Spectral Rounding & Image Segmentation

David A. Tolliver

CMU-RI-TR-06-44

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Robotics.

Committee:
Gary L. Miller (co-chair)
Robert T. Collins (co-chair)
Simon Baker
Tai Sing Lee
Ramin Zabih (Cornell)

August 2006

Abstract

*The task of assigning labels to pixels is central to computer vision. In automatic segmentation an algorithm assigns a label to each pixel where labels connote a shared property across pixels (*e.g. *color, bounding contour, texture). Recent approaches to image segmentation have formulated this labeling task as partitioning a graph derived from the image. We use spectral segmentation to denote the family of algorithms that seek a partitioning by processing the eigenstructure associated with image graphs.*

*In this thesis we analyze current spectral segmentation algorithms and explain their performance, both practically and theoretically, on the Normalized Cuts (NCut) criterion. Further, we introduce a novel family of spectral graph partitioning methods, spectral rounding, and apply them to image segmentation tasks. Edge separators of a graph are produced by iteratively reweighting the edges until the graph disconnects into the prescribed number of components. At each iteration a small number of eigenvectors with small eigenvalue are computed and used to determine the reweighting. In this way spectral rounding directly produces discrete solutions where as current spectral algorithms must map the continuous eigenvectors to discrete solutions by employing a heuristic geometric separator (*e.g. *k-means).*

We show that spectral rounding compares favorably to current spectral approximations on the NCut criterion in natural image segmentation. Quantitative evaluations are performed on multiple image databases including the Berkeley Segmentation Database. These experiments demonstrate that segmentations with improved NCut value (obtained using the SR-Algorithm) are more highly correlated with human hand-segmentations.

Acknowledgments

This work would not have been possible without the patient guidance of many people during my graduate career. I would like to thank Robert Collins for introducing me to Computer Vision and planting the seed of graduate school in my mind. I owe Jianbo Shi a huge debt of gratitude for his patience and infectious enthusiasm for research. His interest in spectral graph theory and image segmentation have obviously influenced my work. Over the past two years I have had the great pleasure of working with Gary Miller on the bulk of the work contained in this document. Access to Gary's time and energy has been a rare privilege for which I am deeply grateful.

As the complexities of graduate school often occur outside the scope of research and coursework I owe Suzanne Lyons Muth a great deal. I doubt I would have successfully completed the Robotics program without her help. Her willingness to accommodate my absent mindedness and innumerable character deficits in areas of personal organization enabled the smooth navigation of the pitfall ridden process that is graduate school.

Over my years in graduate school I have drawn inspiration and motivation from my peers in the program. In particular, I owe Matt Deans, Kiran Bhat, and Stella Yu a great thanks for numerous conversations in which they deftly uncovered and eliminated a little ignorance of mine during my early years as a student. Drew Bagnell and Aaron Courville kept daily life interesting with conversations on a wide range of topics – from the economics of environmentalism to monte carlo methods for parameter estimation. I owe Sonya Allin an immeasurable debt of gratitude for sharing her life with me, as well as her relentlessly positive take on my progress and prospects.

Finally, I owe my family a huge debt of gratitude for their support over the past six years. The support and encouragement of my parents and brothers for my seemingly quixotic desire to complete a Ph.D. buoyed my spirits during many a low point in the process.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The task of assigning labels to pixels is central to computer vision. Image segmentation requires that the algorithm assign a label to each pixel in the image. This broad definition includes operations such as stereo depth assignment, object recognition, and image segmentation. At the low-level, shared segment labels connote a shared statistical quantity between pixels in the image. At higher levels segmentation problems undertake the task of assigning labels with semantic value. For example labeling the image pixels as members of either of the abstract classes *figure* or *ground*, or supplying concrete labels such as *face*, and *car*.

In recent years segmentation has become a central component in mid and high level vision algorithms. The image parsing system of [TCYZ05] used segmentation to validate detection events in the image. In their work on geometric pop-up [HEH05] and contextual classification [HEH06] initial segmentations play a pivotal role in limiting the support of classification events (both high level, mid-level - *i.e*. surface orientation). Object detection systems, a recent staple of the vision community, are being augmented to include segmentation specific support for feature detection events.

