 5.2.2), and how suitable balancing constraints for our SDP relaxation method can be selected automatically (Section 5.2.3). Several results for large real world images are finally presented in Section 5.2.4.

5.2.1 Preprocessing Step

In order to adequately reduce the problem size without destroying any perceptually significant structure, we apply the mean shift technique [33] at a fine spatial scale. To this end, each pixel is represented by a feature vector comprising its position and its color in the perceptually uniform L*u*v* space. The mean shift algorithm then forms clusters of similar feature vectors based on their Euclidean distances in the corresponding 5-dimensional feature space.

The final number and the size of the image patches obtained in this way are controlled by three parameters: the spatial and the range bandwidth σ_s and σ_r to scale the entries of the feature vectors, and the minimum cluster size M (see Section 3.2). We adjust these parameters manually or semi-automatically in order to obtain a final clustering into 100–700 image patches (corresponding to less than 0.01% of the complete number of pixels for images from the Berkeley segmentation dataset [121], for instance), which is a suitable problem size to be processed efficiently with our SDP relaxation approach. In general, for the examples considered in this section, we derive such a number of patches by fixing $M = 50$ as a reasonable minimum region size, and setting $\sigma_s = 5.0$ (as an adequate fraction of the image size) and $\sigma_r = \frac{d_{\max}}{15}$, where $d_{\max}$ denotes the maximum distance in L*u*v* color space found for a random sample set of 200 pixels.

Figure 5.2 shows an example of the image patches obtained with the mean shift algorithm — more examples were already given in Figure 3.7. They reveal

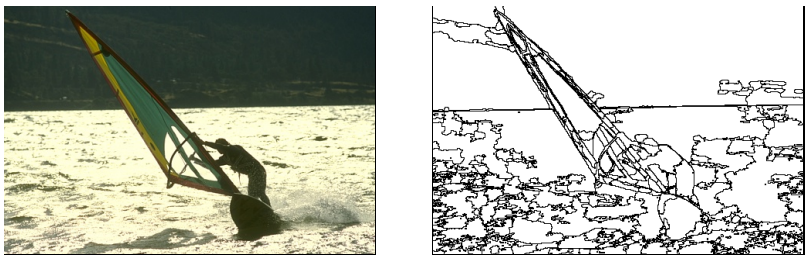


Figure 5.2: Over-segmentation with the mean shift technique. For this sample image from the Berkeley segmentation dataset [121], 304 image patches are obtained. Note that in accordance with the homogeneous regions of the image, the patches differ in size.

an important quality of the mean shift over-segmentation: when large homogeneous regions exist in the image, they are also represented by large patches instead of being split artificially. This results in superpixels of considerably varying size. As will be shown in Section 5.2.3, our SDP relaxation approach can take this aspect into account appropriately. In contrast to this, spectral techniques usually require patches of similar size. Moreover, the splitting of such homogeneous regions during the following partitioning process (something spectral techniques tend to do) is effectively prevented.

5.2.2 Constructing the Graph

Basically, a graph representation of the image can be constructed by associating each image patch i obtained in the preprocessing step with a graph vertex, and computing the similarity matrix based on the distances of the corresponding mean colors y_i in L*u*v* space. In order to additionally take into consideration the *spatial distances* between the image patches, we already indicated several different methods in Section 4.4.2. The simplest approach consists in constructing a locally connected graph with weighted edges being defined only between neighboring patches based on their color differences. From this representation we may obtain a fully connected graph by, e.g., calculating shortest paths between all vertices (cf. method (ii) in Section 4.4.2). Alternatively, such a fully connected graph can also be derived directly by using the spatial distance between two patches as an additional cue in the calculation of their similarity.

Yet due to the varying size and shape of the image patches, it is hard to define an appropriate spatial distance measure directly: for instance, the spatial barycenter of a patch may lie outside its region, which may result in completely misleading spatial distances. For this reason, we connect only directly neighboring image patches. In contrast to the experiments in Section 4.4.3, we also abandon the time consuming calculation of shortest paths between all vertices, especially since experiments indicated that a fully connected graph does not result in better segmentations. However, we slightly modify the calculation of the similarity values w_{ij} in order to take into consideration the varying boundary lengths l_{ij} between two neighboring image patches i and j:

$$w_{ij} = l_{ij} e^{-\frac{\|y_i - y_j\|}{\sigma_r}} \,, \tag{5.1}$$

where y_i denotes the mean color of patch i, and the normalization parameter σ_r corresponds to the range bandwidth computed for the mean shift (see previous section).

The multiplication with l_{ij} in (5.1) simulates a standard coarsening technique for graph partitioning [118]: the edge-weights between two neighboring clusters are calculated as the sum of the edge-weights of the vertices which were grouped together within these clusters. Assuming a first-order neighborhood (four connections for each pixel inside the image) of the original graph, this is equivalent to adding the weights along the boundary between two patches. As each image patch contains pixels of similar color, the exact color of the boundary pixels can be replaced with the mean color y_i of the patch without considerably changing the resulting weight.

Note that additional cues like texture or intervening contours can be incorporated into the classification process by computing corresponding similarity values based on the image patches, and combining them appropriately (see e.g. [118, 165, 120]). However, we did not consider modified similarities values here.

5.2.3 Balancing Constraint Selection

Remember that our SDP relaxation approach is based on the unsupervised segmentation problem (2.6), which involves a balancing constraint $c^\top x = \beta$ that can be set application-dependent. In this context, the classical graph (equi-)partitioning approach uses $c = e$ (the vector of all ones) and $\beta = 0$ to obtain a balanced clustering. While this is reasonable for graphs where each vertex is equally important, the vertices now correspond to image patches of varying size, which suggests different weights. As we already argued in Section 4.4.3, a more appropriate balancing constraint in this case is derived by counting the number of pixels m_i contained in each image patch i, and setting the entries of the constraint vector to $c_i = m_i$ instead of to $c_i = 1$ (while keeping $\beta = 0$). In this way, our approach seeks two coherent parts in the image with each containing approximately the same number of pixels.

If the SDP relaxation method is applied within the hierarchical framework presented in Section 5.1, we may get into the situation where the current part of the image contains a dominating patch k which is much larger than the other patches in this part of the image: $c_k = \max_i c_i \gg c_j$ for all $j \neq k$. In this case, a segmentation into two parts of the same size may no longer be possible. In fact, due to Theorem 4.4, the SDP relaxation (4.7) has a feasible solution only if the constraint vector c in combination with β is balanced: $|c_i| \leq |\beta| + \sum_{j \neq i} |c_j|$ for all $i = 1, \ldots, n$, and $|\beta| \leq \sum_j |c_j|$. Yet if this is not the case, this theorem also reveals that a feasible instance of the SDP relaxation can be derived by adjusting the value of β appropriately: as β corresponds to the desired difference between the size of both parts of a segmentation, setting $\beta = c_k - \sum_{i \neq k} c_i$ makes (4.7) feasible. Since this constraint strictly requires to separate the largest patch k from the rest of the patches, we enlarge β by a fraction of the number of pixels contained in the small remaining patches, $\beta = c_k - \frac{1}{2} \sum_{i \neq k} c_i$, to allow for more flexibility in practice.

Finally, note that such adjustments are not admissible for spectral relaxation

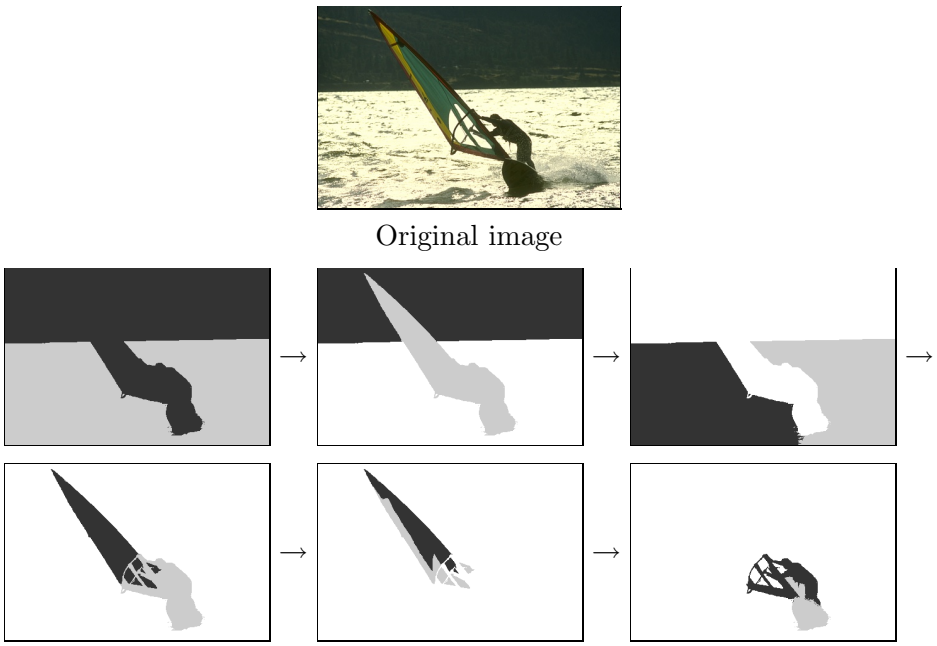


Figure 5.3: Evolution of a hierarchical segmentation based on the SDP relaxation approach. In each step, the parts indicated in gray and black are separated. Note the coarse to fine nature of the evolution: the broad parts of the image are segmented first (middle row), whereas the finer details arise later (bottom row).

methods: while the Fiedler vector (3.13) always requires balancing the number of patches, the normalized cut relaxation (3.16) balances the degrees of the patches. Hence, neither can the number of pixels contained in a patch be incorporated, nor is it possible to influence the gap between both parts of the solution *before* the thresholding step.

5.2.4 Experimental Results

To evaluate the performance of the over-segmentation-based SDP relaxation method for large real world scenes, we apply it to images from the Berkeley segmentation dataset [121] within the hierarchical framework presented in Section 5.1. As a first example, Figure 5.3 shows how the final segmentation obtained by SDP relaxation evolves hierarchically. One can clearly see the coarse to fine advancement of the hierarchical segmentation: the first steps segment the broad parts like the sky and the water, whereas the finer details of the surfer are partitioned later. Also note that although the water contains many patches (cf. Figure 5.2), its further splitting is effectively prevented within the hierarchical framework since the the corresponding cut-values are too high (see Section 5.1.2).

Figure 5.4 depicts the encouraging final segmentations for several different images, which are computed with the hierarchical SDP relaxation method in less than 5 minutes on a 2 GHz Pentium IV PC. Concerning the parameters introduced in connection with the hierarchical framework in Sections 5.1.2 and

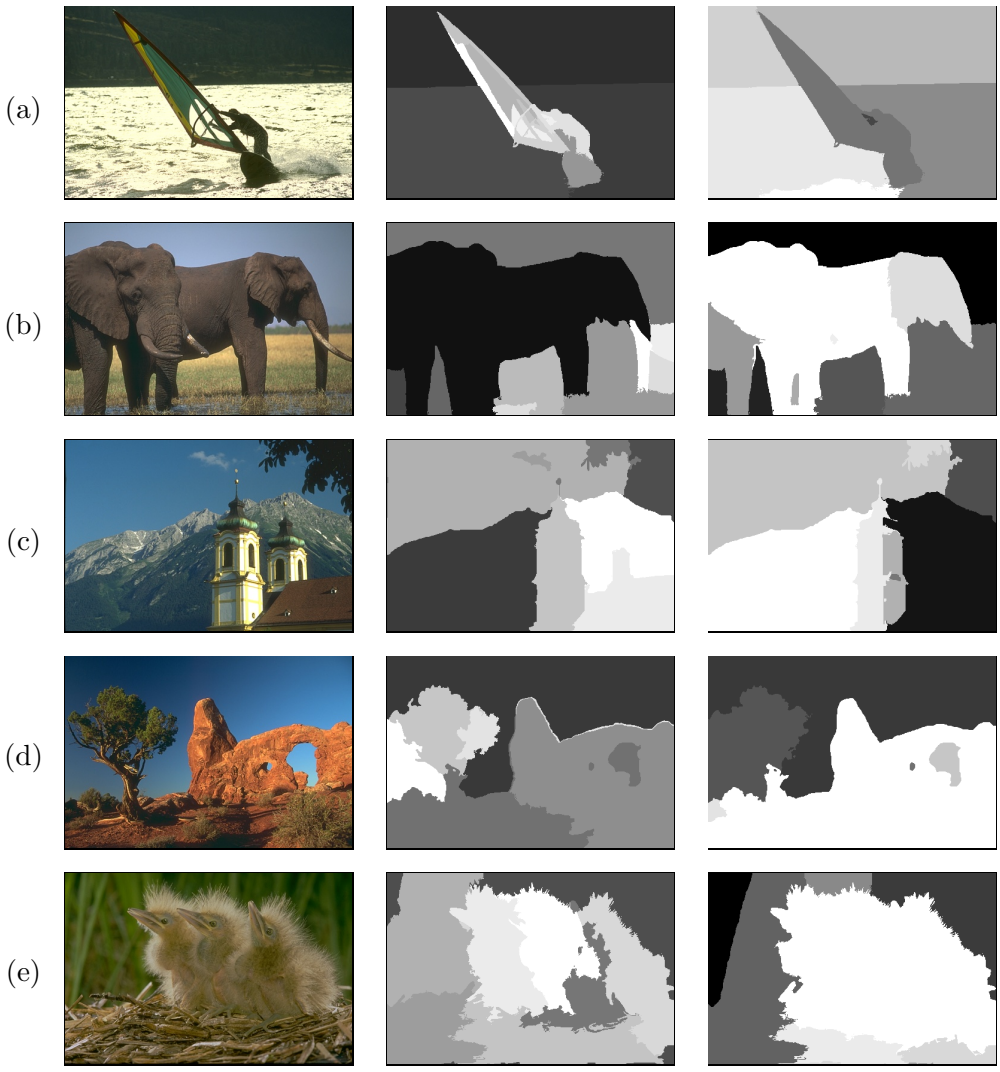


Figure 5.4: Hierarchical segmentation results for five color images of size 481×321 pixels from the Berkeley segmentation dataset [121]. Note the superior quality of the segmentations obtained with the SDP relaxation approach in comparison to the normalized cut relaxation, which are approved by the higher F-measures given in Table 5.1

5.1.3, we set $a = 0.02$ (as the fraction of the sum of all edge-weights) for the maximally allowed cut-value, and $p_{\min} = 7$ as the minimum number of patches that must be contained in a candidate-segment. Since appropriate values for the parameter q controlling the granularity turned out to differ from image to image, we simply set the maximum number of parts to 10 and picked the final segmentation from the hierarchy by visual inspection. The automatic determination of a suitable stopping criterion thus remains an open point for future research.

For comparison, we also compute the corresponding normalized cut segmen-

Image	# Segments	F-measure SDP	Ncut
(a)	7	0.92	0.77
(b)	9	0.76	0.73
(c)	8	0.69	0.64
(d)	6	0.68	0.61
(e)	8	0.58	0.35

Table 5.1: Quantitative performance of the SDP relaxation. The high F-measures obtained for the images in Figure 5.4 confirm the encouraging results in comparison to the normalized cut relaxation. Also given is the corresponding final number of segments picked from the partitioning hierarchy.

tations that contain the same number of final segments as the SDP solutions. To this end, we use the hierarchical framework with identical parameter settings, and only replace the binary partitioning technique. The results given in Figure 5.4 indicate the superiority of the SDP relaxation method: especially note that for some images, the normalized cut relaxation yields quite small segments instead of producing balanced partitionings. However, this apples-to-apples comparison should be judged with care: as the normalized cut relaxation cannot take into account the varying size of the image patches appropriately, the over-segmentation produced with mean shift possibly is not an adequate starting point for this method.

Since the Berkeley segmentation dataset also provides "ground-truth" data in the form of segmentations produced by humans, we are able to measure the performance of our SDP relaxation algorithm quantitatively by comparing the results with the corresponding human segmentations. To this end, we use the precision-recall framework presented in [120], which is a standard method in the information retrieval community [182]. Based on the boundaries of the segments, the precision p measures the fraction of pixel-pairs that are correctly grouped together in comparison to the human segmentations, while recall r is the fraction of pixel-pairs from a ground-truth segment that are accurately detected by the partitioning algorithm. The F-measure then captures the trade-off between accuracy and noise by combining precision and recall as their weighted harmonic mean: $F = \frac{pr}{\gamma p+(1-\gamma)r}$. As in [120], we use $\gamma = 0.5$. Resulting in a value between 0 (corresponding to bad segmentations) and 1 (good segmentations), the F-measure is a valuable statistical indicator for the performance of a partitioning algorithm. The F-measures obtained for the final segmentations from Figure 5.4 are given in Table 5.1: they confirm the visual impression that the results of the SDP relaxation method outperform the normalized cut segmentations.

5.3 Probabilistic Sampling

An alternative approach to reduce the computational effort for solving graph-based image segmentation problems is based on probabilistic sampling of the

input data: by picking only a small random subset of pixels, a graph partitioning problem of small scale is obtained for which the solution can be calculated efficiently. Due to the fact that the number of coherent parts in an image is typically much smaller than the number of pixels, this solution usually can be generalized well to a solution of the full original problem.

The mathematical foundation for this approach is derived from fast matrix approximation methods that rely on probabilistic sampling techniques [56, 1]. Basically, these methods find good low-rank approximations (concerning the spectral structure) of a given matrix M efficiently by sampling the entries of M. In the context of graph-based image segmentation, this idea amounts to closely approximating the symmetric problem matrix $M \in \mathcal{S}^n$ (which is derived from the similarity matrix W) with a matrix $\hat{M}$ of considerably lower rank $k \ll n$ by randomly selecting a small number of pixels (represented by the corresponding rows or columns of M) from the image.

One example of such a probabilistic matrix approximation approach is the Nyström method, which originates from the numerical treatment of integral equations (see, e.g., [7]), and which recently has been applied successfully in connection with normalized cut relaxations of different grouping problems [54] and for machine learning tasks [171, 191]. As a natural alternative, we present a *probabilistic SVD approximation method* in Section 5.3.2, which has been introduced by Drineas et al. [48] in a different clustering context. Before the relation of this approach to the Nyström method is discussed (Section 5.3.3), we briefly address the topic of how to pick the sample points appropriately (Section 5.3.1). In Section 5.3.4, we then describe how the probabilistic SVD approximation method can be applied to solve unsupervised partitioning problems based on both the normalized cut and the SDP relaxation. Finally, the performance of these probabilistic approaches is evaluated experimentally by considering statistical results for ground-truth experiments on bipartitioning point sets as well as hierarchical segmentations of large real world images (Section 5.3.5).

5.3.1 Sample Selection

The success of any partitioning technique that is based on probabilistic matrix approximation certainly depends on an appropriate sample selection procedure, i.e. to pick suitable points (which are represented by columns of the problem matrix M) that provide enough information to closely approximate the complete problem. To this end, Drineas et al. [48] propose to sample points from the input data with probabilities that are proportional to the squared norm of the corresponding columns of M. For the rank-k matrix $\hat{M}$ obtained based on such a selection process, they are able to prove a theoretical bound on the approximation quality which yet requires a large number of samples to be selected. On the other hand, they also state that in practice it suffices to pick a much smaller number of samples to obtain good approximation results.

If the problem matrix M is *dense*, Frieze et al. [56] argue that a column M_i of M (representing the point i) can be selected with probability $\Pr[i] = \frac{1}{n}$ in order to fulfill the requirements of the proven approximation bound. Since this corresponds to a uniform distribution of the input data, we can pick the desired

number s of samples independently at random from the complete set of points. In connection with the Nyström method, Fowlkes et al. [54] successfully use the same random selection procedure for picking columns of their dense problem matrices. Note that besides its simplicity, this sample selection procedure has the enormous advantage that it does not require to calculate the complete matrix M to decide which points to pick — in contrast to the approach of Drineas et al. [48] mentioned above. Due to these facts, we will only consider dense problem matrices in this section, and also rely on the aforementioned, simple random procedure to select the sample points.

To facilitate the analysis of the following sections, we assume that the pixels are rearranged so that the s selected sample points precede the remaining points of the image. The corresponding (symmetric) problem matrix $M \in \mathcal{S}^n$ can then be partitioned into smaller submatrices:

$$M = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix} ,$$

with $A \in \mathbb{R}^{s\times s}$, $B \in \mathbb{R}^{s\times n-s}$ and $C \in \mathbb{R}^{n-s\times n-s}$. This rearrangement does not change the spectral structure of the matrix.