1.1 Problems and Solutions

In this work segmentation is formulated as partitioning an graph defined over the pixels. In the simplest instance a weighted graph $G = (V, E, w)$ is built from the image in the following fashion. The vertex set of the graph is taken to be the pixels of the image. The edge-set E is a fixed for the image and connects neighboring pixels (on a mesh-like lattice). The weights of these edges w are determined as a function of the pixel similarity - *e.g.* intensity or color correlation. Finding a good segmentation of the image can now be thought of as finding an inexpensive cut (or cut-set) in G.

In partitioning a graph we must specify a measure of cut cost or quality. Most measures of partition quality yield NP-hard optimization problems. This raises two important questions. First, does a particular optimization problem capture good partitions for the image segmentation domain, especially in light of the optimization being NP-hard and thus we may never know the true optimum anyway. Second, given that the optimal value is a good characterization are approximations quickly constructible and do they return good partitions?

One popular formulation, used in image processing and clustering, is the normalized cut (NCut) of a graph introduced by Shi and Malik [SM00]. The ideas contained therein were further explored by Ng *et al*. [NJW02] and Yu and Shi [YS03a] both of whom motivated multi-way partitioning algorithms. In part, our method was motivated by observations made in [NJW02, YS03a]. Now, how does the NCut optimization problem fare against our two questions?

It is not difficult to construct image examples for which common image percept does not correspond to the optimal NCut of the image (*e.g.* see Shental *et al.*'s example [SZHW03] and see Figure 5.6 for an analogous case in natural images). This is unsurprising, and an acknowledged attribute of all objective measures of cluster or partition quality (see Kleinberg [Kle03] and Meilă [Mei05] for treatment of this issue). But, for many images, as we shall show, there are segmentations with a smaller normalized cut value than in those generated by earlier methods that are at the same time more pleasing. For example, one of the main empirical advantages of spectral rounding technique seems to be that it is less likely to split the image in homogeneous regions, see Figure 5.2, while returning smaller NCut values. Thus good image segmentations are generated as graph partitions without reformulating the underlying combinatorial problem.

The two common paradigms for approximating such objective functions are 1) linear or semidefinite programming [LR88, XJ03, ARV04]. and 2) spectral methods [Chu92, SM00]. In this paper we introduce a spectral technique that empirically improves upon existing spectral algorithms for quotient cuts. Earlier spectral methods consisted of a two stage algorithm. In the first stage a small collection of, say k, eigenvectors with small eigenvalues are computed. While in the second stage these vectors are used to map the graph vertices into $\mathbb{R}^k$ and a geometric separator is then applied [CGT94].

In chapter 3 we introduce **Spectral Rounding** as an alternative to rounding with geometric separator. Spectral rounding iteratively reweights the graph in such a fashion that it eventually disconnects. At each iteration we will use the eigenvalues and eigenvectors of the reweighted graph to determine new edge weights. At first hand this may seem very inefficient, since the most expensive step in the two stage method is the eigenvector calculation. By using the eigenvector from the prior step as a starting point, for finding the new eigenvector, simple powering methods seem to work in only a small number of steps.

1.2 Related Work

In this section we briefly touch on related and prior work in areas of image segmentation. We make the distinction between low and high level formulations of the segmentation problem. Additionally, we briefly touch on the distinguishing differences between modern approaches, focusing on what types of constraints are easily encoded in a particular method, and the computational overhead of the various techniques.

1.2.1 Low-Level Segmentation

The low-level segmentation problem has been posed as a feature space clustering problem [DS80], a statistical estimation problem [CM02], a diffusion process [ZLY95], a level set optimization [Set96], a graph partition [Vek00, SM00], and as inference on probabilistic graphical models of the image generation process [GG84, Bes86, Li01]. Most modern segmentation systems employ a mode-seeking statistical estimation approach such as the *mean-shift*, or a probabilistic technique derived from a graphical model of the image data.

Before continuing on it is worth commenting on why one would choose to use a spectral segmentation method such as [SM00, YS03a, TM06]. In particular, what desirable features do spectral relaxations possess as approximations for image segmentation applications. Given that the pixels are connected in spatially localized patterns, a spectral method will generate image segments that are connected in the image plane (as well at the graph). This property of the segments is not guaranteed by segmentations algorithms using [Bis95, CM99, Vek00, GG84, Li01]. Spectral methods provide a means of bounding several combinatorial (discrete) functions of cut quality (known collectively as quotient cuts). Such cut functions are covered in chapter 2 and form the basis of assessment for segmentation quality in graph partitioning formulations.

We use the term low-level segmentation to denote the collection of segmentation techniques that are free of semantic information. These methods seek to partition the image into contiguous regions that share a level of coherence. This coherence can be as simple as color similarity, texture similarity, or boundary smoothness. The majority of such approaches are motivated by the work of the Gestalt psychologists on the heuristics that guide visual binding. This process in biological systems is analogous to image segmentation in computer vision systems – features are grouped together to form coherent regions in the visual field.

Clustering Approaches

We group clustering and *mean-shift* algorithms [CM99] into statistical approaches for image segmentation. The $k-$means algorithm [Bis95] seeks k points in feature space that explain the data as the means of disjoint populations. These points are then used to label pixels in the image domain by their mean in feature space. The *mean-shift* algorithm is a mode seeking algorithm. For a given distribution in feature space, the maxima of a density (under a kernel image) are located through repeated restarting and hill climbing. These modes are then used to assign labels to pixels in the image domain. In terms of our requirements, both of these methods share a major short-coming: as the feature spaces may not preserve the geometry of the image plane, there is no constraint enforcing contiguous segments in image plane.

Contours and Level-Sets

Contour based approaches seek a closed curve in the image plane that satisfies assumptions about the local differential structure. These methods typically seek to maximize the coverage of detected edge pixels, while satisfying smoothness constraints on the contour. The *snake* contour approach for low level segmentation was proposed by Kass and Witkin in [KWT87]. This family of approaches optimizes over a parametric curve with penalty terms on smoothness. Unlike graph theoretic approaches to low level segmentation, active contours typically require initialization. In most cases placing a contour interior to the object of interest is sufficient. In many applications these models are inappropriate because the smoothness terms may dominate image structure.

Level set optimization can be applied to solve Mumford-Shah functionals [BKS02] and contour parameters. The solutions to Mumford-Shah optimization, a set of piece-wise linear functionals covering the image, results in an image segmentation. Alternately, the variational level-set optimization framework can also be applied directly to contour estimation, to extend contour estimation to an edgeless case [CV01] or to combine region and edge information[PD99].

There are two major drawbacks to these approaches. The first is nature of the objective functions. In general the error surface contains many local maxima, and in the case of Mumford-Shah functional optimization is computationally expensive. Typically contour based methods suffer from a trade off between the smoothness of the contour and the stability of the solution. If the penalty term on the smoothness is too low the contour explains a great deal incorrectly detected edges. If the penalty term is too high, the result is a rigid, overly simple, estimate of the target segmentation.

Probabilistic Approaches

The probabilistic family of approaches to low-level segmentation originate from the seminal work of Geman and Geman [GG84]. These methods determine low-level pixel label configurations that are probable under a Markov Random Field (MRF) such that $P(S|I) \propto \prod_{x \in S} P(i(x)|s(x))P(S)$, where $i(x)$ is the image observation at the pixel x, $s(x)$ is the segment label assigned to x, and S is the configuration of pixel labels. There are two major components worth discussing in the MRF model. First the independence assumption that $P(I|S) = \prod_{x \in S} P(i(x)|s(x))$, which asserts that the observations are conditionally independent given the labels. Second, the prior $P(S) = \sum_{x \in S} \sum_{v \in \mathcal{N}(x)} \rho(s(x), s(v))$, called the interaction field, biases configurations of S toward consistent labels on adjacent pixels under the neighborhoods $\mathcal{N}(x)$. Priors of the form above can be thought of as graphs with topologies defined by label interactions.

For two class segmentation problems, with an exponential family pair-wise prior, the maximum likelihood configuration of S can be computed in polynomial time as a reduction to max-flow [FF62]. There are a few technical conditions that must be met by the prior $P(S)$

related to graph topology and the penalty function ρ. The class of priors was broadened to include convex functions by Ishikawa in [Ish03].