5.3.2 Probabilistic SVD Approximation

The probabilistic matrix approximation technique we present in this section is based on the following well known fact from linear algebra [69] about the singular value decomposition (SVD) of a matrix:

Lemma 5.1. *Let the SVD of the symmetric matrix $M \in \mathcal{S}^n$ be given by*

$$M = \sum_{i=1}^{n} \sigma_i q_i p_i^\top ,$$

and denote by $Q_k \in \mathbb{R}^{n\times k}$ the matrix comprising the left orthonormal singular vectors q_i for the k largest singular values $\sigma_1 \geq \cdots \geq \sigma_k$ of M. Then the best rank-k approximation to M within a suitable matrix norm[1] is given by

$$\min_{\substack{D\in\mathcal{S}^n \\ \operatorname{rank}(D)=k}} \|M - D\| = \|M - Q_k Q_k^\top M\| . \tag{5.2}$$

Based on (5.2), Drineas et al. [48] propose to approximate the top k left singular vectors q_i of the complete matrix M by calculating the top k left singular vectors $\hat{q}_i$ of the sampled $n \times s$ submatrix[2]

$$S = \begin{pmatrix} A \\ B^\top \end{pmatrix} . \tag{5.3}$$

[1] More specifically, the approximation (5.2) holds for every unitarily invariant matrix norm (cf. [175]), which includes the spectral norm $\|.\|_2$ and the Frobenius norm $\|.\|_\mathrm{F}$.

[2] More precisely, Drineas et al. [48] use a weighted submatrix $S' := SR^{-\frac{1}{2}}$ instead of S, where R is a diagonal matrix with the entries $R_{ii} = s \Pr[i]$ that weights the columns of S with their selection probabilities. However, since we use uniform sampling (see Section 5.3.1), we get $R = \frac{s}{n} I$, which yields $S' = \sqrt{\frac{n}{s}} S$ and thus is equivalent to employing S directly.

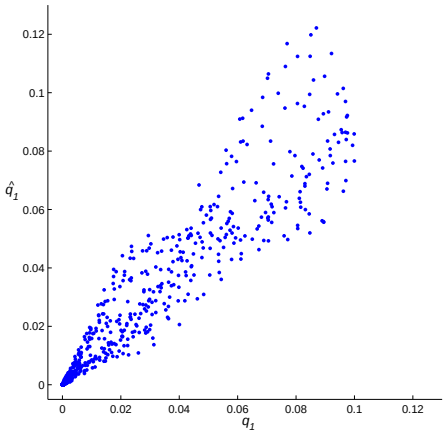


Figure 5.5: Visualization of the approximation quality obtained with the probabilistic SVD method. For the similarity matrix W of the clustering problem depicted in Figure 5.7, the top eigenvector q_1 is plotted against the corresponding approximation $\hat{q}_1$ based on sampling 15% of the points.

As can be seen easily [48], this can be accomplished efficiently by computing the eigenvectors y_i corresponding to the k largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_k \geq 0$ of the much smaller, positive semidefinite matrix $S^\top S \in \mathcal{S}_+^s$, and calculating

$$\hat{q}_i = \frac{Sy_i}{\|Sy_i\|} = \frac{Sy_i}{\sqrt{\lambda_i}} \quad \text{for } i = 1, \ldots, k \; .$$

Hence, this probabilistic SVD approximation method yields an efficient way to find a matrix $\hat{M}$ which approximates the best rank-k approximation to M from (5.2). Putting everything together, we obtain the following relations:

$$M \approx Q_k Q_k^\top M \approx \hat{Q}_k \hat{Q}_k^\top M = \hat{M} \; ,$$

where $\hat{Q}_k \in \mathbb{R}^{n \times k}$ again denotes the matrix containing the (orthonormal) vectors $\hat{q}_i$ as columns.

Under specific conditions, a theoretical bound on the approximation quality of $\hat{M}$ can be proven [56, 48]. Since however, this bound requires large sampling rates, it is of no practical value in the context of efficient image segmentation, and is therefore not considered here. Instead, Figure 5.5 gives a first impression of the performance of the probabilistic SVD method: for the similarity matrix W of a clustering problem that will be considered in Section 5.3.5 (cf. Figure 5.7), the entries of the eigenvector q_1 belonging to the largest eigenvalue are plotted against the corresponding entries of the approximation $\hat{q}_1$. Although only 15% of the points are sampled, the approximation quality is quite good.

A different interpretation of the SVD approximation method is derived by observing that the left singular vectors $\hat{q}_i$ of the sampled submatrix S equal the top singular vectors (or equivalently the top eigenvectors) of the symmetric, positive semidefinite matrix $SS^\top \in \mathcal{S}_+^n$, and that the same holds for the matrices M and $MM^\top$ (cf. Lemma A.2). Using these facts, we see that the above method also corresponds to approximating the matrix $MM^\top$ with the matrix $SS^\top$ of

smaller rank s in the following way:

$$\begin{aligned} MM^\top &= \begin{pmatrix} AA^\top + BB^\top & AB + BC^\top \\ (AB)^\top + (BC)^\top & B^\top B + CC^\top \end{pmatrix} \\ &\approx \begin{pmatrix} AA^\top & AB \\ (AB)^\top & B^\top B \end{pmatrix} \\ &= \begin{pmatrix} A \\ B^\top \end{pmatrix} \begin{pmatrix} A^\top & B \end{pmatrix} = SS^\top \\ &= \hat{Q}_s \Sigma \hat{Q}_s^\top \ , \end{aligned} \tag{5.4}$$

where the thin SVD[3] $\hat{Q}_s \Sigma \hat{Q}_s^\top$ of $SS^\top$ is obtained from the eigenvalue decomposition $Y \Sigma Y^\top$ of the positive semidefinite and much smaller matrix $S^\top S \in \mathcal{S}_+^s$ by setting

$$\hat{Q}_s := S Y \Sigma^{-\frac{1}{2}} \ . \tag{5.5}$$

Here, $\Sigma := \mathrm{Diag}(\lambda_1, \ldots, \lambda_s)$ denotes the diagonal matrix containing the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_s \geq 0$ of $S^\top S$ (which are equal to the top singular values of $SS^\top$) on the diagonal in descending order, and the orthogonal matrix $Y \in \mathbb{R}^{s \times s}$ contains the corresponding eigenvectors y_i as columns.

5.3.3 Comparison to the Nyström Method

Recently, a different sampling-based, efficient matrix approximation technique has been proposed in the context of spectral grouping [54] and machine learning [191], which is derived from the so-called Nyström extension [7]. Based on the sampled submatrix S from (5.3), the basic idea of this approach is to directly approximate the problem matrix M with a rank-s matrix $\hat{M}$ by implicitly approximating the submatrix C of M with the matrix $B^\top A^{-1} B$ (we assume for the moment that the submatrix A is positive definite):

$$\begin{aligned} M &= \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix} \\ &\approx \begin{pmatrix} A & B \\ B^\top & B^\top A^{-1} B \end{pmatrix} \\ &= \begin{pmatrix} A \\ B^\top \end{pmatrix} A^{-1} \begin{pmatrix} A^\top & B \end{pmatrix} = S A^{-1} S^\top \\ &= \hat{P}_s \Lambda \hat{P}_s^\top \ . \end{aligned} \tag{5.6}$$

The thin SVD $\hat{P}_s \Lambda \hat{P}_s^\top$ of the approximating matrix $\hat{M} := S A^{-1} S^\top$ again can be calculated efficiently by computing the eigenvalue decomposition $Z \Lambda Z^\top$ of the smaller matrix

$$A' := A + A^{-\frac{1}{2}} B B^\top A^{-\frac{1}{2}} = A^{-\frac{1}{2}} S^\top S A^{-\frac{1}{2}} \in \mathcal{S}_+^s$$

and setting

$$\hat{P}_s := S A^{-\frac{1}{2}} Z \Lambda^{-\frac{1}{2}} \ . \tag{5.7}$$

[3] In this context, thin SVD means the SVD of a matrix without the singular vectors corresponding to the zero singular values.

This time, $\Lambda := \mathrm{Diag}(\lambda_1, \ldots, \lambda_s)$ denotes the diagonal matrix containing the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_s \geq 0$ of A' on the diagonal in descending order, and the orthogonal matrix $Z \in \mathbb{R}^{s \times s}$ comprises the corresponding eigenvectors as columns. In practice, only the top k eigenvectors of the matrix A' are calculated, which leads to the rank-k approximation $\hat{P}_k \Lambda_k \hat{P}_k^\top$ of M.

Comparing the approximations (5.4) and (5.6) reveals the similarity of the Nyström method and the probabilistic SVD approximation: whereas the latter approach calculates the eigenvectors $\hat{q}_i$ of the matrix $SS^\top$ as an approximation to the top eigenvectors of $MM^\top$ (which are the same as the top eigenvectors of M if the matrix M is positive semidefinite), the Nyström method approximates the top eigenvectors of M by computing the eigenvectors $\hat{p}_i$ of the matrix $SA^{-1}S^\top$. Hence, for positive semidefinite matrices $M \in \mathcal{S}_+^n$, both approximations will become very similar if the submatrix A resembles the identity matrix, $A \approx I$. In fact, both approaches can be considered as special cases of the general low-rank matrix approximation method presented by Frieze et al. [56] for rectangular matrices [191].

Interestingly, both methods maintain the inner products of the columns of the sample matrix S after projection onto the subspaces spanned by the corresponding approximative eigenvectors. For the probabilistic SVD approximation, we can verify this by using the definition (5.5) of $\hat{Q}_s$ in connection with the eigenvalue decomposition $S^\top S = Y \Sigma Y^\top$ with $Y^\top Y = I$:

$$\begin{aligned} S^\top \hat{Q}_s \hat{Q}_s^\top S &= S^\top S Y \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} Y^\top S^\top S \\ &= Y \Sigma Y^\top Y \Sigma^{-1} Y^\top Y \Sigma Y^\top \\ &= Y \Sigma Y^\top = S^\top S \,. \end{aligned}$$

Similarly, using the definition (5.7) of $\hat{P}_s$ in connection with the fact $S^\top S A^{-\frac{1}{2}} = A^{\frac{1}{2}} Z \Lambda Z^\top$ derived from the eigenvalue decomposition of A', we get for the Nyström method:

$$\begin{aligned} S^\top \hat{P}_s \hat{P}_s^\top S &= S^\top S A^{-\frac{1}{2}} Z \Lambda^{-\frac{1}{2}} \Lambda^{-\frac{1}{2}} Z^\top (A^{-\frac{1}{2}})^\top S^\top S \\ &= A^{\frac{1}{2}} Z \Lambda Z^\top Z \Lambda^{-1} Z^\top Z \Lambda Z^\top (A^{\frac{1}{2}})^\top \\ &= A^{\frac{1}{2}} Z \Lambda Z^\top A^{\frac{1}{2}} = S^\top S \,. \end{aligned}$$

Concerning the applicability of the Nyström method, we had to assume (as opposed to the SDP approximation method) that the submatrix A is positive definite in order to assure that the inverse A^{-1} and the square root $A^{\frac{1}{2}}$ exist. In fact, the second requirement is already fulfilled if the matrix M is positive semidefinite: in this case, A is also positive semidefinite, which guarantees the existence of the square root $A^{\frac{1}{2}}$. Yet the inverse A^{-1} may not be calculated if any of the eigenvalues of A is 0. As a remedy for the indefinite case, Fowlkes et al. [54] propose a modification of the Nyström method which utilizes the pseudoinverse instead of A^{-1}. However, besides increasing the computational effort, this modification may lead to a significant loss in numerical precision, and thus should only be applied when necessary [54].

In contrast to that, the probabilistic SVD approximation method does not involve the calculation of any square roots or inverse matrices (except for $\Sigma^{-\frac{1}{2}}$, which is unproblematic since Σ is diagonal and positive definite). Hence, this approach is more convenient concerning the computational complexity, and less sensitive when being applied to nearly singular matrices. Indeed, it can also be used to calculate rank-k approximations for non-positive semidefinite matrices. In this case, however, one has to be cautious when applying the SVD approximation method to spectral partitioning approaches which are based on the largest eigenvectors of the problem matrix: as the largest singular vectors could also correspond to eigenvectors of negative eigenvalues (cf. Lemma A.2), they may yield incorrect partitionings.

5.3.4 Application to Binary Partitioning

In this section, we describe how the probabilistic SVD approximation method presented in Section 5.3.2 can be applied to solve binary partitioning problems based on the SDP relaxation approach (see Section 4.2) and the normalized cut relaxation (see Section 3.1.4), respectively. In this context, first note that the problem formulations corresponding to both partitioning techniques involve *minimizing* the objective function, whereas the probabilistic SVD method approximates the singular vectors belonging to the *largest* singular values. However, since $x^\top Lx = x^\top(D-W)x = -x^\top Wx + d(V)$ for $x \in \{-1,+1\}^n$, we can directly transform the objective for both problem formulations to maximization by substituting the Laplacian L with the similarity matrix W. For the SDP relaxation (4.7), this results in the equivalent problem

$$\begin{aligned} \max_{X \succeq 0} \quad & W \bullet X \\ \text{s.t.} \quad & cc^\top \bullet X = \beta^2 \\ & \operatorname{diag}(X) = e \; . \end{aligned} \tag{5.8}$$

More specifically, since we are only concerned with image partitioning problems on a *pixel basis* in this section, the corresponding graph vertices can be assumed to be of equal importance. For this reason, we generally set $c = e$ and $\beta = 0$ in the balancing constraint in (5.8).

On the other hand, the above substitution yields for the normalized cut relaxation (3.16):

$$\begin{aligned} \max_{z \in \mathbb{R}^n} \quad & \frac{z^\top Wz}{z^\top Dz} \\ \text{s.t.} \quad & z^\top De = 0 \; , \end{aligned}$$

which corresponds to computing the eigenvector belonging to the second largest eigenvalue of the normalized similarity matrix $W' = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ (cf. Lemma 3.4).

Moreover, we assume in this section that the similarity matrix W is positive semidefinite in order to guarantee the equivalence of its largest singular vectors and its largest eigenvectors (see Lemma A.2). In general, this is no restriction

since an indefinite matrix W can always be changed into a positive semidefinite matrix by adding a multiple of the identity matrix, $\tilde{W} := W + \gamma I$, with $\gamma \in \mathbb{R}^+$ chosen large enough: this transformation increases the eigenvalues of W by γ, but does not change the eigenvectors or the order of the eigenvalues.[4] Since experiments indicate that the positive eigenvalues are mostly dominating for real image data, this transformation yet is usually not necessary in practice.

SDP Relaxation

To solve the SDP relaxation (4.7) by means of the probabilistic SVD approximation method, we use the fact that the randomized hyperplane technique (see Section 4.2.3) calculates an integer solution $x \in \{-1, +1\}^n$ based on the Cholesky decomposition $X^* = VV^\top$ of the solution matrix X^* of (4.7). This results in the equivalent problem formulation (4.9) with the objective function $\mathrm{Tr}(V^\top LV)$, where the second constraint ensures that the rows v_i of V have unit norm. If we disregard the balancing constraint in (4.9), the complete eigenvalue decomposition $W = Q\Lambda Q^\top$ of the similarity matrix W yields a special instance of the SDP relaxation in the maximization form (5.8),

$$\max_{\substack{X \succeq 0 \\ \mathrm{diag}\, X = e}} W \bullet X = \max_{\substack{V \in \mathbb{R}^{n \times n} \\ \|v_i\|^2 = 1}} \mathrm{Tr}(V^\top WV) \geq \mathrm{Tr}(Q^\top WQ) = \sum_{i=1}^{n} \lambda_i \; ,$$

since $QQ^\top = Q^\top Q = I$. For this reason, we suggest to calculate an approximative Cholesky decomposition of the solution matrix X^* in the same way as an approximation to the top eigenvectors of W is obtained with the probabilistic SVD approximation method.

In more detail, the solution steps are as follows:

1. Calculate the sampled submatrix $S \in \mathbb{R}^{n \times s}$ of W to obtain the matrix $S^\top S \in \mathcal{S}_+^s$.

2. Solve the following *small-size* version of the SDP problem 5.8, which yields the solution matrix $\tilde{X}^* \in \mathcal{S}_+^s$:

$$\begin{aligned} \max_{\tilde{X} \succeq 0} \quad & S^\top S \bullet \tilde{X} \\ \text{s.t.} \quad & ee^\top \bullet \tilde{X} = 0 \\ & \mathrm{diag}(\tilde{X}) = e \; . \end{aligned} \tag{5.9}$$

3. Compute an approximative Cholesky factor $\hat{V} \in \mathbb{R}^{n \times s}$ for the solution matrix X^* of the original SDP problem (5.8) by setting $\hat{V} = S\tilde{V}$ (similarly to (5.5)), where $\tilde{V} \in \mathbb{R}^{s \times s}$ denotes the Cholesky factor of the solution matrix $\tilde{X}^* = \tilde{V}\tilde{V}^\top$ of (5.9).

4. Normalize the rows $\hat{v}_i$ of $\hat{V}$ to satisfy the original norm constraint on the rows of the Cholesky factor.

[4] In particular, if we apply this modification directly to W' for the normalized cut, we can set $\gamma = 1$ since all eigenvalues of W' are known to be larger than -1 (see Lemma 3.4).

5. Adapt the randomized hyperplane technique (Section 4.2.3) to calculate binary vectors $x \in \{-1,+1\}^n$: for random vectors r from the unit sphere in $\mathbb{R}^s$, compute $x_i = \operatorname{sgn}(\hat{v}_i r)$ for $i = 1,\ldots,n$ (note that $\hat{v}_i$ is a row vector). As final combinatorial solution we then select that binary vector x which yields the smallest value for the following adjusted version of the original objective function in (4.5):

$$x^\top L_s x \;, \tag{5.10}$$

where L_s is obtained from L by setting all columns to zero that correspond to points which are not sampled: $L_s = D_s - W_s = \begin{pmatrix} \operatorname{Diag}(S^\top e) & 0 \\ 0 & 0 \end{pmatrix} - (S \;\; 0)$.

Concerning the final step, the modification (5.10) can be interpreted in the following way: instead of seeking a minimum cut in the complete problem graph, a sparsified graph is examined that only contains the edges between the samples with their full weight and the half-weighted edges between samples and non-samples. Thus, the objective function is slightly adjusted to take the confidence in the entries of the solution vector into account.

Normalized Cut Relaxation

If the similarity matrix W is positive semidefinite, so is W', and we can apply the probabilistic SVD approximation method directly to approximate the top eigenvectors of W'. Since this requires the sampled part S of W' to be calculated exactly, we assume that besides the similarities to the sampled points (the sampled part of W), the complete degree vector $d = De$ is computed in a preprocessing step.[5] In this context, a slightly different method is proposed in [54] in connection with the Nyström method: instead of W', the original similarity matrix W is approximated first and normalized afterwards. In this way, the time consuming computation of the correct degree vector d is avoided. However, this procedure is not applicable for the SVD approximation method, which requires to approximate W' *directly*.

In order to obtain a binary solution, the second step of the normalized cut relaxation demands to find a suitable threshold value on the second smallest eigenvector v_2 of the generalized eigenvalue problem (3.17) (see Section 3.1.2). While by multiplication with $D^{-\frac{1}{2}}$, the computed approximate eigenvectors $\hat{Q}_k$ of W' are easily transformed into the corresponding approximative eigenvectors of the original normalized cut problem, we meet two other problems in this context: first, several experiments indicated that the information contained in the second largest eigenvector of the full problem matrix W' may now be shifted to another of the top approximative eigenvectors, so using only the second largest approximative eigenvector can be misleading. Second, due to time and memory restrictions, we cannot make use of the original normalized cut criterion (3.4) (which involves the complete similarity matrix W) to calculate the optimal threshold value.

In order to deal with these problems, we calculate the binary solution with a modified technique that is still directly based on the normalized cut criterion,

[5] Note that the calculation of the complete degree vector d is not necessary for the SDP relaxation approach.

but only requires the sampled part of the similarity matrix W along with the complete degree vector $d = De$. To this end, we first normalize the rows of the matrix $D^{-\frac{1}{2}}\hat{Q}_k$ (which contains the approximative eigenvectors to the original normalized cut problem) to unit length to project them onto the unit sphere (cf. [190, 134]). For each of the top "projected" eigenvectors obtained in this way, we then compute the threshold which yields a binary vector $x \in \{-1, +1\}^n$ that minimizes the following adjusted version of (3.4):[6]

$$\frac{x^\top L_s x}{x^\top D_s(x+e)} + \frac{x^\top L_s x}{x^\top D_s(x-e)} \,, \tag{5.11}$$

with L_s and D_s being defined as in (5.10). Comparing the optimal binary vectors x obtained for each of the first top projected eigenvectors, the final solution is given by that x which results in the smallest value for (5.11).

Note that in connection with the Nyström method, a different technique is proposed to compute the final partitioning for the normalized cut relaxation [54]: based on the embedding of the points into the space $\mathbb{R}^k$ given by the (scaled) rows of the matrix $D^{-\frac{1}{2}}\hat{Q}_k$, a direct segmentation into multiple parts is obtained by applying the k-means algorithm. Although this method seems to be more robust by considering the information contained in several approximate eigenvectors at once, we do not study it here. In fact, it has been criticized that the solution calculated with k-means is not based on the original normalized cut criterion any more, since the embedding yields a different grouping problem [155]. Instead, we focus on *hierarchical binary* partitionings in this section; direct multiclass segmentation methods are considered in Section 6.2.2.

5.3.5 Experimental Results

In this section, we present several results for unsupervised partitioning problems obtained by applying the probabilistic SVD approximation technique in connection with the SDP relaxation approach and the normalized cut relaxation, respectively. Besides evaluating the performance statistically for artificial point sets, we segment large real world images within the hierarchical framework as suggested in Section 5.1.

Concerning the graph representation of the problems, the SVD approximation method needs a *dense* similarity matrix W to work: otherwise, the sampled submatrix S may contain zero-rows, which prohibits to infer the group membership of the corresponding unsampled pixels. Since applying a shortest paths approach like method (ii) from Section 4.4.2 for this purpose is prohibited due to time and memory restrictions, we simply expand the feature vectors y_i in this section by including the position of each point i along with its color in the perceptually uniform L*u*v* space. The required similarity values w_{ij} are then computed directly for each pair of points i and j as

$$w_{ij} = e^{-d_M(i,j)} \,,$$

[6] The original version (3.4) corresponds to (5.11) with L_s and D_s being replaced by the full matrices L and D, respectively.

where $d_M(i,j) := \sum_k \frac{(y_i - y_j)_k^2}{\sigma_k}$ denotes the Mahalanobis distance between the corresponding feature vectors y_i and y_j, with appropriately adjusted scaling factors σ_k. As this results in a positive semidefinite similarity matrix W, we can use W unchanged for all the following applications.

Statistical Performance Evaluation

To measure the performance of the sampling-based versions of the partitioning methods statistically, we create two different point sets as depicted in Figure 5.6, top. Using the complete similarity matrix (based on the Euclidean distances of the points), both the SDP relaxation approach and the normalized cut relaxation are able to separate the clusters correctly. For different sampling rates, we then compute partitionings based on the probabilistic SVD approximation method and compare them to the optimal solution by counting the number of misclassified points. To derive some significant statistics, this experiment is repeated 100 times for each sampling rate, with different sample points selected.

The diagrams in Figure 5.6 reveal that for both the SDP and the normalized cut relaxation, good results are obtained with the probabilistic SVD approximation method, also for relatively small sampling rates. In particular, the quite simple example in Figure 5.6, left, always gives a mean error below 5% if at least 10% of the points are sampled. In comparison, note that for the point set depicted in Figure 5.6, right, decreasing sample numbers soon result in a significant loss of structure as the similarity values are solely based on Euclidean distances, which makes this problem quite intricate. Although for this reason, the measured mean error increases for smaller sampling rates, it should be mentioned that both partitioning approaches still were able to find the optimal solution at least once down to a sampling rate of 10%.

Moreover, Figure 5.6, bottom, shows that the computational effort is reduced by applying the probabilistic SVD approximation method. With a nearly quadratic decay of the time needed to solve the partitioning problems, this is most significant for the SDP relaxation: for small sampling rates, the corresponding computational effort even becomes comparable to the normalized cut relaxation.

For comparison, we also apply the Nyström method to the normalized cut relaxation for these examples. While the corresponding clustering performance resembles that of the probabilistic SVD approximation-based approach (Figure 5.6, middle), there are two striking discrepancies: the bad performance for high sampling rates depicted in Figure 5.6, middle left, is due to the fact that for this example the similarity matrix W is nearly singular, which leads to inaccurate results in the calculation of the inverse submatrix A^{-1}. The increase of the computational effort when sampling based on the Nyström method is introduced (cf. Figure 5.6, bottom) can be attributed to the same cause: the high complexity of calculating an inverse matrix makes this method inefficient for larger sample numbers. Hence, these results approve the theoretical drawbacks of the Nyström method indicated in Section 5.3.3.

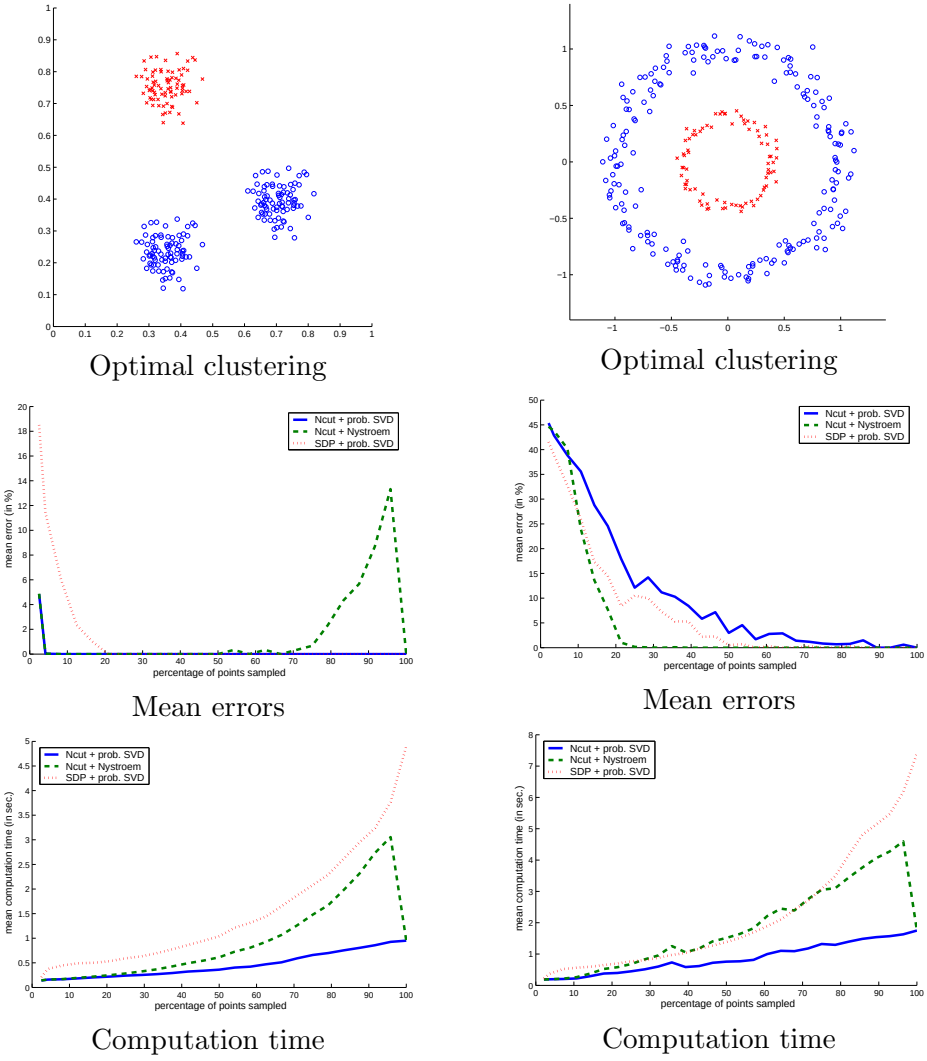


Figure 5.6: Statistical performance for two clustering problems, based on 100 experiments for each sampling rate. Using all points, the optimal solutions for both point sets are found with the SDP relaxation approach as well as with the normalized cut relaxation (top). Concerning the quality of the solutions obtained with the probabilistic SVD approximation technique (middle), both methods give good results also for relatively small sampling rates, especially for the quite simple example depicted left. In particular, the computational effort is reduced strongly for the SDP relaxation (bottom), thus making it comparable to that of the normalized cut relaxation for small sample numbers.

Image Segmentation

Figure 5.7 gives the segmentation results obtained with the probabilistic SVD approximation method for a small patch of a larger color image (cf. Figure 3.6) when 6.2% of the points ($s = 80$) are sampled. In order to compare the performance for both partitioning approaches, we also computed the corresponding optimal solutions based on the complete similarity matrix for this example (Fig-

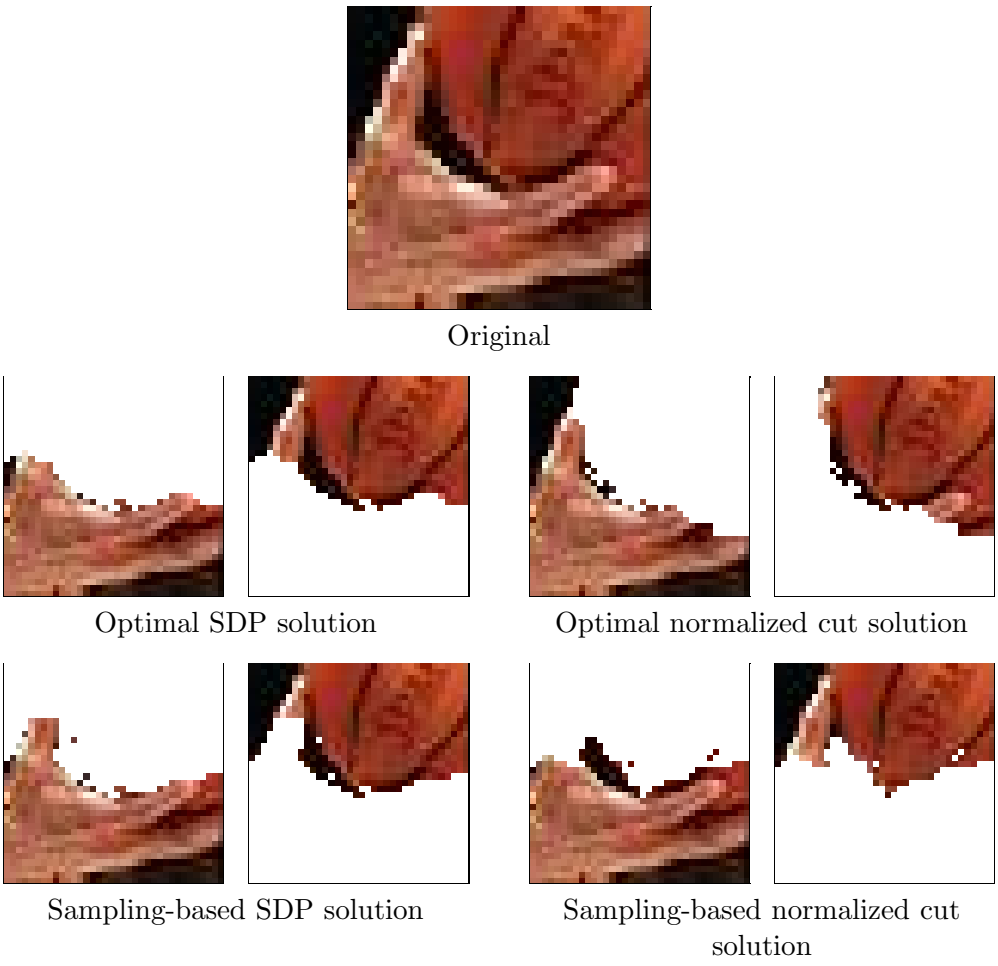
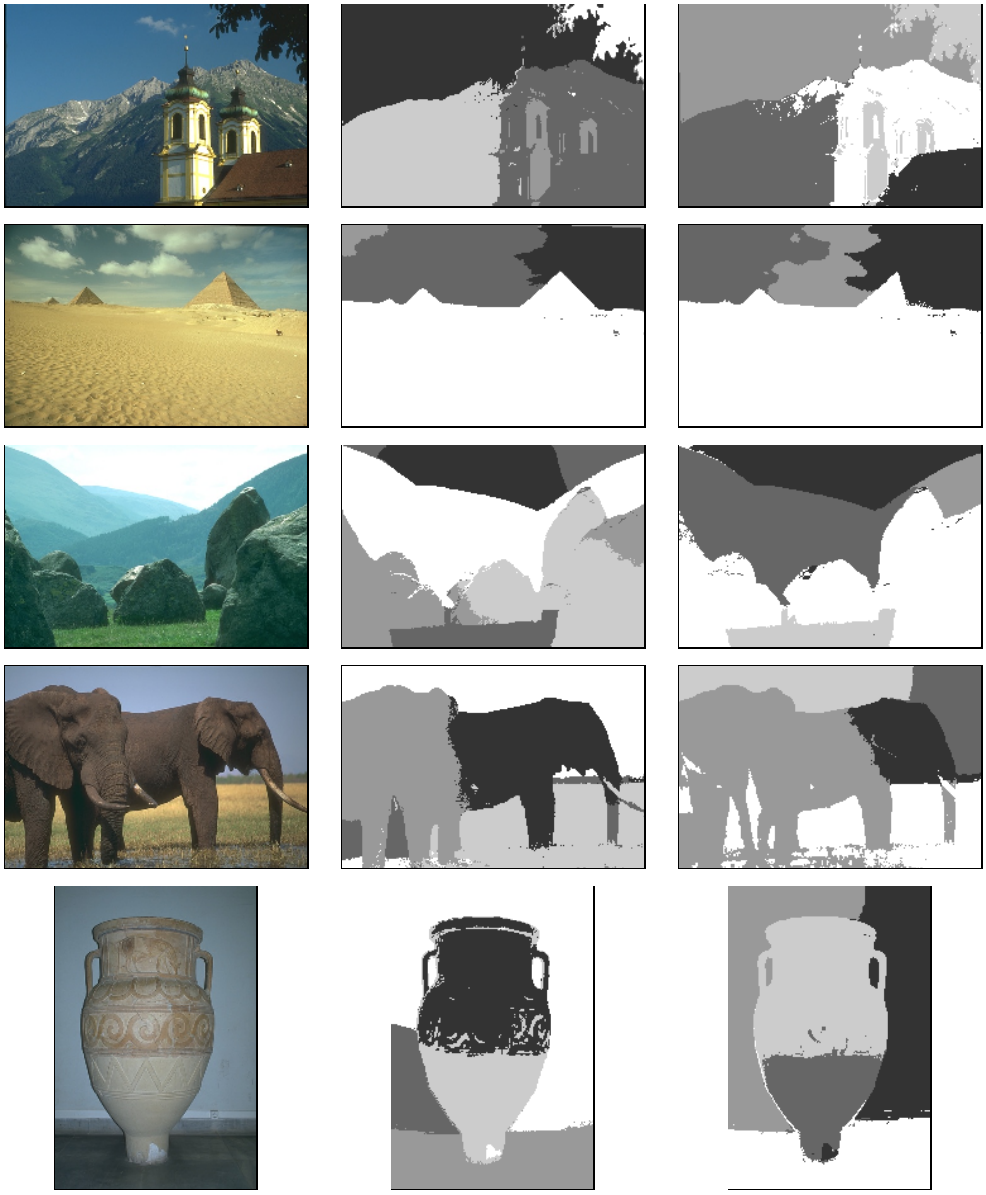


Figure 5.7: Comparison of segmentations with and without sampling. For a small patch (36×36 pixels) of a larger color image (cf. Figure 3.6), the probabilistic SVD approximation method is applied based on sampling 6.2% of the pixels. While maintaining a satisfying segmentation quality, the computational effort to produce the approximate solutions is reduced enormously (by more than 95%).

ure 5.7, middle row). While the sampling-based binary partitions are reasonable approximations to the optimal segmentations, the computational effort needed to produce these results is drastically reduced: from 13.5 minutes for the complete SDP, and 3.5 minutes for the complete normalized cut, to 5–6 seconds for *both* sampling-based approaches.

Finally, we apply the sampling-based partitioning approaches to large real world image segmentation problems. The results for several examples from the Berkeley segmentation dataset [121] are depicted in Figure 5.8. In order to produce partitionings into more than two segments, we again use the hierarchical framework presented in Section 5.1. For these examples, the binary partitionings computed for each candidate segment are always based on 100 randomly selected pixels (which corresponds to 0.26% of the entire image). To decide which segment should be split next in each step, we apply a selection



Original | Sampling-based SDP solution | Sampling-based normalized cut solution

Figure 5.8: Hierarchical segmentation based on sampling. For five different images of size 240×160 pixels from the Berkeley segmentation dataset [121], the probabilistic SVD approximation method is applied based on sampling $s = 100$ pixels in each hierarchical step (corresponding to 0.26% of the total number of pixels). Both partitioning approaches, the SDP relaxation as well as the normalized cut relaxation, give satisfactory results.

procedure that slightly differs from the one presented in Section 5.1.2: instead of generally choosing the largest segment, we pick the candidate that yields the lowest normalized cut value. This procedure is stopped after four steps, which results in a partitioning into at most five segments.

The results in Figure 5.8 reveal that for both the SDP and the normalized cut segmentation approach, the application of the probabilistic SVD approximation method is successful: taking into account that no effort is made to smooth the segments, to select the sampled pixels more deliberately or to stop the partitioning process at a more adequate number of segments, the quality of the segmentations is satisfactory. Concerning the computational effort, it takes just about 350 seconds for the sampling-based normalized cut relaxation and about 110 seconds for the sampling-based SDP relaxation to find the first binary partitioning for these images. In this context, note that the most time consuming step for problems of this size consists in the final calculation of the binary solution: the vectors to be examined are by orders of magnitude larger than the solutions of the corresponding sampling-based small scale problems. This fact also explains the larger solution time for the normalized cut relaxation, since we test several approximative eigenvectors for good cut values — in contrast to the SDP relaxation, where only a fixed number of random hyperplanes is evaluated.

In comparison to the over-segmentation-based partitioning method presented in Section 5.2, the image segmentation results of the sampling-based techniques naturally are more noisy, since they are obtained on a pixel basis without any smoothing effort. For this reason, we do not provide quantitative quality measures (like the F-measure from Section 5.2.4) here. However, the reduced computational effort permits computing multiple segmentations of a single image based on different samplings, which then can be combined to obtain the final solution by calculating the most probable group membership for each pixel (see e.g. [176]). For the sampling-based normalized cut relaxation, this topic is addressed in more detail in the diploma thesis of Hanno Ackermann [2], along with an elaborate analysis of the parameters involved in the probabilistic SVD approximation method.

Chapter 6

Solving Non-Binary Problems

So far, we have only considered segmentation tasks that are based on *binary* optimization problems which can be cast in the general form (2.1). Problems of this type are amenable to the general convex relaxation approach presented in Chapter 4. In practice, however, the situation may be more complicated: instead of seeking good binary solutions, a direct decomposition of the image into more than two parts is often desired.

In this context, a first approach was presented in Section 5.1, where a segmentation into multiple parts was obtained by computing binary partitionings in a hierarchical way. However, such a hierarchical framework is only appropriate for the unsupervised segmentation task, and not for perceptual grouping or image restoration. In fact, the perceptual grouping problem (see Section 2.2) is intrinsically binary as a one-against-all task: image elements either belong to a shape (indicated by high interaction coefficients w_{ij}) or they do not; other shape definitions lead to different grouping problems, that have to be solved independently. On the other hand, the energy functional (2.11) for the binary image restoration problem (see Section 2.3) involves comparisons with previously defined, prototypical representatives of each group. Thus for multiple classes, a hierarchical framework would lead to binary restoration problems that involve the comparison of one class against all the others, which usually is not feasible since a suitable representative for the all-class cannot be defined.

In this chapter, we will therefore present a natural, direct extension of the binary restoration problem to the non-binary case, which can also be solved by semidefinite relaxation (Section 6.1). For the sake of completeness, and since the relaxation is quite similar to that of the multiclass restoration problem, we will also briefly introduce how the binary unsupervised partitioning problem can be extended to find a multiclass segmentation in a direct way (Section 6.2).