For $k-$class segmentations, where $k > 2$, probable configurations of the MRF can be found by multi-commodity flow optimizations. This observation inspired the segmentation algorithm of Veksler [Vek00], using the multi-commodity flow algorithm proposed by Boykov *et al.* in [BVZ98]. The class of admissible reductions to multi-flow optimizations where further charted by Kolmogrov and Zabih in [KZ04]. More complex models beyond the scope of a flow reduction were proposed by Tu *et al.* in [KWT01]. Optimization of label configurations for these models employ a data-driven Markov Chain Monte Carlo to find a local maxima.

Graph Partitioning

The model essential to the MRF formulation in §1.2.1 can be thought of a graph whose topology is defined by the neighborhood functions $\mathcal{N}(x)$. By dropping the probabilistic interpretation and thinking only about optimizing a function of the label collisions, this problem can be thought of as a graph partition. These objective functions include the normalized cut [SM00], the $p-$way max flow approaches [BVZ98], as well as a host of other cut functions and optimization techniques. These approaches are addressed in greater detail in chapter 3.

1.2.2 Mid-Level Segmentation

We use mid-level segmentation to denote the set of image segmentation problems that include some top-level input, but no object information. Examples include the figure-ground method proposed by Yu in [YS02], and the spatially coherent clustering method developed by Zabih in [ZK04].

In the work of Yu [YS02] a small set of pixels are constrained to the same class. This effectively contracts the graph on the labeled vertices. This approach was applied to figure-ground segmentation by constraining the border pixels to a single class. The mid-level input information is taken to be the set of pixel equivalence constraints.

In [ZK04] an iterative scheme is proposed in which a small set of parameters are estimated for each segment as the flow optimization is performed. These parameters are then used to re-weight the graph during the optimization. The result is a segmentation routine that is robust to unstructured noise *(e.g. salt-and-pepper)* in the image plane. Here the mid-level input is the model class used in parameter estimation.

High-level Segmentation

The missing ingredient in most image segmentation approaches is the Gestaltist rule of *model consistency*. Rightly, this heuristic suggests that the constituent regions of a known object should be grouped in the visual field. Notable exceptions include the covering work of Borenstein and Ullman [BU02], and that of Yu and Shi [YS03b].

The method proposed by Borenstein [BU02] requires a large collection of patches taken from the parts of the target class. This loose collection of patches, coupled with overlap compatibility scores, is used to cover the pixels of interest in the image. Thus a segmentation is derived as the union of on-object pixels from the covering patch collection. The optimization procedure described in [BU02] contains no guarantees regarding the quality or uniqueness of the solution. As the template based approach requires hand segmentation for the class patches, the method carries a heavy burden in hand labeling as well. This method has an implicit representation of shape in the patch compatibility functions supplied by the exemplars.

The method proposed in [YS03b] attempts to segment out detected objects in the scene by combining a graph on pixel similarities with scores for object feature configurations. The configuration of object features are weighted if they are probable under the training set. These object features are then linked to low-level pixel interactions through edge weights. However, the method does not have an explicit representation for the global shape of the object.

Perhaps the easiest to tailor to high-level shape information are the contour based methods. Active contours for tracking and segmentation use a distribution over contour parameters estimated from a training corpus to constrain the set of likely shapes. These models, while powerful, often return overly smooth results in the image. Additionally, a similarity transform is generally factored out of the shape statistics. By optimizing over the shape and a similarity transform, many local maxima exist. In earlier work [TMC05] we combined simple statistical contour models with graph partitioning method to obtain shape guided image segmentations.

1.3 Contributions

The primary contribution of this document is introduced in Chapter 3 on **Spectral Rounding**. However, in getting to this point a number of useful results were obtained that extend beyond the scope of a particular algorithm.

1. A novel rounding algorithm that projects the eigenvectors of a graph Laplacian onto the feasible set of partition matrices (Chapter 4).

2. An analysis of the “spectral segmentation” problem, including a graph augmentation scheme that ameliorates some of the short-comings of spectral methods when coupled with planar local neighborhood graphs (Chapter 3).