6.1 Multiclass Restoration

As already stated above, the binary restoration problem (2.12) is not suited for a hierarchical application. However, in contrast to unsupervised segmentation tasks, the number of classes k the image should be partitioned into is usually defined in advance. This knowledge allows deriving a direct extension of the

binary restoration problem to multiple classes in a straightforward way (Section 6.1.1), which can also be solved by a semidefinite relaxation approach (Section 6.1.2). First experimental results are given in Section 6.1.3.

6.1.1 Problem Formulation

To extend the binary energy functional (2.11) to multiple classes, we now indicate the class membership of an image element i by a vector $x_i \in \{e_1, \ldots, e_k\}$ taking as value one of the k unit vectors from $\mathbb{R}^k$. Moreover, we assume that each locally measured feature vector $g_i \in \mathbb{R}^m$ is known to originate from one of k prototypical vectors $u_1, \ldots, u_k \in \mathbb{R}^m$, which yield the columns of the prototype matrix $U \in \mathbb{R}^{m \times k}$. Generalizing the separation costs for associated image elements i, j in (2.11) to $P_{ij} D(x_i, x_j) = \lambda \|x_i - x_j\|^2 = 2\lambda(1 - x_i^\top x_j)$, and the assignment costs to

$$\begin{aligned} C_i(x_i) &= \|Ux_i - g_i\|^2 \\ &= (Ux_i - g_i)^\top (Ux_i - g_i) \\ &= x_i^\top (U^\top U) x_i - 2x_i^\top (U^\top g_i) + \|g_i\|^2 \\ &= x_i^\top \operatorname{diag}(U^\top U) - 2x_i^\top (U^\top g_i) + \|g_i\|^2 \\ &= x_i^\top \Big(\operatorname{diag}(U^\top U) - 2U^\top g_i \Big) + \|g_i\|^2 \,, \end{aligned}$$

we obtain the generalized energy functional

$$\begin{aligned} E_{MR}(x) &:= \sum_i \|Ux_i - g_i\|^2 + \lambda \sum_{\langle i,j \rangle} \|x_i - x_j\|^2 \\ &= \operatorname{Tr}\Big(X \big(\operatorname{diag}(U^\top U) e^\top - 2U^\top G \big) \Big) + \|G\|_{\mathrm{F}}^2 - \lambda \operatorname{Tr}(MXX^\top) + 2\lambda |\langle i,j \rangle| \\ &= \operatorname{Tr}\Big(-\lambda MXX^\top + \big(e \operatorname{diag}(U^\top U)^\top - 2G^\top U \big) X^\top \Big) + \|G\|_{\mathrm{F}}^2 + 2\lambda |\langle i,j \rangle| \end{aligned} \tag{6.1}$$

by inserting into the generic energy function (2.10). In (6.1), $X \in \mathbb{R}^{n \times k}$ denotes the indicator matrix containing the class indicator vectors x_i as rows, $G \in \mathbb{R}^{m \times n}$ comprises the feature vectors g_i as columns, and $|\langle i,j \rangle|$ denotes the total number of element associations. The symmetric matrix $M \in \mathcal{S}^n$ subsumes these associations between the image elements by entries $M_{ij} = 1$ whenever two elements i, j are neighbored, and is zero otherwise.

Disregarding the constant terms, this results in the following optimization problem for *multiclass restoration*:

$$\begin{aligned} z_{MR}^* := \min_{X \in \mathbb{R}^{n \times k}} \quad & \operatorname{Tr}\Big(-\lambda MXX^\top + \big(e \operatorname{diag}(U^\top U)^\top - 2G^\top U \big) X^\top \Big) \\ \text{s.t.} \quad & Xe^k = e^n \\ & X_{ij} \in \{0,1\} \quad \forall\, i,j \,, \end{aligned} \tag{6.2}$$

where $e^j \in \mathbb{R}^j$ denotes the vector of all ones of appropriate size. Note that the first constraint in (6.2) requires each row of X to sum to one, which in

connection with the second constraint ensures that each row corresponds to a unit vector e_i.

This combinatorial optimization problem resembles the *quadratic assignment problem* (QAP, see, e.g., [137, 28]), which aims for optimally placing n given activities at n given locations by minimizing a cost function of the form $\mathrm{Tr}(AXBX^\top - 2CX^\top)$ with positive matrices $A, B, C \in \mathbb{R}^{n\times n}$. In fact, if we allow multiple activities to be placed at the same location, the multiclass restoration problem exactly becomes a special case of the *uncapacitated* QAP [105]; this can easily be established by defining the cost matrix $B = E - I$ and the flow matrix $A = \lambda M$.

The original QAP has attracted considerable interest in connection with semidefinite relaxation approaches [206, 26]. In the next section, we will show how these methods can be generalized to find approximate solutions for problems of the type (6.2). Alternatively, several of the approaches already mentioned in Section 2.3 can be used for this purpose. Moreover, other relations of (6.2) to different optimization problems are stated in the literature: whereas Kleinberg and Tardos [105] point out the equivalence of this special case of their uniform labeling problem to a pairwise homogeneous Markov random field, Boykov et al. [24] prove the equivalence to a multiway cut problem.

6.1.2 Lagrangian Relaxation

Analogously to the relaxation of the QAP presented in [206], we perform Lagrangian relaxation of (6.2). For ease of notation, let $\tilde{M} := -\lambda M$ and $C := e\,\mathrm{diag}(U^\top U)^\top - 2G^\top U \in \mathbb{R}^{n\times k}$. We start with representing the constraints in a quadratic form, to obtain the following equivalent problem to (6.2):

$$\begin{aligned} z^*_{MR} = \min_{X\in\mathbb{R}^{n\times k}} \; & \mathrm{Tr}\Big(\tilde{M}XX^\top + CX^\top\Big) \\ \text{s.t.} \quad & \|Xe^k - e^n\|^2 = 0 \\ & X_{ij}^2 - X_{ij} = 0 \quad \forall\, i,j\ . \end{aligned} \tag{6.3}$$

Using the Lagrange multipliers $W \in \mathbb{R}^{n\times k}$ and $u_0 \in \mathbb{R}$, we add the constraints to the objective function, and perform relaxation by virtue of the "minimax inequality" [152] (see Section 1.4 for notations):

$$\begin{aligned} z^*_{MR} &= \min_X \max_{W,u_0} \mathrm{Tr}\Big(\tilde{M}XX^\top + CX^\top\Big) + \sum_{i,j} W_{ij}(X_{ij}^2 - X_{ij}) \\ &\qquad + u_0(Xe^k - e^n)^\top(Xe^k - e^n) \\ &\geq \max_{W,u_0} \min_X \mathrm{Tr}\Big(\tilde{M}XX^\top + CX^\top\Big) + \mathrm{Tr}\Big(W(X\circ X - X)^\top\Big) \\ &\qquad + u_0\,\mathrm{Tr}\Big(XE_kX^\top - 2E_{n\times k}X^\top\Big) + u_0 n \\ &= \max_{W,u_0} \min_X \mathrm{Tr}\Big(\tilde{M}XX^\top + W(X\circ X)^\top + X(u_0E_k)X^\top \\ &\qquad + (C - W - 2u_0E_{n\times k})X^\top\Big) + u_0 n\ . \end{aligned}$$

Next we homogenize the objective function by multiplying X with a constrained scalar x_0, which increases the dimension of the problem by one. This additional constraint is then inserted into the objective function by introducing the Lagrange multiplier w_0:

$$\begin{aligned} z_{MR}^* &\geq \max_{W,u_0} \min_{X,x_0^2=1} \operatorname{Tr}\Big(\tilde{M}XX^\top + W(X\circ X)^\top + X(u_0E_k)X^\top \\ &\qquad + x_0(C - W - 2u_0E_{n\times k})X^\top\Big) + u_0nx_0^2 \\ &\geq \max_{W,u_0,w_0} \min_{X,x_0} \operatorname{Tr}\Big(\tilde{M}XX^\top + W(X\circ X)^\top + X(u_0E_k)X^\top \\ &\qquad + x_0(C - W - 2u_0E_{n\times k})X^\top\Big) + u_0nx_0^2 + w_0x_0^2 - w_0 \\ &=: s_d^* \,. \end{aligned}$$

Transforming the problem variables x_0 and X into a vector by defining $y := \binom{x_0}{\operatorname{vec}(X)}$, we finally obtain

$$s_d^* = \max_{W,u_0,w_0} \min_{y} \; y^\top\Big(L_{\tilde{M},C} + A_{W,w_0} + u_0F\Big)y - w_0 \,, \tag{6.4}$$

with

$$L_{\tilde{M},C} := \begin{pmatrix} 0 & \frac{1}{2}\operatorname{vec}(C)^\top \\ \frac{1}{2}\operatorname{vec}(C) & I_k \otimes \tilde{M} \end{pmatrix} \,, \tag{6.5}$$

$$A_{W,w_0} := \begin{pmatrix} w_0 & -\frac{1}{2}\operatorname{vec}(W)^\top \\ -\frac{1}{2}\operatorname{vec}(W) & \operatorname{Diag}(\operatorname{vec}(W)) \end{pmatrix} \,, \tag{6.6}$$

$$F := \begin{pmatrix} n & -(e^{nk})^\top \\ -e^{nk} & E_k \otimes I_n \end{pmatrix} \,, \tag{6.7}$$

where $A \otimes B$ denotes the Kronecker product of A and B (see Section 1.4).

There is a hidden semidefinite constraint in (6.4): the inner minimization is bounded below only if the matrix in the quadratic term is positive semidefinite, in which case the corresponding minimum becomes zero. This yields the following relaxation of (6.2):

$$\begin{aligned} s_d^* = \max_{W,u_0,w_0} \quad & -w_0 \\ \text{s.t.} \quad & L_{\tilde{M},C} + A_{W,w_0} + u_0F \succeq 0 \,. \end{aligned} \tag{6.8}$$

To obtain a direct semidefinite relaxation of (6.2), we derive the Lagrangian dual of (6.8). To this end, first observe that the matrix in (6.6) can be split into $A_{W,w_0} = \sum_{i=0}^{nk} w_iA_i$ by defining $w := \operatorname{vec}(W)$ and $A_i \in \mathcal{S}^{nk+1}$ with

$$(A_i)_{i_1,i_2} := \begin{cases} 1 & i_1 = i_2 = i+1 \\ -\frac{1}{2} & i \neq 0, i_1 = 1, i_2 = i+1 \text{ and } i \neq 0, i_1 = i+1, i_2 = 1 \\ 0 & \text{elsewhere} \end{cases} \,. \tag{6.9}$$

Using the dual positive semidefinite matrix variable $Y \in \mathcal{S}_+^{nk+1}$, we get

$$\begin{aligned} s_d^* &= \max_{w_0,w,u_0} \min_{Y \succeq 0} -w_0 + \operatorname{Tr}\Big(Y(L_{\tilde{M},C} + \sum_{i=0}^{nk} w_i A_i + u_0 F)\Big) \\ &\leq \min_{Y \succeq 0} \max_{w_0,w,u_0} \operatorname{Tr}(L_{\tilde{M},C} Y) + w_0\big(\operatorname{Tr}(A_0 Y) - 1\big) + \sum_{i=1}^{nk} w_i \operatorname{Tr}(A_i Y) \\ &\qquad\qquad + u_0 \operatorname{Tr}(FY) \\ &=: s_p^* \,. \end{aligned}$$

As the inner maximization is unconstrained, this minimization problem is finite only if the factors in the last three terms are zero. Using this hidden constraint, we finally obtain the following semidefinite program as the dual of (6.8):

$$\begin{aligned} s_p^* = \min_{Y \succeq 0} \quad & L_{\tilde{M},C} \bullet Y \\ \text{s.t.} \quad & A_0 \bullet Y = 1 \\ & A_i \bullet Y = 0 \quad \text{for } i = 1, \ldots, nk \\ & F \bullet Y = 0 \,. \end{aligned} \tag{6.10}$$

The connection of this semidefinite relaxation with the original integer problem (6.2) becomes clear immediately: the $(0,1)$-matrix $X \in \mathbb{R}^{n \times k}$ is transformed into a vector $\operatorname{vec}(X)$ and then lifted into the higher-dimensional space of positive semidefinite matrices $\mathcal{S}_+^{nk+1}$ by setting

$$Y := \begin{pmatrix} 1 \\ \operatorname{vec}(X) \end{pmatrix} \Big(1, \operatorname{vec}(X)^\top\Big) \,. \tag{6.11}$$

As for the binary case (cf. Section 4.2.1), the relaxation consists in discarding the intractable rank one constraint on Y, and minimizing over the space of positive semidefinite matrices instead. Besides the A_0-constraint, which is an artificial one to enable the homogenization of the objective function, the other constraints in (6.10) directly correspond to the constraints in the original problem formulations (6.2) or (6.3): the A_i-constraints guarantee that the diagonal and the first row (and column) of Y are identical, thus modeling the $(0,1)$-constraint on the entries of X, whereas the F-constraint is derived from the sum-one-constraint on the indicator vectors constituting the rows of X.

Concerning the solvability of the SDP relaxation (6.10), we have the following lemma:

Lemma 6.1. *A feasible solution matrix Y for* (6.10) *is singular, with at least n of its eigenvalues being equal to zero.*

Proof. The constraint matrix $F \neq 0$ is positive semidefinite: as can easily be calculated, its non-zero eigenvalues are $\lambda_{nk+1} = n + k$ and $\lambda_{n(k-1)+2} = \cdots = \lambda_{nk} = k$. As Y is also positive semidefinite, the constraint $F \bullet Y = 0$ in (6.10) directly implies that FY has to be the null-matrix [3, Lemma 2.9]. Hence, $YF_i = 0$ for each column F_i, which shows the singularity of Y. As exactly n columns F_i of F are linearly independent (namely $i = 2, \ldots, n+1$), the dimension of the null space $\ker(Y)$ is at least n. □

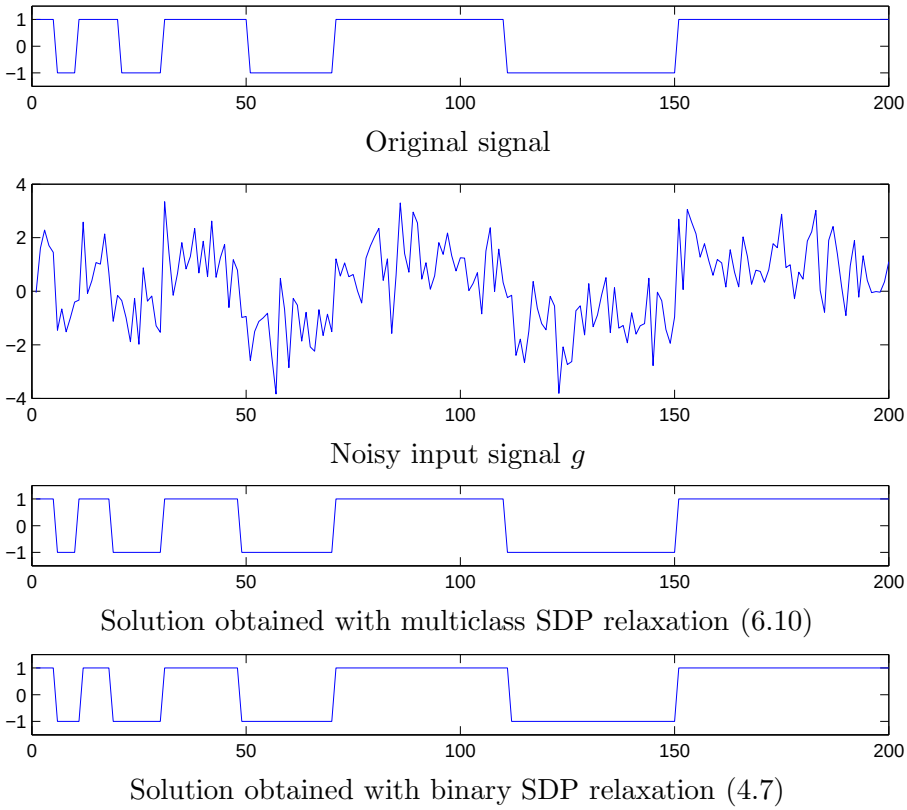


Figure 6.1: One-dimensional restoration. The original signal (top) is distorted by Gaussian white noise (middle) and restored with the SDP relaxations presented in Section 4.2 and Section 6.1.2. The reconstructions obtained for this representative example differ in only two points from each other.

Lemma 6.1 implies that the primal semidefinite program (6.10) has no strictly interior point. On the other hand, it is not difficult to find a strictly interior point for the dual SDP (6.8).[1] Hence, the Slater condition holds for the dual, so that by Theorem 4.2 there is no duality gap: $s_p^* = s_d^*$. However, due to Lemma 6.1, it is not guaranteed that the optimal value of the dual SDP (6.8) is attained (cf. also Section 4.2.2). Therefore, interior point methods can suffer from instability when solving the SDP relaxation (6.10) and may not converge [206]. This problem is circumvented by reverting to other SDP solvers that are based on different algorithms (like, e.g., PENNON [108], which uses a generalized version of the augmented Lagrangian method), or by projecting the problem onto a lower dimensional face of the semidefinite cone [206].

For the QAP, Zhao et al. [206] show that it is possible to tighten the SDP relaxation by incorporating additional constraints, that are redundant for the original problem. Such constraints may also be added for the multiclass restoration problem. For example, we know that the diagonal entries of the matrix $XX^\top$ (the squared norms of the indicator vectors) are equal to 1, which gives

[1] This can be accomplished by setting $u_0 = 0$, and choosing w_0 and the entries of W large enough to make the diagonal of the matrix $L_{\tilde{M},C} + A_{W,w_0}$ as dominant as necessary to yield a positive definite matrix.

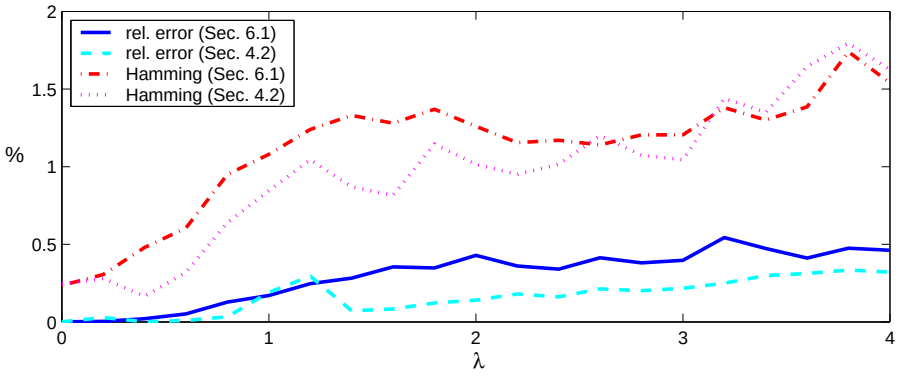


Figure 6.2: Statistics of the SDP relaxation comparison. The results reveal that the binary SDP relaxation in general performs slightly better than the multiclass SDP relaxation.

the additional constraint $\mathrm{Tr}(XX^\top) = n$. Future work will show which meaningful constraints can be added and in how far they are useful to find good approximative solutions. In this work, however, we rely on the basic relaxation (6.10).

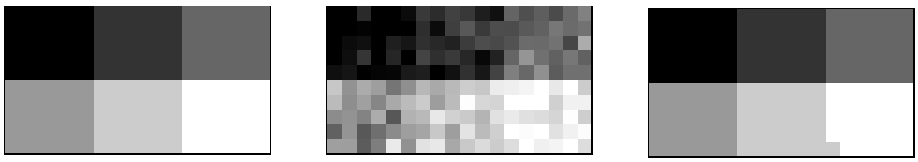
6.1.3 Experimental Results

In this section, we present some results obtained for the multiclass restoration problem based on the primal-dual SDP relaxation pair (6.10) and (6.8). In order to derive a suboptimal integer solution for the original problem (6.2), the first column Y_1 of the solution matrix Y^* of the primal problem (6.10) is used: since $Y_1 = \binom{1}{\mathrm{vec}(X)}$, we seek the largest value in each block of length k in Y_1 (starting with the second entry) to give the position of the one-entry in the corresponding indicator vector.[2]

As a first experiment, we compare the performance of the multiclass SDP relaxation (6.10) for the case $k = 2$ with the direct binary relaxation (4.7). To this end, a synthetic one-dimensional signal (Figure 6.1, top) is first distorted by adding Gaussian white noise and then restored based on both SDP relaxation approaches — see Figure 6.1 for a representative example. In order to derive some significant statistics, we compute 100 noisy versions of the signal for different values of the smoothness parameter λ, and calculate the mean relative errors of the objective values and the mean relative Hamming distance, both in comparison to the optimal solution (cf. Section 4.4.1 for a more detailed explanation of this experiment).