3. Data and analysis suggesting that it is not spectral segmentation (properties simply of the eigenvectors and eigenvalues), but rather the combinatorial objective function **Normalized Cut** which should be optimized when segmenting images (Chapter 5). This is shown, both in the prior over cuts induced by the **NCut** (figure 2.3) and the positive correlation between improved perceptual relevance and lower **NCut** cost. The former property refutes the “balance bias” that is often ascribed the normalized cut function and

demonstrates its a property of the spectral relaxation. The latter is validated with medical images and human segmentation of natural imagery.

4. The Cheeger inequality, relating the sparsest cut of a graph to the eigenvalues of the normalized Laplacian, is adapted to the normalized 2-cut. This bound is then generalized to the k-way normalized cut and an algorithm is furnished to achieve it (Chapter 2).

1.4 A Roadmap

The document is loosely organized in the following manner. Chapter 2 collects the mathematical preliminaries required for the analysis of spectral graph algorithms and the derivation of the spectral rounding algorithm. Chapter 3 provides a detailed analysis of the relationship between the underlying graph topologies used in spectral image segmentation and the eigenvectors of the graph. An algorithm is introduced to called "expansion augmentation" that improves spectral segmentation with computational overhead. Chapter 4 introduces the spectral rounding algorithm which efficiently improves spectral segmentation. Chapter 5 presents results on random geometric graphs, natural images, medical images and the Berkeley Hand Segmentation Database.

Chapter 2 is meant primarily as a reference for use by the reader when faced with an unfamiliar concept. The chapter contains a collection of useful definitions and notational conventions §2.1. The spectrum of the graph Laplacian is addressed in §2.2. In §2.2.1 the differentials of the eigenvalues and eigenvectors of these matrices is covered, with a basic proof existence for the generalized eigenvalues of a matrix pencil. These tools are needed later in the study of the spectral rounding algorithm. In §2.3 the Cheeger inequality is proved for the Normalized 2-cut of the graph. This fundamental inequality bounds the cost of the normalized cut by the second smallest eigenvalue of the normalized Laplacian. The chapter ends with the statement of a recent generalization of Fiedler's theorem relating the topology of the graph and the structure of the Laplacian eigenvectors.

Chapter 3 discusses spectral image segmentation in detail. The standard spectral partitioning algorithm is introduced in §3.1. The interaction between mesh-like (spatially localized) graphs and the standard spectral algorithm is covered in §3.2. A simple method for improving the performance of standard spectral algorithms is proposed §3.2.2. This method adorns the mesh-like graph with a small set of random edges (weighted by data in the underlying mesh). This augmented graph is less likely to produce plane waves as its topological structure is no longer reducible to that of a string (or mattress of springs).

Chapter 4 introduces the **Spectral Rounding** algorithm (in §4.1). In §4.1.1 a collection of reweighting schemes are proposed. These schemes are applied iteratively to reweight the graph until is eventually disconnects into a prescribed number of pieces. In §4.2, the notion of a fractional average is introduced and connected to both the reweightings that decrease a target Rayleigh quotient and the eigenvalues of the normalize Laplacian. In §4.3 convergence is proved for three reweighting schemes. In §4.4.1 a heuristic for selecting eigenvectors and choosing k (the number of partitions) based on the derivatives of the eigenvalues are put forward.

Chapter 5 provides results and analysis of the spectral rounding algorithm on random geometric graphs, natural images and medical images. The analysis includes the expected improvement in normalized cut cost between the standard spectral algorithm and the cuts produced by spectral rounding. The partitions generated by the two methods are compared using an information theoretic metric that reports the number of bits required to encode the residual entropy of the two partitions once their mutual information is removed. Finally, image segmentations are validated by human labels when the information is available. This analysis is done on a simple medical image processing task and the Berkeley segmentation database. The results provide evidence that the combinatorial (graph theoretic) interpretation of NCuts should be preferred to the strictly spectral explanation of NCuts.

Chapter 6 addresses promising directions for future work. The three main issues surround improved theoretical results for spectral or semi-spectral partitioning methods, the efficient integration of high-level information into the segmentation task and the computational cost of the initial eigencomputation. Preliminary approaches are put forth.

The remainder of this chapter sketches out problem definitions and concerns prior work in the area of segmentation – with special emphasis on graph theoretic approaches in general and spectral methods in particular. The putative contributions of the thesis can be found at the end of this chapter.