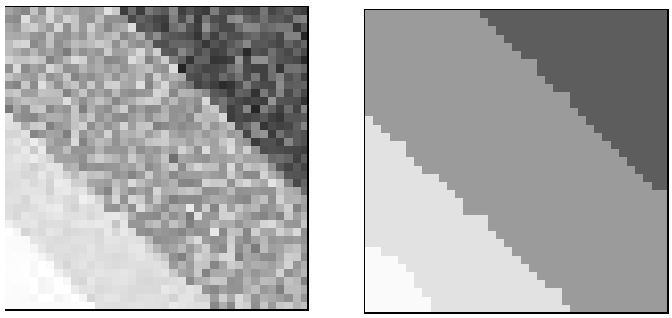
Figure 6.2 depicts the corresponding statistical results. Interestingly, a slightly better performance of the binary SDP relaxation (4.7) can be observed. However, this is not due to the tightness of the relaxation: the experiments reveal that the objective values of both relaxations coincide. Instead, this difference rather indicates that the randomized hyperplane technique which is

[2]Note that we cannot apply the randomized hyperplane technique (Section 4.2.3) here, since the original problem variables are k-dimensional vectors now.



Original Noisy input SDP restoration

Figure 6.3: Multiclass image restoration result. The original image (left) of 10×18 pixels is degraded by adding Gaussian white noise (middle). The result of the reconstruction (using $\lambda = 0.05$) is almost perfect (right): only one pixel is classified incorrectly.



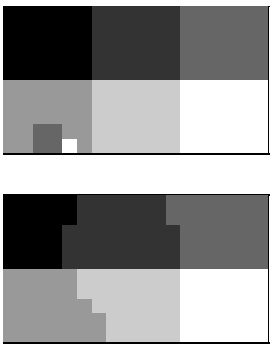
Noisy input SDP restoration

Figure 6.4: Multiclass image restoration result. A patch of the noisy diamond image from [24] of 37×37 pixels (left) is restored with the SDP relaxation approach with $\lambda = 0.05$ (right).

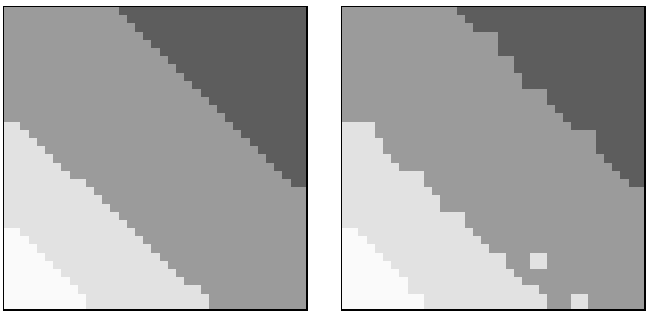
used to obtain a combinatorial solution from the solution matrix X^* of (4.7) performs better than the method used to find the indicator vectors from the first column Y_1 of the solution of the multiclass relaxation (6.10). Nevertheless, the restorations are still remarkably good: the average relative error of the objective value and the average relative Hamming distance both are below 2%, with standard deviations below 0.58% (objective error) and 1.56% (Hamming distance), respectively. Yet it should be mentioned that the solution of the binary SDP relaxation (4.7) is calculated in less than a second, whereas it takes 6–7 seconds to solve the multiclass SDP relaxation (6.10), which is obviously caused by the larger problem size ($Y \in \mathcal{S}_+^{401}$ for (6.10) vs. $X \in \mathcal{S}_+^{201}$ for (4.7)).

Figures 6.3 and 6.4 show the restorations of two noisy images originally comprised of multiple gray-values. For these examples, we first convert the pixel values to the interval $[0, 1]$, and employ the PENNON solver [108] to find the solution of the SDP relaxation (6.10) based on a first-order neighborhood (horizontal and vertical adjacent pixels are connected). The reconstructions obtained for both examples are promising: only few pixels are grouped incorrectly.

Figure 6.5 depicts the corresponding restorations computed with the ICM algorithm (see Section 3.3). Comparing the results with the SDP solutions reveals two facts: for the image in Figure 6.3, the tighter SDP relaxation yields a clearly superior reconstruction. On the other hand, the ICM result appears to be slightly better for Figure 6.4. However, this is mainly due to the second-



ICM restoration for Figure 6.3 ($\lambda = 0.02$ and $\lambda = 0.1$)



ICM restoration for Figure 6.4 ($\lambda = 0.1$), second- (left) and first-order (right) neighborhood

Figure 6.5: Restorations obtained with the ICM algorithm, for the noisy images from Figures 6.3 and 6.4, respectively. For the correct choice of the parameter λ, the ICM algorithm is able to find satisfactory results. However, at least for the left image, the reconstruction obtained with the SDP relaxation approach is better. For the right image, the ICM algorithm profits from the second-order neighborhood that also considers the diagonally adjacent pixels.

order neighborhood that is used by ICM, which better accounts for the diagonal structures present in this image. If we switch to a first-order neighborhood, the results become much worse (cf. Figure 6.5, right).

As already indicated by the small size of the sample images, there is one major drawback of the SDP relaxation: since the problem size increases quadratically with nk (the product of the number of pixels and the number of classes), the corresponding semidefinite programs soon become intractable in terms of memory and computational time requirements. For instance, it takes 2.5 minutes to find the solution for the image in Figure 6.3 (problem size: $nk = 1080$), and already more than 3.6 hours for the image in Figure 6.4 ($nk = 5477$), with memory requirements of several hundreds of megabytes. In contrast to that, the ICM algorithm computes the solution in less than one second. Hence, the application of the multiclass SDP relaxation is (yet) restricted to small restoration problems consisting of only few different classes. The future will show whether algorithms emerge which make larger problem instances tractable.

6.2 Unsupervised Multiclass Partitioning

In Section 5.1, we already presented an obvious extension of the binary unsupervised partitioning problem to multiclass segmentation: applying the binary approach in a hierarchical way yields a segmentation of the image into multiple parts. In this section, we will show how the binary problem formulation can directly be extended to a multiclass segmentation problem (Section 6.2.1).

Concerning the choice of the cost function, this approach may be considered more adequate than the hierarchical method, since it minimizes a *global* objective criterion. Spectral relaxation methods are able to exploit this property by

calculating multiple eigenvectors at once (Section 6.2.2). However, the direct method involves at least one drawback: since the cost function directly depends on k, the desired number of clusters in the final segmentation has to be defined in advance. Although there are suggestions for eigenvalue-based criteria which may find appropriate values for k automatically [125], different choices for k may still yield completely different segmentations, which does not seem to be in correspondence to human vision.

Nevertheless, if k is known beforehand, direct multiclass partitioning is a reasonable technique to find a desired segmentation. In Section 6.2.3, we will therefore also briefly investigate how this problem can be handled by SDP relaxation. First experimental results are presented in Section 6.2.4.

6.2.1 Problem Formulation

Assume that the number k of groups the input data should be partitioned into is known in advance. As in Section 6.1.1, we indicate the cluster membership of the image element i by setting x_i to one of the k unit vectors $e_1, \ldots, e_k$ from $\mathbb{R}^k$. The binary cut cost function $\mathrm{cut}(S, \overline{S})$ in (2.4) can then be extended to express graph partitionings into multiple groups $S_1, \ldots, S_k$ in the following way (see, e.g. [153, 96, 204]):

$$\begin{aligned}
\mathrm{cut}(S_1, \ldots, S_k) &= \frac{1}{4} \sum_{i,j \in V} w_{ij} \|x_i - x_j\|^2 = \frac{1}{2} \sum_{i,j \in V} w_{ij} (1 - x_i^\top x_j) \\
&= \frac{1}{2} \operatorname{Tr}(W(E - XX^\top)) = \frac{1}{2} \operatorname{Tr}(DXX^\top - WXX^\top) \\
&= \frac{1}{2} \operatorname{Tr}(LXX^\top) = \frac{1}{2} \operatorname{Tr}(X^\top LX) \\
&= \frac{1}{2} \sum_{j=1}^{k} X_j^\top L X_j \ ,
\end{aligned} \tag{6.12}$$

with the columns X_j of the multiway cut indicator matrix $X \in \mathbb{R}^{n \times k}$ indicating the elements from class j, and the rows corresponding to unit vectors.

As in (2.6), we can balance the groups by introducing a linear constraint $X^\top c = \beta$ with $c \in \mathbb{R}^n, \beta \in \mathbb{R}^k$ and $\sum_i \beta_i = \sum_i c_i$, to obtain the following representation of the *multiclass partitioning problem*:

$$\begin{aligned}
z^*_{MP} := \min_{X \in \mathbb{R}^{n \times k}} \ & \frac{1}{2} \operatorname{Tr}(X^\top LX) \\
\text{s.t.} \quad & X^\top c = \beta \\
& Xe^k = e^n \\
& X_{ij} \in \{0, 1\} \quad \forall\, i, j \ ,
\end{aligned} \tag{6.13}$$

where $e^j \in \mathbb{R}^j$ denotes the vector of all ones of appropriate size (cf. (6.2)).

For vertices of equal weight, $c = e = (1, \ldots, 1)^\top$, an interesting interpretation of this problem can be derived [153]: first observe that the objective function can equivalently be written as $\frac{1}{2} \operatorname{Tr}(X^\top LX) = \frac{1}{2} d(V) - \frac{1}{2} \operatorname{Tr}(X^\top WX)$,

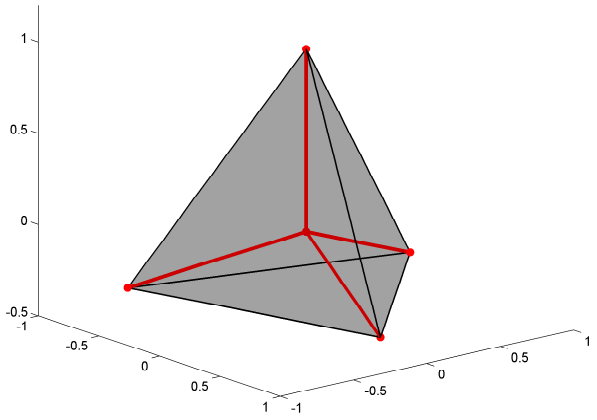


Figure 6.6: Simplex defined by the 4 unit vectors $v_1, \ldots, v_4 \in \mathbb{R}^3$, with its barycenter at the origin.

where W denotes the adjacency matrix of the graph (cf. (6.12)). Defining the partitioning representation matrix $Y := XX^\top$, where $Y_{ij} = 1$ if i and j belong to the same subset S_t, we obtain by comparing W and Y in Frobenius norm:

$$\begin{aligned} \|W - Y\|_{\mathrm{F}}^2 &= \|W\|_{\mathrm{F}}^2 + \|Y\|_{\mathrm{F}}^2 - 2\,\mathrm{Tr}(WY) \\ &= \sum_{i,j=1}^{n} w_{ij}^2 + \sum_{i=1}^{k} \beta_i^2 - 2\,\mathrm{Tr}(X^\top W X)\ . \end{aligned}$$

Hence, since the first two terms are constant, the multiclass partitioning problem (6.13) is equivalent to the following *matrix approximation* problem, which seeks the best approximation to the adjacency matrix W:

$$\begin{aligned} &\min_{Y} \|W - Y\|_{\mathrm{F}} \\ &\text{s.t.} \quad Y \text{ represents a partition .} \end{aligned}$$

A different representation of the multiclass partitioning problem is derived by using another set of possible values for the indicator vectors x_i, as was suggested by Frieze and Jerrum [55] for the max-k-cut problem: instead of signifying the group membership by the k unit vectors $e_i \in \mathbb{R}^k$, they employ a different set of unit vectors $v_1, \ldots, v_k \in \mathbb{R}^{k-1}$ which are required to point as far apart as possible. Geometrically, this demands that all pairs of vectors enclose an angle of the same size. Thus, the unit vectors v_i form an equilateral simplex in $\mathbb{R}^{k-1}$ with its barycenter at the origin [55], as is illustrated in Figure 6.6 for the case $k = 4$. As can be proven easily [55, 78], these vectors $v_i \in \mathbb{R}^{k-1}$ satisfy

$$\begin{aligned} v_i^\top v_i &= 1 && \text{for } i = 1, \ldots, k, \\ v_i^\top v_j &= -\frac{1}{k-1} && \text{for } 1 \le i \ne j \le k\ . \end{aligned}$$

Analogously to (6.12), a partitioning $x_1, \ldots, x_n$ based on the unit vectors v_i yields the cut-value

$$\begin{aligned}\operatorname{cut}(S_1, \ldots, S_k) &= \frac{k-1}{2k} \sum_{i,j \in V} w_{ij}(1 - x_i^\top x_j) \\ &= \frac{k-1}{2k} \operatorname{Tr}(X^\top L X) \ ,\end{aligned}$$

where $X \in \mathbb{R}^{n \times (k-1)}$ again contains the indicator vectors $x_i \in \{v_1, \ldots, v_k\}$ as rows. As in (6.13), it is possible to balance the groups by a linear constraint with $c \in \mathbb{R}^n$ and $\beta \in \mathbb{R}^k$:

$$\begin{aligned}X^\top c &= \sum_{i=1}^{n} c_i x_i \\ &= \left(\sum_{i \in S_1} c_i\right) v_1 + \cdots + \left(\sum_{i \in S_k} c_i\right) v_k \\ &= \sum_{j=1}^{k} \beta_j v_j \ .\end{aligned}$$

Unfortunately, this constraint involves the unit vectors v_j to be given explicitly.[3] However, for a constant vector $\beta = \bar{\beta} e$ with $\bar{\beta} := \frac{1}{k} \sum_i c_i$, this can be obviated by reverting to the quadratic constraint

$$c^\top X X^\top c = c^\top X \left(\sum_{j=1}^{k} \bar{\beta} v_j\right) = \bar{\beta} \left(1 - \frac{1}{k-1}(k-1)\right) c^\top e = 0 \ ,$$

where we use the fact that each row of X corresponds to one of the unit vectors $v_i \in \mathbb{R}^{k-1}$.

In this case, we obtain the following representation of the *k-equipartition problem* (cf. [78]):

$$\begin{aligned}\min_{X \in \mathbb{R}^{n \times (k-1)}} \quad & \frac{k-1}{2k} \operatorname{Tr}(X^\top L X) \\ \text{s.t.} \quad & c^\top X X^\top c = 0 \\ & (X X^\top)_{ii} = 1 \quad \text{for } i = 1, \ldots, n \\ & (X X^\top)_{ij} \in \{-\tfrac{1}{k-1}, 1\} \quad \forall\, i, j \ .\end{aligned} \tag{6.14}$$

Since in the unsupervised setting we consider here, fixed values for the "sizes" β_i of the individual parts are usually not available in advance, we will only investigate the k-equipartition case with a constant vector $\beta = \bar{\beta} e$ in the following sections. While spectral relaxation is generally based on the problem formulation (6.13), we will show how SDP relaxation techniques can be used to approximately solve both versions of the multiclass partitioning problem, (6.13) and (6.14).

[3] The vectors v_j can indeed be calculated by factorizing the matrix $A = \frac{k}{k-1} I_k - \frac{1}{k-1} E$: as A is positive semidefinite with e being the only eigenvector to the eigenvalue 0, the eigenvalue decomposition gives $A = \tilde{V} \Lambda \tilde{V}^\top = V V^\top$, with $V \in \mathbb{R}^{k \times (k-1)}$ containing the desired vectors as its rows [78].

6.2.2 Spectral Relaxation

Spectral relaxation approaches for partitioning a given data set directly into several groups are based on calculating multiple eigenvectors of the (normalized) Laplacian matrix L. This idea has recently been studied by many authors (see [4, 190, 134, 204, 25], and references therein). Although the proposed algorithms differ with respect to the computational details, they coincide in key aspects. Specifically, the combinatorial complexity of multiclass partitioning problems like (6.13) is always dealt with in two steps: first, a transformed formulation of the original partitioning problem is relaxed to an eigenvector problem by dropping several of the given constraints. In a second step, these constraints are taken into account again by projecting the eigenvectors appropriately, which yields an embedding of the n points into a k-dimensional subspace. Finally, a corresponding discrete solution close to the continuous solution of the relaxation is found by applying some clustering heuristic.

A detailed comparison of different methods from the literature is beyond the scope of this work. Nevertheless, we will next briefly explain how the two main steps of spectral relaxation approaches can be justified in the context of multiclass partitioning problems of the form (6.13). To this end, we consider the k-equipartition version of (6.13) with constant constraint vectors $c = e^n$ and $\beta = \frac{n}{k}e^k$. To derive a spectral relaxation for this problem representation, first observe that the columns of the indicator matrix X are orthogonal to each other: $X^\top X = \frac{n}{k}I$. Adding this redundant constraint to (6.13), and substituting $Z := \sqrt{\frac{k}{n}}X$, the problem is relaxed by dropping the other constraints:

$$\begin{aligned} z^*_{MPSR} := \min_{Z \in \mathbb{R}^{n\times k}} \; & \frac{n}{2k}\operatorname{Tr}(Z^\top L Z) \\ \text{s.t.} \quad & Z^\top Z = I \; . \end{aligned} \tag{6.15}$$

By virtue of Fan's Theorem (see Theorem A.4), a solution for this problem is given by $Z^* = V$, with the columns v_i of $V \in \mathbb{R}^{n\times k}$ containing the eigenvectors of L corresponding to the k smallest eigenvalues $\lambda_1(L) \leq \cdots \leq \lambda_k(L)$.[4] This results in an optimal value for (6.15) of $z^*_{MPSR} = \sum_{i=1}^k \lambda_i(L)$. The same result can be obtained for the multiclass normalized cut problem based on the normalized Laplacian matrix $L' = D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$ (see [204]).

Hence, the first step in spectral relaxation approaches requires to compute the k smallest eigenvectors v_i of L (or L', respectively). In practice, often more than k eigenvectors are calculated, since it has been observed that better partitionings can be obtained in this way [4]. Observing that the *rows* z_i of the solution matrix $Z^* = V$ yield an embedding of the original image elements into the space $\mathbb{R}^k$, the second constraint in (6.13) can now be taken into account by normalizing these rows z_i, which corresponds to projecting them onto the k-dimensional unit-sphere: $z'_i = \frac{z_i}{\|z_i\|}$ (cf. [134, 204]).

[4] Note that the optimal solution Z^* is not unique: in fact, it is easy to show that each matrix VR with $R \in \mathbb{R}^{k\times k}$ being an arbitrary orthonormal matrix (i.e. $R^\top R = RR^\top = I$) results in the same objective value [204].

In order to finally derive a corresponding discrete solution, the points z'_i are partitioned into k groups by applying some heuristic clustering procedure. For this final step, different methods have been proposed in the literature, like:

- Apply the k-means algorithm on the points z'_i to minimize the distortion [118, 134].
- Partition the points z'_i based on the enclosed directional angle [29].
- Compute a linear ordering of the points z'_i and derive a multiclass partitioning by finding $k-1$ splitting points in this ordering [4].
- Iteratively seek better solutions by using alternating projections [25] or by alternately rotating and discretizing the matrix Z' [204].

Considering computational aspects, spectral relaxation is a convenient technique to find approximate solutions for the multiclass partitioning problem. However, from the theoretical point of view, this method exhibits two weak points: first, the relaxation (6.15) is not very strong, since many constraints of the original problem are dropped. In fact, a lower bound on the optimal multiway cut (6.13) that is better than z^*_{MPSR} has already been presented by Donath and Hoffman [47]:

$$z^*_{MP} \geq z_{DH} := \max_{e^\top v = 0} \frac{1}{2} \sum_{i=1}^{k} \beta_i \lambda_i (L + \mathrm{Diag}(v)) \; . \tag{6.16}$$

Note that this bound generalizes the spectral bound for the binary partitioning problem given in (4.13). In the next section, we will see that even stronger bounds can be obtained by reverting to semidefinite relaxations of (6.13); also see [153].

Second, the clustering heuristic used to find a discrete solution usually is no longer based on the original objective criterion. In fact, for the normalized cut it can be shown [155] that standard embedding techniques always yield a grouping problem in the corresponding vector space which is not equivalent to the original problem. Therefore, it is not clear how good the final solution really is.

6.2.3 Semidefinite Relaxation

Semidefinite programming relaxations of the multiclass graph (equi-)partitioning problem (with constant constraint vector $c = e$) have been researched quite thoroughly in recent years, based on both problem formulations (6.13) and (6.14) [153, 96, 197]. Since the extension of the results to the more general case with arbitrary constraint vector $c \neq e$ considered in this work is straightforward, we will only present the final results here and refer to the corresponding literature instead.

SDP Relaxation of (6.13)

An SDP relaxation of the unsupervised partitioning problem based on the representation (6.13) can be obtained in the same way as was presented for the restoration problem (6.2) in Section 6.1.2. To this end, we first equivalently reformulate (6.13) by representing the constraints in a quadratic form:

$$\begin{aligned} z^*_{MP} = \min_{X\in\mathbb{R}^{n\times k}} \; & \frac{1}{2}\operatorname{Tr}(LXX^\top) \\ \text{s.t.} \quad & \|Xe^k - e^n\|^2 = 0 \\ & \|X^\top c - \beta\|^2 = 0 \\ & X_{ij}^2 - X_{ij} = 0 \quad \forall\, i,j\ . \end{aligned} \tag{6.17}$$

Comparing this with (6.3), we immediately see (by setting $C = 0$ and $\tilde{M} = \frac{1}{2}L$) that the only difference is the additional balancing constraint present in (6.17). Since this can be taken into account during Lagrangian relaxation in a similar way as the first constraint (see [197]), we analogously to the relaxation (6.10) of the restoration problem obtain the following SDP relaxation of the multiclass partitioning problem (6.13):

$$\begin{aligned} s^*_1 := \min_{Y\succeq 0} \quad & \tilde{L}\bullet Y \\ \text{s.t.} \quad & A_0\bullet Y = 1 \\ & A_i\bullet Y = 0 \quad \text{for } i = 1,\dots,nk \\ & F\bullet Y = 0 \\ & F_2\bullet Y = 0\ , \end{aligned} \tag{6.18}$$

where F and the A_i-matrices are defined as in (6.7) and (6.9), and

$$\tilde{L} := \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2}I_k\otimes L \end{pmatrix}\ ,$$

$$F_2 := \begin{pmatrix} \beta^\top\beta & -(\beta\otimes c)^\top \\ -\beta\otimes c & I_k\otimes cc^\top \end{pmatrix}\ .$$

More precisely, in the equipartition case with constant constraint vector $\beta = \bar{\beta}e^k$ considered here we have $(F_2)_{11} = \frac{1}{k}(\sum_i c_i)^2$ for the first entry of F_2.

Concerning the solvability of the SDP relaxation (6.18), the same propositions hold as were stated in Section 6.1.2 for the multiclass restoration problem. Especially, the problem matrix Y can again be interpreted as a relaxation of the rank one matrix obtained from the vectorized matrix X (cf. (6.11)):

$$Y = \begin{pmatrix} 1 \\ \operatorname{vec}(X) \end{pmatrix}\left(1, \operatorname{vec}(X)^\top\right)\ .$$

Hence, as suggested in Section 6.1.3, a combinatorial solution can be computed from the first column (or row) Y_1 of the optimal solution Y^* of (6.18): starting with the second entry, find the largest entry in each block of length k, and set

the corresponding entry of the indicator matrix X to one. Future work will show if better discrete solutions can be obtained by other rounding techniques.

Finally note that the size of the problem matrix Y for the relaxation (6.18) is $(nk+1) \times (nk+1)$, which makes this approach only applicable for problems of moderate size. Yet we will show next that a convenient relaxation of smaller size $n \times n$ is obtained by resorting to the problem formulation (6.14).

SDP Relaxation of (6.14)

In order to derive a semidefinite relaxation for the representation (6.14) of the multiclass partitioning problem, first observe that the matrix X in (6.14) only occurs in the quadratic form $XX^\top$. This suggests to directly replace the positive semidefinite matrix $XX^\top \in \mathcal{S}_+^n$ of rank $k-1$ with an arbitrary matrix $Y \in \mathcal{S}_+^n$, which results in the following SDP relaxation (cf. [78]):

$$\begin{aligned} s_2^* := \min_{Y \succeq 0} \quad & \frac{k-1}{2k} L \bullet Y \\ \text{s.t.} \quad & cc^\top \bullet Y = 0 \\ & \operatorname{diag}(Y) = e \\ & Y_{ij} \geq -\frac{1}{k-1} \quad \forall\, i,j \ . \end{aligned} \tag{6.19}$$

In comparison to the binary case (4.7), this relaxation may be interpreted as a straightforward generalization: the only difference is the additional inequality constraint on the entries of Y. However, this constraint is of essential importance: if it were dropped, the entries of Y were admitted to take values between -1 and $+1$, which would result in a much weaker relaxation. In contrast, note that the corresponding constraint $X_{ij} \geq -1$ in the binary case (4.7) is redundant, since the matrix entries are automatically restricted to the interval $[-1,+1]$ due to the constraint on the diagonal entries of X.

By applying the above relaxation approach to the problem formulation (6.13), Karisch and Rendl [96] derive a lower bound on the objective value that is equivalent to (6.19). Besides presenting relations between several other SDP relaxations of the equipartition problem, they also show that (6.19) represents a stronger bound than the spectral bound (6.16) of Donath and Hoffman [47]:

$$z_{MP}^* \geq s_2^* \geq z_{DH} \ .$$

Concerning the solvability of the SDP relaxation (6.19), we have to deal with two difficulties: on the one hand, this semidefinite program obviously has no strictly interior solution, since the first constraint requires c to be an eigenvector of Y with eigenvalue zero. On the other hand, due to the additional inequality constraints on the entries of Y we can no longer use one of the standard SDP solvers that are applicable for problems of the general form (4.1). For these reasons, we employ the spectral bundle method of Helmberg and Rendl [79] to solve the SDP relaxation (6.19), which is able to handle semidefinite programs of this type.

In order to finally derive a combinatorial solution from the optimal solution matrix Y^*, we apply the following generalized version of the binary randomized hyperplane technique [55]:

1. Compute the factorization $Y^* = VV^\top$ (e.g. by Cholesky decomposition, see Theorem A.6) with the rows of $V \in \mathbb{R}^{n \times n}$ corresponding to unit vectors $v_j \in \mathbb{R}^n$.

2. Generate k independent random vectors $r_1, \ldots, r_k \in \mathbb{R}^n$ from the unit sphere by choosing from the standard normal distribution with mean 0 and variance 1. Each r_i then represents one group i of the partitioning.

3. Assign each element j to the group i' for which the corresponding random vector $r_{i'}$ is closest to v_j, i.e. for which $v_j^\top r_{i'} = \max_i v_j^\top r_i$.

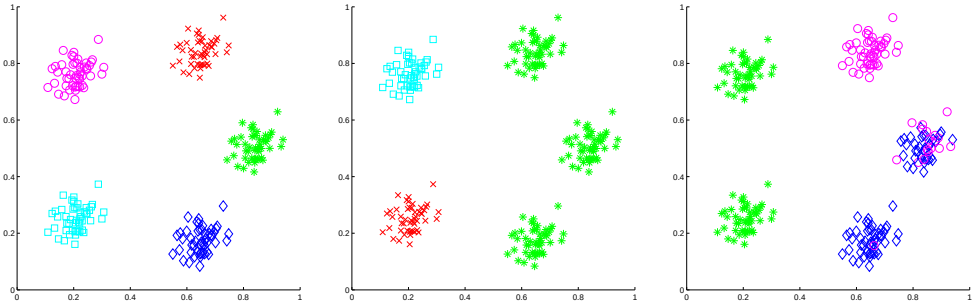
For $k = 2$, this algorithm reduces to the binary randomized hyperplane technique presented in Section 4.2.3. As the probability that two elements j_1 and j_2 are assigned to the same set only depends on their inner product $v_{j_1}^\top v_{j_2}$, it is possible to prove strong performance guarantees of this algorithm, at least for the max-cut problem [55]. However, due to the additional equipartition constraint present in (6.14), these bounds do not apply here.

The final combinatorial solution is obtained by applying the above randomized hyperplane technique multiple times and picking the segmentation that yields the best objective value for (6.14). In this context, note that like for the binary case, we do not enforce the balancing constraint $X^\top c = \beta$ for the combinatorial solution, which may result in partitionings that contain a lower number of groups than is specified by k. However, this is no drawback, since the "correct" k is usually unknown for *unsupervised* partitioning tasks. This insight further justifies the use of the balancing constraint, which rather serves as a bias to guide the search than as a strict requirement.

6.2.4 First Experimental Results

In this section, we present some experimental results obtained with the different spectral and SDP relaxations of the multiclass partitioning problem. Since the corresponding research is still in progress, the results have to be considered as preliminary, but a couple of important observations can be given nevertheless.

A first experiment is depicted in Figure 6.7: for this simple point set consisting of five identical clusters, the similarity matrix is computed based on the Euclidean distances of the points (cf. Section 4.4.2, method (i)). Setting $k = 5$, we obtain the correct partitioning if this cluster number is enforced for the final result (Figure 6.7, left). While for spectral relaxation based on the normalized cut criterion, this is achieved by applying the k-means algorithm on the normalized row vectors of the eigenvector matrix V (see Section 6.2.2), we find a corresponding solution from the result of the SDP relaxation (6.19) by only accepting those final clusterings computed with the randomized hyperplane technique that comprise the desired number of groups. If this requirement is omitted for rounding the SDP solution, a segmentation into fewer clusters with a lower objective value is found (Figure 6.7, middle).



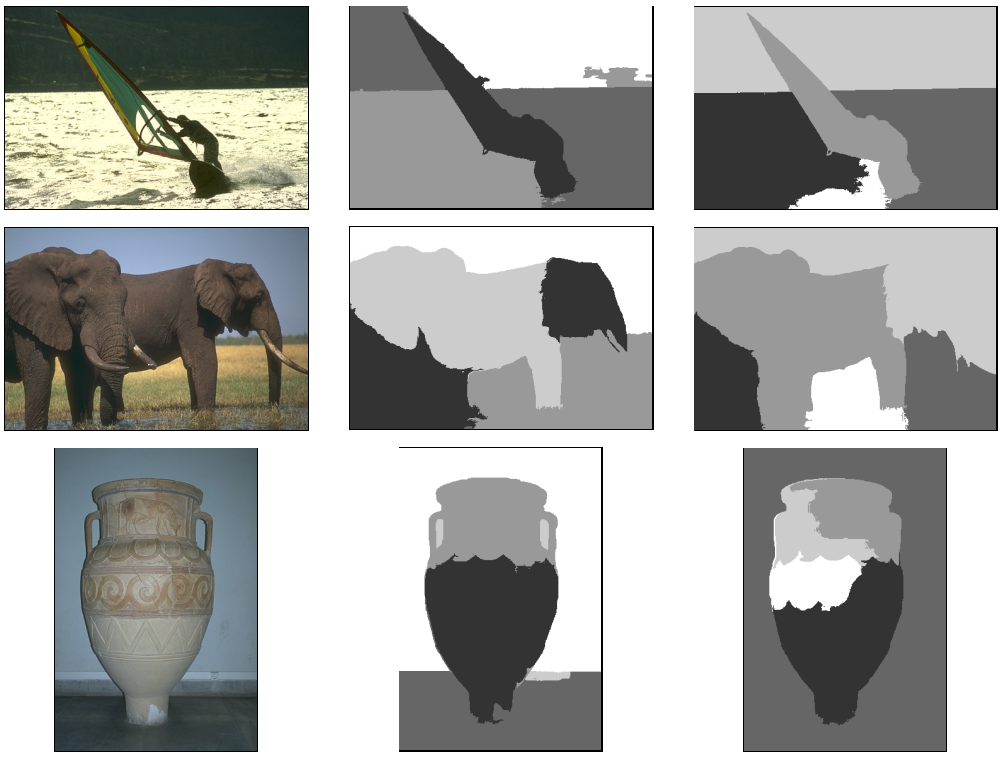
Spectral/SDP relax., enforce 5 clusters | SDP relaxation (6.19) | SDP relaxation (6.18)

Figure 6.7: Simple multiclass clustering example. For $k = 5$, both the SDP relaxation (6.19) and the normalized cut relaxation (with k-means) are able to find the correct partitioning, if we enforce the right number of clusters (left). Otherwise, the SDP relaxation yields less clusters, since the corresponding objective value is lower (middle). In contrast to that, the SDP relaxation (6.18) is too weak to reasonably partition this dataset (right).

We also see that the SDP relaxation (6.18) of the multiclass partitioning problem (6.13) does not give a reasonable result (Figure 6.7, right). A closer look at the first column of the corresponding solution matrix Y reveals the reason for this behavior: since the entries are nearly constant ($Y_{1,i} \approx 0.2$ for all i), they do not yield valuable suggestions for the corresponding indicator vectors. Hence, this relaxation seems to be too weak; in fact, Wolkowicz and Zhao [197] use an additional constraint (the so-called "gangster operator") and a projection technique to arrive at a stronger relaxation. Future work will show if this will also lead to better results for the application considered here. Due to the above result, however, we will no further consider the SDP relaxation (6.18) in this section.

We also apply the multiclass partitioning relaxations to a few real images from the Berkeley segmentation dataset [121]. In order to reduce the problem size, these images are first preprocessed with the mean shift technique, and a graph representation is constructed by calculating similarity values w_{ij} only for neighboring patches i and j (cf. Section 5.2.4). Moreover, we again use the corresponding patch-sizes as entries of the constraint vector c in the equipartition problem (6.14). Since this requires each segment to contain $\frac{n}{k}$ pixels (with n denoting the total number of pixels), the resulting problem may have no feasible solution, e.g. if the largest patch comprises more than $\frac{n}{k}$ pixels. For this reason, we eliminate such large patches before applying the SDP relaxation (6.19), and add them afterwards as a single cluster.

By generally setting $k = 5$, we obtain the segmentations depicted in Figure 6.8. Considering that we did not elaborate on the similarity values, the number of clusters or more sophisticated techniques to compute a combinatorial solution, the results are quite promising: important parts of the images are separated from each other. The corresponding spectral solutions given in Figure 6.8, right, are in this case based on the normalized cut criterion by calculating the 5 smallest eigenvectors of the normalized Laplacian, and applying k-means



Original image SDP relaxation (6.19) Spectral relaxation

Figure 6.8: Multiclass segmentations obtained with spectral and SDP relaxation, respectively. For these examples, we generally set $k = 5$. Note that the SDP relaxation may yield segmentations into fewer clusters, while the normalized cut relaxation with k-means always results in 5 parts.

on the scaled rows of the matrix containing these eigenvectors as columns.

Concerning the computational effort to derive these results, it only takes a few seconds to find the normalized cut solution. In contrast to that, we generally stop the spectral bundle method to solve the SDP relaxation after 60 minutes, since the approximate solution obtained after that time is sufficiently close to the optimum to calculate the corresponding combinatorial solution.

Chapter 7

Conclusion

7.1 Summary

The main topic of this thesis is to introduce a novel optimization technique to the field of computer vision. The resulting semidefinite programming framework can be applied to a broad class of combinatorial optimization problems which arise naturally in early and mid-level computer vision, like unsupervised image segmentation, image restoration or perceptual grouping tasks. The investigations show that the SDP relaxation approach is an attractive alternative to established optimization techniques, especially due to its sound underlying mathematical principles and the absence of tuning parameters.

Comparison of Spectral Techniques

As they are closely related to our SDP relaxation approach, we study different spectral relaxation methods for graph-based image segmentation problems in Section 3.1. In particular, we present a general framework to define suitable measures for the quality of a segmentation which is based on scaled graph cuts. Minimizing these measures is NP-hard, but a general relaxation can be derived that leads to the computation of the second smallest eigenvector of a matrix related to the Laplacian matrix L of the underlying graph. A corresponding binary solution of the original combinatorial problem is then obtained by applying an appropriate thresholding technique.

As special cases, this general framework comprises the average cut approach [74, 159] resulting in the computation of the Fiedler vector [51], and the normalized cut approach of Shi and Malik [168]. We show how the corresponding relaxations are connected to other graph theoretical measures, like the isoperimetric number or the Cheeger constant. Moreover, we consider the average association as the analogon to the normalized association measure, which is equivalent to the normalized cut. Since such an equivalence does not hold for the average cut, we derive an appropriate relaxation for the average association measure, which results in computing the second largest eigenvector of the centered similarity matrix $\tilde{W}$. This contrasts weaker relaxations presented in the literature in this context [142, 168], which rather yield relaxations to a foreground association measure instead.

Several experiments confirm the theoretical findings:

- The solutions for the spectral relaxation approaches can be computed in short time (less than a minute for a few thousand pixels).
- If the degrees of the graph vertices do not differ considerably, the average and normalized cut approaches yield very similar partitioning results.
- Concerning the choice of the similarity measure, the cut approaches are more robust than the average association criterion, which therefore has to be considered a weaker partitioning technique.
- In certain situations (e.g. when only one part of the image gives a high inner association measure), however, the foreground association criterion is more appropriate and yields better clustering results.

Semidefinite Relaxation of Binary Optimization Problems

In Chapter 4, we present the core of this thesis: a semidefinite relaxation approach that can be applied to general binary minimization problems comprising a quadratic objective function that is subject to an additional linear constraint. The resulting semidefinite program has several favorable properties from both the mathematical as well as the computational point of view (cf. Section 1.1.1), which e.g. enable the computation of the optimal solution to arbitrary precision in polynomial time. We prove that such an optimal solution for the relaxation exists under mild conditions (balanced constraint vector), and that it yields a tighter lower bound on the minimum of the original optimization problem than spectral relaxation with the Fiedler vector.

Subsequently, a corresponding combinatorial solution is obtained based on a randomized approximation technique. We discuss interpretations of this approximation procedure, and provide performance bounds on the final solution that hold under certain conditions (missing linear constraint). Numerous applications to diverse computer vision problems approve the advantages of the SDP relaxation approach:

- Ground-truth experiments on the restoration of one-dimensional signals show that in practice, the SDP relaxation yields much better results than are assured by the theoretical performance bounds. For most alternative optimization approaches, however, similar bounds are missing completely.
- Spectral relaxation may result in unsatisfactory partitionings when no appropriate threshold value is found or chosen. In contrast to that, the solution obtained from the SDP relaxation by randomized rounding does not depend on tuning such a parameter.
- The SDP relaxation approach works for a wide range of optimization problems, where spectral methods fail (like separating dense foreground from sparse background) or cannot be applied (like perceptual grouping or image restoration tasks).

- The general balancing constraint allows us to appropriately account for varying importance of the graph vertices, as it may be required when the extracted image elements do not correspond to pixels (but e.g. to patches of differing size). Such an adjustment is not possible for spectral relaxation approaches.

- For perceptual grouping problems that have been considered as difficult [192] due to the complexity of the underlying global optimization criterion [84], convenient solutions are computed in only a few seconds.

- In comparison to the local greedy ICM algorithm [16], the global SDP approach yields a much tighter relaxation of the underlying combinatorial problem. Hence, the corresponding image restoration results are much better.

Efficient Graph-Based Image Segmentation

The price for the convenient properties of the SDP relaxation is the squared number of variables that are involved in the optimization approach. While this is no problem for perceptual grouping tasks with a few hundred image primitives, it prevents at present the application to large-scale problems as they naturally arise for real world image segmentation problems on a pixel basis. Although there exist SDP solvers [12, 194] that are able to exploit a sparse problem structure (as it is present for image restoration or segmentation problems represented by locally connected graphs), memory requirements and computation times grow quickly with the number of variables. Experiments show that currently, problems with up to $10,000$ variables can reasonably be solved (cf. Section 4.4.6).

In Chapter 5, we therefore present two different methods which efficiently reduce the problem size for binary unsupervised partitioning tasks. The first approach is based on a preprocessing step which computes an over-segmentation of the image by applying the mean shift technique [33] at a fine spatial scale. Instead of thousand of pixels, we then use the obtained image patches (or "superpixels") as the basic elements to define the corresponding graph representation of the image. The second method pursuits the idea of probabilistic sampling: by randomly selecting a small number of pixels from the image, we obtain an optimization problem of small scale, the solution of which can conveniently be generalized to a complete segmentation afterwards.