Chapter 2

Linear Algebra of Graphs

The connection between graphs and matrices provides powerful tools for tackling both graph theoretic and linear algebra problems. In this chapter we provide a collection of useful results on the relationship between Laplacians and their underlying graphs. The chapter begins with the introduction of notation and definitions needed in the remainder of the document. The combinatorial Laplacian is then introduced, followed by two major theorems - the Cheeger inequality which bounds the Normalized Cut (*see theorem 2*) and the Nodal Domain theorem of Fiedler (*see theorem 3*).

Both major theorems in this chapter have implications for spectral image segmentation and clustering. In the first we furnish an approximation algorithm for the NCut using the eigenstructure of the normalized Laplacian. The proof is an adaptation of the bound on the "sparsest cut" to the normalized 2-way cut. The bound is then generalized to the k-way normalized cut.

The second generalizes Fiedler's famous theorem relating structure in the eigenvectors to the adjacency pattern of the graph. This in turn bounds the number of connected components eigenvectors can partition the graph into by their **spectral index**.

2.1 Notation and Definitions

We denote a graph $G = (V, E, w)$ with vertex set V, edge set E and an edge weighting w. By convention $|V| = n$ and $|E| = m$ and $w : E \rightarrow [0, 1]$. The graph G is assumed to be undirected graph without multiple edges or self-loops.

Definition 1. *The* **weighted adjacency** *matrix of $G = (V, E, w)$, $A(G)$ or simply A, is defined as*

$$A(i,j) = \begin{cases} a_{ij} = w_{ij} & \textit{if } (i,j) \in E \\ 0 & \textit{otherwise} \end{cases} \tag{2.1.1}$$

Definition 2. *The* **weighted degree** *of vertex i is $d_i = vol(v_i) = \sum_{j=1}^{n} w_{ij}$. We assume that no vertex has zero degree. The weighted degree matrix of G, $D(G)$ or simply D is defined as*

$$D(i,j) = \begin{cases} d_i & \textit{if } i = j \\ 0 & \textit{otherwise} \end{cases} . \tag{2.1.2}$$

Definition 3. *The* **volume** *of a vertex set $V' \mid V' \subseteq V(G)$, denoted $vol(V')$, is computed as*

$$vol(V') = \sum_{i \in V'} \sum_{(i,j) \in E(G)} a_{ij} = \sum_{i \in V'} d_i,$$

where d_i is the weighted degree of the vertex i^{th} vertex in V'.

Definition 4. *The* **cut** *associated with a vertex set V', i.e. those edges connecting V' to the remainder of the vertices $V \setminus V'$, denoted $\delta V'$ is defined as*

$$\delta V' = \sum_{u \in V', v \in V \setminus V' (i,j) \in E} a_{ij}.$$

Definition 5. *The* **normalized Laplacian** *of a graph $G = (V, E, w)$ denoted $\mathcal{L}$, is defined as $\mathcal{L} = D^{-1/2} L D^{-1/2} = D^{-1/2}(D - A)D^{-1/2} = I - D^{-1/2} A D^{-1/2}$, and is co-spectral with minimax points of the generalized Rayleigh quotient $\frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T D \mathbf{x}}$.*

Definition 6. *The* **normalized cut** *of a graph $G = (V, E, w)$ is defined as*

$$nc_k(G) = \underset{\{V_i\}}{min} \; : \; \frac{1}{k} \sum_{i=1}^{k} \frac{\delta(V_i, V \setminus V_i)}{vol(V_i)}$$

where $V_i \cap V_j = \emptyset$ *and* $V(G) = \bigcup_{i=1}^{k} V_i$.

Definition 7. *The* **conductance** *of a graph* $G = (V, E, w)$ *is defined as*

$$\Phi(G) = \underset{V'}{min} \ : \ \frac{\delta(V', V \setminus V')}{vol(V')}$$

where $vol(V') \leq \frac{1}{2} vol(G)$.

Definition 8. *The* **generalized Laplacian** *of a weighted graph* $G = (V, E, w)$ *denoted* $L(G)$ *or simply* L *is defined as* $L = D - A$.