Experimental results reveal the success of these approaches:

- The over-segmentation reduces the problem size by several orders of magnitude (to less than 0.01% of the pixel-based graph), so that a solution based on the SDP relaxation can be computed in short time (less than one minute). For the probabilistic sampling approach, it is sufficient to select less than 0.5% of the pixels to obtain convenient segmentations.

- Using small image patches instead of pixels leads to a more natural image representation — the pixels are merely the result of the digital image discretization process, and do not occur in the real world. Moreover, in

comparison to pixel-based partitioning methods, smoother final segmentations are obtained.

- The varying size of the superpixels can be taken into account appropriately by adjusting the balancing constraint of the SDP relaxation approach. This yields results of superior quality in comparison to the normalized cut relaxation, which is approved quantitatively by comparing the obtained partitionings to image segmentations produced by humans [121].
- For the SDP relaxation approach, probabilistic sampling drastically reduces the computational effort so that it becomes comparable to solving the normalized cut relaxation. In fact, the most time consuming step consists in the final calculation of the corresponding binary solutions, which are by orders of magnitude larger than the solutions of the sampling-based optimization problems. Due to the reduced effort it is possible to compute several solutions based on different samplings, and to combine them or to pick the best one in order to derive the final segmentation.

Extension to Non-Binary Problems

The SDP relaxation approach presented in Chapter 4 is suited for binary problems only. In practice, however, the decomposition of an image into more than two parts is often desired. A straightforward extension suitable for unsupervised partitioning tasks consists in a hierarchical application of the binary method: by recursively computing two-way partitions, a segmentation into multiple parts is obtained (Section 5.1).

In Chapter 6, we follow an alternative idea by presenting direct multiclass extensions for the image restoration and the unsupervised segmentation problem, respectively. Fixing the number k of desired classes, natural modifications of the corresponding problem formulations are derived in a first step. Based on an indicator matrix $X \in \mathbb{R}^{n \times k}$, we obtain different combinatorial optimization problems for both vision tasks, which involve quadratic objective functions that are subject to several constraints. Using Lagrangian relaxation, we show that in both cases suitable SDP relaxations can be defined. However, since either the corresponding problem matrices are very large now ($nk \times nk$ entries) or a high number ($\frac{1}{2}n(n-1)$) of additional inequality constraints is introduced, solving these semidefinite programs becomes much more elaborate than for the binary case.

Nevertheless, first experimental results are promising:

- For binary restoration problems ($k = 2$), ground-truth experiments show that the multiclass SDP relaxation and the original binary SDP relaxation yield identical objective values. Moreover, the corresponding combinatorial solutions are very similar, with an error of less than 2%. The difference is caused by the employed rounding techniques: whereas the binary SDP relaxation uses a randomized approximation, the multiclass SDP relaxation finds a combinatorial solution directly by comparing certain entries of the solution matrix.

- For small multiclass image restoration problems, high quality solutions are obtained. However, the required computational effort is quite demanding: for an image with $n = 1369$ pixels, it takes more than 3.5 hours to compute the reconstruction.

- For unsupervised segmentation problems, the number k of parts present in the image is usually unknown in advance. This fact favors the hierarchical approach, which allows us to select the final number of segments during the partitioning process. Moreover, the resulting coarse-to-fine hierarchy yields similar segmentations of varying granularity, which seems to be close to human perceptual organization.

- Concerning direct multiclass segmentation, two different SDP relaxations are presented: the first one is applicable to general partitioning problems, but only yields unsatisfactory results which is most likely due to an insufficiently tight relaxation. In contrast, the second approach is restricted to equipartition problems. The segmentations obtained with this method are promising, especially since a modification of the binary randomized approximation technique computes final combinatorial solutions that are allowed to consist of less than the specified number k of segments. In this way, the correct choice of k becomes less critical.

7.2 Future Work

So far, the focus of our work has primarily been on analyzing the mathematical characteristics of the SDP relaxation approach that arise in connection with combinatorial optimization problems in computer vision: Lagrangian relaxation, duality, feasibility issues, performance bounds, and comparison to spectral relaxation. Moreover, we investigated topics regarding modifications of the SDP relaxation to make it more efficient, and possible extensions to non-binary problems. Although we demonstrated the applicability of our approach for several non-trivial computer vision tasks, there are still several issues that suggest further research:

- An important topic in the context of unsupervised segmentation is the derivation of suitable similarity measures. Up to now, we mostly relied on quite elementary color and texture cues, without working out tailor-made metrics for specific applications. Accordingly, a significant aspect of our future work will consist in defining more elaborate similarity measures that are adequate in the SDP relaxation framework. This especially includes the related problem of learning such suitable measures for classification, which can also be handled with semidefinite programming techniques (cf. [109]).

- The binary optimization problems considered in this thesis involve one linear constraint at most. However, there is neither a limit on the number of constraints for the SDP relaxation approach, nor is it restricted to linear constraints; in fact, quadratic constraints are even more suitable

to be included. In this context, note that other optimization approaches like spectral relaxation do not permit considering additional constraints. Future work will show in how far this generality of the SDP relaxation method allows us to tackle other computer vision tasks as well.

- In connection with graph partitioning, we already indicated that the SDP relaxation approach can also be used to solve partly supervised segmentation tasks by adjusting the constraint variable β (see Section 4.4.3). In this context, note that the membership of certain points to the same cluster can also conveniently be modeled by additional quadratic constraints on the entries of x: for instance, $x_i x_j = 1$ enforces i and j to belong to the same cluster. For this reason, the problem of semi-supervised segmentation is another interesting topic for future research. In recent related work [135], a similar semidefinite programming approach is proposed for the related problem of graph partitioning with preferences.

- Concerning the sampling-based version of the SDP relaxation method presented in Section 5.3, there are several aspects which suggest future work. On the one hand, it would be interesting to study whether better segmentations could be obtained if the sampled pixels were selected in a more sophisticated way than just randomly. On the other hand, the idea seems to be promising to calculate several partitionings based on different samplings which then can be combined to produce a stable final segmentation (cf. [176]). Furthermore, smoother results can certainly be obtained by applying an appropriate postprocessing step.

- The investigation of direct multiclass SDP relaxations in Chapter 6 has just been started, but first preliminary results are promising. Future research will show if stricter relaxations (cf. [206, 197]) give better restorations and image segmentations, respectively. Moreover, it would be interesting to develop algorithms that are able to handle the corresponding large problem sizes appropriately.

Appendix A

Symmetric and Positive Semidefinite Matrices

This appendix collects some important facts about symmetric and positive semidefinite matrices that are used throughout this thesis, and gives references to where the corresponding proofs can be found. For further information, we refer to the standard literature on linear algebra and matrix analysis, like [87, 175, 69, 205], or books on semidefinite programming [78, 196].

In general, the space of symmetric matrices $\mathcal{S}^n$ can be interpreted as a vector space in $\mathbb{R}^{\binom{n+1}{2}}$, with the natural inner product between two matrices $A, B \in \mathcal{S}^n$ defined as

$$A \bullet B = \operatorname{Tr}(A^\top B) = \sum_{i,j=1}^{n} A_{ij} B_{ij} \ .$$

A symmetric matrix A is completely characterized by the solutions of the linear equations

$$Aq = \lambda q \ ,$$

which are given by its eigenvalues and eigenvectors:

Theorem A.1 (Eigenvalue decomposition). *All eigenvalues $\lambda_i(A)$ of a symmetric matrix $A \in \mathcal{S}^n$ are real. There exists an orthonormal matrix $Q \in \mathbb{R}^{n\times n}$, $QQ^\top = Q^\top Q = I$, such that*

$$A = Q\Lambda Q^\top = \sum_{i=1}^{n} \lambda_i q_i q_i^\top \ ,$$

where $\Lambda = \operatorname{Diag}(\lambda_1(A), \ldots, \lambda_n(A)) \in \mathcal{S}^n$ is a diagonal matrix containing the eigenvalues $\lambda_i(A)$ on its main diagonal. The columns q_i of Q comprise the corresponding eigenvectors of unit length.

Proof. See [69, Theorem 8.1.1]. □

Obviously, the eigenvalue decomposition and the *singular value decomposition (SVD)* [69, Theorem 2.5.2] are very similar for symmetric matrices:

Lemma A.2. *Denote the SVD of the symmetric matrix $A \in \mathcal{S}^n$ by*

$$A = U\Sigma V^\top = \sum_{i=1}^{n} \sigma_i u_i v_i^\top ,$$

where $\Sigma = \mathrm{Diag}(\sigma_1, \ldots, \sigma_n)$ is a diagonal matrix containing the singular values σ_i of A on its main diagonal. The columns u_i of U and v_i of V comprise the corresponding left and right singular vectors of unit length, respectively.

Using the eigenvalue decomposition of A from Theorem A.1, appropriate sorting of the eigenvalues results in $\sigma_i = |\lambda_i|$, with the corresponding singular vectors and eigenvectors satisfying $u_i = v_i = q_i$ for $\lambda_i \geq 0$ and $u_i = -v_i = q_i$ for $\lambda_i < 0$, respectively.

An important characterization of the eigenvalues of a symmetric matrix is based on a Rayleigh quotient formulation:

Theorem A.3 (Courant-Fischer Minimax Theorem). *Let $\lambda_1(A) \leq \cdots \leq \lambda_n(A)$ denote the eigenvalues of $A \in \mathcal{S}^n$. Then*

$$\lambda_k(A) = \min_{\substack{U \subset \mathbb{R}^n \\ \dim(U)=k}} \max_{0 \neq v \in U} \frac{v^\top A v}{v^\top v} .$$

More generally, for the eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ of the general eigenvalue problem $Aq = \lambda Bq$ with positive definite B we have

$$\lambda_k = \min_{\substack{U \subset \mathbb{R}^n \\ \dim(U)=k}} \max_{0 \neq v \in U} \frac{v^\top A v}{v^\top B v} .$$

Proof. See e.g. [69, Theorem 8.1.2] (special case), and [175, Corollary VI.1.16] (general case). □

Especially, we get the Rayleigh-Ritz principles

$$\lambda_1(A) = \min_{0 \neq v \in \mathbb{R}^n} \frac{v^\top A v}{v^\top v} \qquad \text{and} \qquad \lambda_n(A) = \max_{0 \neq v \in \mathbb{R}^n} \frac{v^\top A v}{v^\top v}$$

for $k = 1$ and $k = n$, respectively [87, Theorem 4.2.2].

A generalization of these characterizations to the sum of the smallest eigenvalues is given by the following theorem:

Theorem A.4 (Fan's Theorem). *Let $\lambda_1(A) \leq \cdots \leq \lambda_n(A)$ denote the eigenvalues of $A \in \mathcal{S}^n$. Then*

$$\sum_{i=1}^{k} \lambda_i(A) = \min_{\substack{V \in \mathbb{R}^{n \times k} \\ V^\top V = I}} \mathrm{Tr}(V^\top A V) ,$$

and the minimum is attained for $V = (q_1, \ldots, q_k)$ containing the eigenvectors corresponding to $\lambda_1(A), \ldots, \lambda_k(A)$ as columns.

Proof. For a short proof see [136]. Fan's Theorem is also a special case of the more general Wielandt principle, which is proven e.g. in [175, Theorem 4.5]. □

To recall the general definition, a matrix $A \in \mathcal{S}^n$ is called *positive semidefinite* (denoted by $A \in \mathcal{S}^n_+$ or $A \succeq 0$), if $x^\top Ax \geq 0$ for all $x \in \mathbb{R}^n$. If strict inequality holds for all $x \neq 0$, A is called *positive definite*.

The following theorem collects several equivalent characterizations for positive semidefinite matrices:

Theorem A.5 (Positive semidefiniteness). *For $A \in \mathcal{S}^n$ the following statements are equivalent:*

- *A is positive semidefinite: $A \in \mathcal{S}^n_+$.*
- *$x^\top Ax \geq 0$ for all $x \in \mathbb{R}^n$.*
- *The eigenvalues of A are nonnegative: $\lambda_i(A) \geq 0$ for $i = 1, \ldots, n$.*
- *$A \bullet B \geq 0$ for all $B \in \mathcal{S}^n_+$ (Fejer's Theorem).*
- *The determinant of every principal submatrix (a submatrix obtained by deleting rows and corresponding columns) of A is nonnegative.*
- *$A = VV^\top$ for some matrix $V \in \mathbb{R}^{n \times m}$, with* $\operatorname{rank}(V) = \operatorname{rank}(A)$.

Proof. See [78, Theorem 1.1.13 and Corollary 1.2.7], and [205, Theorem 6.2]. □

The last statement in this theorem includes the following special case:

Theorem A.6 (Cholesky decomposition). *For $A \in \mathcal{S}^n_+$, there exists a lower triangular matrix $V \in \mathbb{R}^{n \times n}$ with nonnegative diagonal entries such that $A = VV^\top$. If A is positive definite, then this decomposition is unique.*

Proof. See e.g. [78, Theorem 1.1.10]. □

A different special case is given by the root of a positive semidefinite matrix:

Theorem A.7 (Square root). *For $A \in \mathcal{S}^n_+$, there exists a unique positive semidefinite matrix $C \in \mathcal{S}^n_+$ such that $A = CC$. This* square root *of A is also denoted as $C = A^{\frac{1}{2}}$.*

Proof. See e.g. [87, Theorem 7.2.6]. □

Another important characterization of the positive semidefiniteness of a matrix is obtained by considering a block decomposition:

Theorem A.8 (Schur complement). *Suppose that the symmetric matrix $M \in \mathcal{S}^n$ can be partitioned into*

$$M = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix},$$

with $A \in \mathcal{S}^m$ being positive definite, $C \in \mathcal{S}^{n-m}$ and $B \in \mathbb{R}^{m \times n-m}$. Then M is positive semidefinite if and only if $C - B^\top A^{-1} B \in \mathcal{S}^n_+$.

Proof. See e.g. [87, Theorem 7.7.6]. □

The simplest positive semidefinite matrices are the rank one matrices $zz^\top$:

Lemma A.9. *Let $z \in \mathbb{R}^n$ denote an arbitrary vector. Then the rank one matrix $zz^\top$ is positive semidefinite with the only nonzero eigenvalue $\lambda_n = z^\top z$.*

Proof. Since $x^\top(zz^\top)x = (x^\top z)^2 \geq 0$ for every vector $x \in \mathbb{R}^n$, the positive semidefiniteness of $zz^\top$ follows directly from the definition. Moreover, $(zz^\top)z = (z^\top z)z$, so that $z^\top z$ is an eigenvalue with the corresponding eigenvector z. As $zz^\top$ has rank one, this is the only nonzero eigenvalue. □

Finally, the following theorem gives a generalization of the fundamental Cauchy-Schwarz inequality for inner products based on positive semidefinite matrices:

Theorem A.10 (General Cauchy-Schwarz inequality). *For a positive semidefinite matrix $A \in \mathcal{S}_+^n$ we have*

$$|x^\top Ay|^2 \leq (x^\top Ax)(y^\top Ay) \; .$$

Proof. This is a special case of Theorem 8.4 in [205]. A direct proof is based on factorizing $A = VV^\top$, substituting $x' = V^\top x$ and $y' = V^\top y$ and applying the original Cauchy-Schwarz inequality. □

Bibliography

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proc. 33rd ACM Symposium on Theory of Computing (STOC)*, pages 611–618. ACM Press, 2001.

[2] H. Ackermann. Sampling matters for efficient clustering and image partitioning. Master's thesis, University of Mannheim, 2003.

[3] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optimization*, 5(1):13–51, 1995.

[4] C. J. Alpert, A. B. Kahng, and S.-Z. Yao. Spectral partitioning with multiple eigenvectors. *Discrete Applied Math.*, 90:3–26, 1999.

[5] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12(9):855–867, 1990.

[6] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *IEEE Trans. Patt. Anal. Mach. Intell.*, 20(2):168–185, 1998.

[7] C. T. H. Baker. *The Numerical Treatment of Integral Equations*. Clarendon Press, Oxford, 1977.

[8] A. Barbu and S.-C. Zhu. Graph partition by Swendsen-Wang cuts. In *Proc. 9th Int. Conf. Computer Vision (ICCV)*, pages 320–327. IEEE Comp. Soc., 2003.

[9] S. T. Barnard and H. D. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and Experience*, 6:101–107, 1994.

[10] A. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discr. Comput. Geom.*, 13(2):189–202, 1995.

[11] S. J. Benson, Y. Ye, and X. Zhang. Mixed linear and semidefinite programming for combinatorial and quadratic optimization. *Optimiz. Methods and Software*, 11&12:515–544, 1999.

[12] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

[13] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. *Proc. IEEE*, 76(8):869–889, 1988.

[14] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, 2nd edition, 1999.

[15] D. Bertsimas and Y. Ye. Semidefinite relaxations, multivariate normal distributions, and order statistics. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 3, pages 1–19. Kluwer Academic Publishers, 1998.

[16] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986.

[17] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[18] M. Bolla and G. Molnár-Sáska. Isoperimetric properties of weighted graphs related to the Laplace spectrum and canonical correlations. *Studia Sci. Math.*, 37:1–17, 2002.

[19] R. B. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Proc. 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 280–285. IEEE Computer Society Press, 1987.

[20] C. F. Borges. On the estimation of Markov random field parameters. *IEEE Trans. Patt. Anal. Mach. Intell.*, 21(3):216–224, 1999.

[21] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

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

[23] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 648–655, Santa Barbara, California, 1998.

[24] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[25] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In C. M. Bishop and B. J. Frey, editors, *Proc. 9th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Key West, Florida, 2003.

[26] N. Brixius and K. Anstreicher. Solving quadratic assignment problems using convex quadratic programming relaxations. *Optimization Methods and Software*, 16(1–4):49–68, 2001.

[27] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming, Series B*, 95(2):329–357, 2003.

[28] E. Çela. *The Quadratic Assignment Problem: Theory and Algorithms.* Kluwer Acad. Publishers, Dordrecht, 1998.

[29] P. Chan, D. Schlag, and J. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. on Computer-Aided Design*, 13(9):1088–1096, 1994.

[30] P. B. Chou and C. M. Brown. The theory and practice of Bayesian image labeling. *Int. J. Comp. Vision*, 4(3):185–210, 1990.

[31] N. Christofides. *Graph Theory: An Algorithmic Approach.* Academic Press, New York, 1975.

[32] F. R. K. Chung. *Spectral Graph Theory.* Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, Providence, RI, 1997.

[33] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(5):603–619, 2002.

[34] F. J. Cortijo and N. Pérez de la Blanca. Improving classical contextual classifications. *International Journal of Remote Sensing*, 19(8):1591–1613, 1998.

[35] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Electronic Computers*, 14:326–334, 1965.

[36] I. J. Cox, S. B. Rao, and Y. Zhong. "Ratio regions": A technique for image segmentation. In *Proc. International Conference on Pattern Recognition (ICPR)*, volume 2, pages 557–564, 1996.

[37] T. F. Cox and M. A. Cox. *Multidimensional Scaling.* Chapman & Hall, London, 1994.

[38] N. Cristianini, J. Shawe-Taylor, and J. Kandola. Spectral kernel methods for clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 649–655. MIT Press, 2002.

[39] D. M. Cvetkovic, M. Doob, and H. Sachs. *Spectra of graphs.* Barth, Heidelberg & Leipzig, Germany, 1995.

[40] E. de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, volume 65 of *Applied Optimization Series*. Kluwer Academic Publishers, 2002.

[41] C. Delorme and S. Poljak. Combinatorial properties and the complexity of a max-cut approximation. *European Journal of Combinatorics*, 14:313–333, 1993.

[42] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming, Series A*, 62(3):557–574, 1993.

[43] L. Devroye. Automatic pattern recognition: A study of the probability of error. *IEEE Trans. Patt. Anal. Mach. Intell.*, 10(4):530–543, 1988.

[44] M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer, Berlin, 1997.

[45] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274. ACM Press, 2001.

[46] Seventh DIMACS implementation challenge on semidefinite and related optimization problems. http://dimacs.rutgers.edu/Challenges/Seventh/, November 2–3, 2000.

[47] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[48] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proc. 10th Symp. Discrete Algorithms*, pages 291–299. ACM-SIAM, 1999.

[49] R. C. Dubes and A. K. Jain. Random field models in image analysis. *J. Applied Statistics*, 16(2):131–164, 1989.

[50] FEX-homepage. Inst. of Photogrammetry, University of Bonn, Germany, http://www.ipb.uni-bonn.de/ipb/projects/fex/fex.html.

[51] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(2):619–633, 1975.

[52] B. Fischer and J. M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 25(4):513–518, 2003.

[53] D. A. Forsyth, J. Haddon, and S. Ioffe. The joy of sampling. *Int. J. Comp. Vision*, 41(1/2):109–134, 2001.

[54] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):214–225, 2004.

[55] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18:67–81, 1997.

[56] A. M. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proc. on FOCS*, pages 370–378, 1998.

[57] K. Fujisawa, M. Kojima, and K. Nakata. Exploiting sparsity in primal-dual interior-point methods for semidefinite programming. *Mathematical Programming, Series A*, 79:235–253, 1997.

[58] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[59] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions of Information Theory*, 21(1):32–40, 1975.

[60] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Proc. 9th Int. Conf. Computer Vision (ICCV)*, pages 716–723. IEEE Comp. Soc., 2003.

[61] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Co., San Francisco, CA, 1979.

[62] Y. Gdalyahu, D. Weinshall, and M. Werman. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(10):1053–1074, 2001.

[63] D. Geman. Random fields and inverse problems in imaging. In P. L. Hennequin, editor, *École d'Été de Probabilités de Saint-Flour XVIII – 1988*, volume 1427 of *Lect. Notes in Math.*, pages 113–193. Springer, 1990.

[64] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(6):721–741, 1984.

[65] M. X. Goemans. Semidefinite programming in combinatorial optimization. *Mathematical Programming, Series A*, 79(1–3):143–161, 1997.

[66] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[67] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(4):377–388, 1996.

[68] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

[69] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[70] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[71] M. Groetschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 2nd edition, 1993.

[72] R. Grone, S. Pierce, and W. Watkins. Extremal correlation matrices. *Linear Algebra and its Applications*, 134:63–70, 1990.

[73] S. Guattery and G. L. Miller. On the quality of spectral separators. *SIAM Journal on Matrix Analysis and Applications*, 19(3):701–719, 1998.

[74] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design*, 11(9):1074–1085, 1992.

[75] U. Hahn and M. Ramscar, editors. *Similarity and Categorization*. Oxford Univ. Press, 2001.

[76] E. Halperin and U. Zwick. A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems. *Random Structures and Algorithms*, 20(3):382–402, 2002.

[77] F. Heitz, P. Perez, and P. Bouthemy. Multiscale minimization of global energy functions in some visual recovery problems. *Comp. Vis. Graph. Image Proc.: Image Understanding*, 59(1):125–134, 1994.

[78] C. Helmberg. Semidefinite programming for combinatorial optimization. ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin, October 2000. Habilitationsschrift, TU Berlin.

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

[80] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6(2):342–361, 1996.

[81] C. Helmberg, F. Rendl, and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, 4(2):197–215, 2000.

[82] C. Helmberg and R. Weismantel. Cutting plane algorithms for semidefinite relaxations. In P. M. Pardalos and H. Wolkowicz, editors, *Topics in Semidefinite and Interior-Point Methods*, volume 18 of *Fields Institute Communications*, pages 197–213. AMS, 1998.

[83] B. Hendrickson and R. Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM Journal on Scientific Computing*, 16(2):452–469, 1995.

[84] L. Hérault and R. Horaud. Figure-ground discrimination: A combinatorial optimization approach. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(9):899–914, 1993.

[85] L. Hermes, D. Frieauff, J. Puzicha, and J. M. Buhmann. Support vector machines for land usage classification in Landsat TM imagery. In *Proc. of the IEEE International Geoscience and Remote Sensing Symposium*, volume 1, pages 348–350, Hamburg, 1999.

[86] T. Hofmann and J. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(1):1–14, 1997.

[87] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1986.

[88] L. Hubert-Moy, A. Cotonnec, L. Le Du, A. Chardin, and P. Pérez. A comparison of parametric classification procedures of remotely sensed data applied on different landscape units. *Remote Sensing of Environment*, 75(2):174–187, 2001.

[89] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 125–131, 1998.

[90] D. W. Jacobs. Robust and efficient detection of salient convex groups. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(1):23–37, 1996.

[91] I. H. Jermyn and H. Ishikawa. Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(10):1075–1088, 2001.

[92] T. Joachims. Transductive learning via spectral graph partitioning. In T. Fawcett and N. Mishra, editors, *Proc. 20th International Conference on Machine Learning (ICML)*, pages 290–297. AAAI Press, 2003.

[93] F. Juhász and K. Mályusz. Problems of cluster analysis from the viewpoint of numerical analysis. In P. Rozsa, editor, *Numerical methods, Keszthely 1977*, volume 22 of *Colloquia Mathematica Societatis Janos Bolyai*, pages 405–415. North-Holland Publishing Company, Amsterdam, 1980.

[94] R. Kannan, S. Vempala, and A. Vetta. On clusterings — good, bad and spectral. In *Proc. 41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 367–377. IEEE Computer Society Press, 2000.

[95] D. Karger, R. Motwani, and M. Sudan. Approximative graph colouring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998.

[96] S. E. Karisch and F. Rendl. Semidefinite programming and graph equipartition. In P. M. Pardalos and H. Wolkowicz, editors, *Topics in Semidefinite and Interior-Point Methods*, volume 18 of *Fields Institute Communications*, pages 77–95. American Mathematical Society, 1998.

[97] H. Karloff. How good is the Goemans–Williamson MAX CUT algorithm? *SIAM Journal on Computing*, 29(1):336–350, 2000.

[98] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data*. John Wiley & Sons, New York, 1990.

[99] J. Keuchel, S. Naumann, M. Heiler, and A. Siegmund. Automatic land cover analysis for Tenerife by supervised classification using remotely sensed data. *Remote Sensing of Environment*, 86:530–541, 2003.

[100] J. Keuchel, C. Schellewald, D. Cremers, and C. Schnörr. Convex relaxations for binary image partitioning and perceptual grouping. In B. Radig and S. Florczyk, editors, *Pattern Recognition (23rd DAGM Symposium, Munich)*, volume 2191 of *Lect. Not. Comp. Sci.*, pages 353–360. Springer, 2001.

[101] J. Keuchel and C. Schnörr. Efficient graph cuts for unsupervised image segmentation using probabilistic sampling and SVD-based approximation. In *3rd International Workshop on Statistical and Computational Theories of Vision*, Nice, France, 2003.

[102] J. Keuchel, C. Schnörr, C. Schellewald, and D. Cremers. Unsupervised image partitioning with semidefinite programming. In L. V. Gool, editor, *Pattern Recognition (24th DAGM Symposium, Zurich)*, volume 2449 of *Lect. Not. Comp. Sci.*, pages 141–149. Springer, 2002.

[103] J. Keuchel, C. Schnörr, C. Schellewald, and D. Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. Patt. Anal. Mach. Intell.*, 25(11):1364–1379, 2003.

[104] I. Y. Kim and H. S. Yang. An integration scheme for image segmentation and labeling based on Markov random field model. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(1):69–73, 1996.

[105] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM*, 49(5):616–639, 2002.

[106] D. C. Knill and W. Richards, editors. *Perception as Bayesian Inference*. Cambridge Univ. Press, 1996.

[107] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Patt. Anal. Mach. Intell.*, 26(2):147–159, 2004.

[108] M. Kočvara and M. Stingl. PENNON - a code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 18(3):317–333, 2003.

[109] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[110] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223/224(1–3):439–461, 1995.

[111] M. Laurent and S. Poljak. On the facial structure of the set of correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 17(3):530–547, 1996.

[112] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In G. N. K. Aardal and R. Weismantel, editors, *Handbook on Discrete Optimization*. Elsevier, 2004.

[113] C. Lemaréchal. Lagrangian relaxation. In M. Jünger and D. Naddef, editors, *Computational Combinatorial Optimization*, volume 2241 of *Lect. Not. Comp. Sci.*, pages 115–160. Springer, 2001.

[114] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, Tokyo, 2nd edition, 2001.

[115] L. Lovász. Semidefinite programs and combinatorial optimization. In B. A. Reed and C. L. Sales, editors, *Recent Advances in Algorithms and Combinatorics*, pages 137–194. Springer, 2003.

[116] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optimization*, 1(2):166–190, 1991.

[117] Z.-Q. Luo. Applications of convex optimization in signal processing and digital communication. *Mathematical Programming, Series B*, 97(1-2):177–207, 2003.

[118] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. J. Comp. Vision*, 43(1):7–27, 2001.

[119] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *J. Amer. Stat. Assoc.*, 82:76–89, 1987.

[120] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Patt. Anal. Mach. Intell.*, 26(5):530–549, 2004.

[121] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int. Conf.*

Computer Vision (ICCV), volume 2, pages 416–423. IEEE Comp. Soc., 2001.

[122] J. D. McCafferty. *Human and Machine Vision: Computing Perceptual Organisation*. Ellis Horwood, New York, 1990.

[123] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, New York, 1992.

[124] M. Meila. Data centering in feature space. In C. M. Bishop and B. J. Frey, editors, *Proc. 9th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Key West, Florida, 2003.

[125] M. Meila and J. Shi. A random walks view of spectral segmentation. In *Proc. 8th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Key West, Florida, 2001.

[126] H. D. Mittelmann. An independent benchmarking of SDP and SOCP solvers. *Mathematical Programming, Series B*, 95(2):407–430, 2003.

[127] B. Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, 47(3):274–291, 1989.

[128] B. Mohar. Some applications of Laplace eigenvalues of graphs. In G. Hahn and G. Sabidussi, editors, *Graph Symmetry: Algebraic Methods and Applications*, volume 497 of *NATO ASI Series C*, pages 227–275. Kluwer, Dordrecht, 1997.

[129] B. Mohar and S. Poljak. Eigenvalues in combinatorial optimization. In R. A. Brualdi, S. Friedland, and V. Klee, editors, *Combinatorial and Graph-Theoretical Problems in Linear Algebra*, volume 50 of *IMA Vol. Math. Appl.*, pages 107–151. Springer, 1993.

[130] R. D. C. Monteiro. Primal-dual path following algorithms for semidefinite programming. *SIAM Journal on Optimization*, 7(3):663–678, 1997.

[131] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9(1-3):141–160, 1998.

[132] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming*. SIAM, 1994.

[133] Y. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, 8(2):324–364, 1998.

[134] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 849–856. MIT Press, 2002.

[135] S. Oliveira, D. E. Stewart, and T. Soma. Semidefinite programming for graph partitioning with preferences in data distribution. In J. M. L. M. Palma, J. Dongarra, V. Hernández, and A. A. de Sousa, editors, *5th International Conference on High Performance Computing for Computational Science (VECPAR 2002)*, volume 2565 of *Lect. Not. Comp. Sci.*, pages 703–716. Springer, 2003.

[136] M. L. Overton and R. S. Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM J. Matrix Anal. Appl.*, 13(1):41–45, 1992.

[137] P. M. Pardalos and H. Wolkowicz, editors. *Quadratic assignment and related problems*, volume 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1994.

[138] P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(8):823–839, 1989.

[139] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Math. Oper. Res.*, 23(2):339–358, 1998.

[140] M. Pavan and M. Pelillo. Dominant sets and hierarchical clustering. In *Proc. 9th Int. Conf. Computer Vision (ICCV)*, pages 362–369. IEEE Comp. Soc., 2003.

[141] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 145–152, 2003.

[142] P. Perona and W. Freeman. A factorization approach to grouping. In H. Burkhardt and B. Neumann, editors, *5th European Conference on Computer Vision (ECCV)*, volume 1406 of *Lect. Not. Comp. Sci.*, pages 655–670. Springer, 1998.

[143] C. Peterson and B. Söderberg. Artificial neural networks. In E. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*, chapter 7. Wiley & Sons, Chichester, 1997.

[144] S. Poljak and F. Rendl. Nonpolyhedral relaxations of graph bisection problems. *SIAM Journal on Optimization*, 5:467–487, 1995.

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

[146] A. Pothen, H. D. Simon, and K.-P. Liou. Partioning sparse matrices with eigenvectors of graphs. *SIAM Journal of Matrix Analysis and Applications*, 11(3):430–452, 1990.

[147] J. Princen, J. Illingworth, and J. Kittler. A formal definition of the Hough transform: Properties and relationships. *Journal of Mathematical Imaging and Vision*, 1:153–168, 1992.

[148] J. Puzicha and J. M. Buhmann. Multiscale annealing for unsupervised image segmentation. *Comp. Vision and Image Underst.*, 76(3):213–230, 1999.

[149] J. Puzicha, J. M. Buhmann, Y. Rubner, and C. Tomasi. Empirical evaluation of dissimilarity measures for color and texture. In *Proc. 7th Int. Conf. Computer Vision (ICCV)*, pages 1165–1172. IEEE Comp. Soc., 1999.

[150] M. Ramana and A. J. Goldman. Some geometric results in semidefinite programming. *Journal of Global Optimization*, 7(1):33–50, 1995.

[151] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. 9th Int. Conf. Computer Vision (ICCV)*, pages 10–17. IEEE Comp. Soc., 2003.

[152] F. Rendl. Semidefinite programming and combinatorial optimization. *Applied Numerical Mathematics*, 29(3):255–281, 1999.

[153] F. Rendl and H. Wolkowicz. A projection technique for partitioning the nodes of a graph. *Annals of Operations Research*, 58:155–180, 1995.

[154] K. Rose, E. Gurewitz, and G. C. Fox. Constrained clustering as an optimization method. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(8):785–794, 1993.

[155] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann. Optimal cluster preserving embedding of non-metric proximity data. *IEEE Trans. Patt. Anal. Mach. Intell.*, 25(12):1540–1551, 2003.

[156] S. Santini and R. Jain. Similarity measures. *IEEE Trans. Patt. Anal. Mach. Intell.*, 21(9):871–883, 1999.

[157] S. Sarkar and K. L. Boyer. Perceptual organization in computer vision: A review and a proposal for a classificatory structure. *IEEE Trans. Systems, Man, and Cybernetics*, 23(2):382–399, 1993.

[158] S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding*, 71(1):110–136, 1998.

[159] S. Sarkar and P. Soundararajan. Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22(5):504–525, 2000.

[160] M. Sato and S. Ishii. Bifurcations in mean-field-theory annealing. *Physical Review E*, 53(5):5153–5168, 1996.

[161] C. Schellewald, J. Keuchel, and C. Schnörr. Image labeling and grouping by minimizing linear functionals over cones. In M. Figueiredo, J. Zerubia, and A. K. Jain, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition (3rd International Workshop EMMCVPR, Sophia Antipolis)*, volume 2134 of *Lect. Not. Comp. Sci.*, pages 267–282. Springer, 2001.

[162] A. J. Seary and W. D. Richards. Spectral methods for analyzing and visualizing networks: An introduction. In R. Breiger, K. Carley, and P. Pattison, editors, *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, pages 209–228. National Academies Press, 2003.

[163] A. Sha'ashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *Proc. 2nd Int. Conf. Computer Vision (ICCV)*, pages 321–327. IEEE Comp. Soc., 1988.

[164] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 70–77, 2000.

[165] N. Shental, A. Zomet, T. Hertz, and Y. Weiss. Learning and inferring image segmentations using the GBP typical cut algorithm. In *Proc. 9th Int. Conf. Computer Vision (ICCV)*, pages 1243–1250. IEEE Comp. Soc., 2003.

[166] R. N. Shepard. Toward a universal law of generalization for psychological science. *Science*, 237:1317–1323, 1987.

[167] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proc. 6th Int. Conf. Computer Vision (ICCV)*, pages 1154–1160, 1998.

[168] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22(8):888–905, 2000.

[169] H. D. Simon and S.-H. Teng. How good is recursive bisection? *SIAM Journal on Scientific Computing*, 18(5):1436–1445, 1997.

[170] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 82(1):93–133, 1989.

[171] A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In P. Langley, editor, *Proc. 17th International Conference on Machine Learning (ICML)*, pages 911–918. Morgan Kaufmann, 2000.

[172] A. H. S. Solberg, T. Taxt, and A. K. Jain. A Markov random field model for classification of multisource satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 34(1):100–113, 1996.

[173] P. Soundararajan and S. Sarkar. An in-depth study of graph partitioning measures for perceptual organization. *IEEE Trans. Patt. Anal. Mach. Intell.*, 25(6):642–660, 2003.

[174] D. A. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. In *Proc. 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 96–105. IEEE Comp. Soc., 1996.

[175] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[176] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2003.

[177] J. F. Sturm and S. Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.

[178] P. H. Tan and L. K. Rasmussen. The application of semidefinite programming for detection in CDMA. *IEEE Journal on Selected Areas in Communications*, 19(8):1442–1449, 2001.

[179] M. J. Todd. Semidefinite optimization. In *Acta Numerica*, volume 10, pages 515–560. Cambridge University Press, 2001.

[180] Z. Tu and S.-C. Zhu. Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(5):657–673, 2002.

[181] B. van Cutsem, editor. *Classification and Dissimilarity Analysis*, volume 93 of *Lecture Notes in Statistics*. Springer, 1994.

[182] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[183] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[184] V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer, 1982.

[185] O. Veksler. Image segmentation by nested cuts. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 339–344, 2000.

[186] L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comp. Vision*, 50(3):271–293, 2002.

[187] Vision texture database. Media Laboratory, MIT, Cambridge, http://www-white.media.mit.edu/vismod/imagery/VisionTexture.

[188] J.-P. Wang. Stochastic relaxation on partitions with connected components and its application to image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 20(6):619–636, 1998.

[189] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *IEEE Trans. Patt. Anal. Mach. Intell.*, 25(6):675–690, 2003.

[190] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *Proc. 7th Int. Conf. Computer Vision (ICCV)*, pages 975–982. IEEE Comp. Soc., 1999.

[191] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13 (NIPS)*, pages 682–688. MIT Press, 2001.

[192] L. R. Williams and K. K. Thornber. A comparison of measures for detecting natural shapes in cluttered backgrounds. *Int. J. Comp. Vision*, 34(2/3):81–96, 2000.

[193] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, volume 27 of *Appl. of Mathematics*. Springer, Heidelberg, 1995.

[194] H. Wolkowicz. Solving semidefinite programs using preconditioned conjugate gradients. Technical Report CORR 2001–49, University of Waterloo, Ontario, Canada, 2001. To appear in OMS.

[195] H. Wolkowicz and M. F. Anjos. Semidefinite programming for discrete optimization and matrix completion problems. *Discrete Applied Mathematics*, 123(1-3):513–577, 2002.

[196] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*, volume 27 of *International series in operations research & management science*. Kluwer Acad. Publ., Boston, 2000.

[197] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Applied Mathematics*, 96-97(1):461–479, 1999.

[198] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(11):1101–1113, 1993.

[199] Y. Ye. *Interior Point Algorithms: Theory and Analysis*. Wiley, 1997.

[200] Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming*, 84:219–226, 1999.

[201] Y. Ye. A .699-approximation algorithm for max-bisection. *Mathematical Programming, Series A*, 90(1):101–111, 2001.

[202] Y. Ye and S. Zhang. New results on quadratic minimization. *SIAM Journal on Optimization*, 14(1):245–267, 2003.

[203] S. Yu, R. Gross, and J. Shi. Concurrent object recognition and segmentation by graph partitioning. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15 (NIPS)*, pages 1407–1414. MIT Press, 2003.

[204] S. X. Yu and J. Shi. Multiclass spectral clustering. In *Proc. 9th Int. Conf. Computer Vision (ICCV)*, pages 313–319. IEEE Comp. Soc., 2003.

[205] F. Zhang. *Matrix Theory: Basic Results and Techniques*. Springer, 1999.

[206] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Combinat. Optimization*, 2(1):71–109, 1998.

[207] S.-C. Zhu and D. Mumford. Prior learning and Gibbs reaction-diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(11):1236–1250, 1997.

[208] S.-C. Zhu, Y. Wu, and D. Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *Int. J. Comp. Vision*, 27(2):1–20, 1998.

[209] U. Zwick. Outward rotations: A tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proc. 31st ACM Symposium on Theory of Computing (STOC)*, pages 679–687, 1999.