The **generalized Laplacian** will be referred to as the Laplacian of the graph from this point forward. We state following useful identity, given a map $f \ : \ V \rightarrow \mathbb{R}$ the symmetric quadratic form on L yields :

$$\begin{aligned} f^T L f &= f^T D f - f^T A f \\ &= \sum_{(ij) \in E(G)} a_{ij}(f_i^2 + f_j^2) - a_{ij}(2 f_i f_j) \\ &= \sum_{(ij) \in E(G)} a_{ij}(f_i - f_j)^2 \end{aligned}$$

and, notably, can be used to compute $\delta(V', V \setminus V')$ on discrete valuations f.

Definition 9. *The matrix* $\mathbf{rank}(M)$ *is equal to the dimension of the column-space of* M*, and the* $\mathbf{co} - \mathbf{rank}(M)$ *is equal to the dimension of the nullspace of* M.

We make a simple observation about these generalized Rayleigh quotients:

Lemma 1. *Given a weighted symmetric graph* $G = (V, E, w)$ *then the normalized Rayleigh quotient can be written as*

$$\frac{f^T L f}{f^T D f} = \frac{\sum_{(i,j) \in E, i<j} (f_i - f_j)^2 w_{ij}}{\sum_{(i,j) \in E, i<j} ((f_i)^2 + (f_j)^2) w_{ij}} \tag{2.1.3}$$

where $f_i = f(v_i)$

Proof. The fact the $f^T L f = \sum_{(i,j) \in E, i<j} (f_i - f_j)^2 w_{ij}$ and $f^T D f = \sum_{(i,j) \in E} ((f_i)^2 + (f_j)^2) w_{ij}$ are obtained by standard calculations. □

Definition	Equation	$\lambda-$map	$f-$map	
$L = D - A$	$Lf = \lambda Df$	$\cdot$	$\cdot$	[Chu92] [SM00]
$\mathcal{L} = D^{-1/2}LD^{-1/2}$	$\mathcal{L}g = \lambda g$	$\cdot$	$g = D^{1/2}f$	[Chu92] [NJW02]
$T = D^{-1}A$	$Tf = \gamma f$	$\gamma = 1 - \lambda$	$\cdot$	[JS89] [Chu92] [MS01]
$\mathcal{A} = D^{-1/2}AD^{-1/2}$	$\mathcal{A}g = \gamma g$	$\gamma = 1 - \lambda$	$g = D^{1/2}f$	[MS01] [NJW02]
$\cdot$	$Lf = \eta Af$	$\eta = \frac{\lambda}{\lambda-1}$	$\cdot$	[SBB00]
$\cdot$	$A^{-1/2}DA^{-1/2}q = \sigma q$	$\sigma = \eta + 1 = \frac{1-\lambda}{\lambda-2}$	$q = A^{1/2}f$	

Table 2.1: A table of linearly equivalent eigenspaces and associated eigenvalue transformations. Where A is taken to be the weighted adjacency, D the weighted degree of G. Each generalized Rayleigh quotient is followed by its symmeterized form, and linear map from the eigenspace of L, D.

The main importance of Lemma 1 is that for each valuation f and each edge e_{ij} we get the fraction $\frac{(f_i - f_j)^2}{(f_i)^2 + (f_j)^2}$. These fractions will figure prominently in Chapter 4 on spectral rounding.

2.2 Spectrum of a graph

The eigenvalues of the matrix-pencil $L(G), D(G)$ and matrix $\mathcal{L}(G)$ provide a great deal of information about the graph G. Consider a positively weighted undirected graph $G = (V, E, w)$, the resulting matrices $L(G)$ and $\mathcal{L}(G)$ are symmetric. If we further assume that the graph has no self-loops then L and $\mathcal{L}$ are symmetric positive semi-definite (SPSD) matrices. Recall that the spectrum of a SPSD matrix M is the complete set of eigenvalues $0 = \lambda_1(M) \leq \ldots \leq \lambda_n(M)$ and that all the eigenvalues are real. The index i in $\lambda_i(M)$ is the **spectral index** of λ_i.

The $\mathbf{co-rank}$ of L (see def. 9) is the number of connected components contained in the graph G under the weighting w. Given k connected components in G there must exist k orthogonal vectors such that $f_i^T f_j = 0$ for $i \neq j$ and $f_i^T L f_i = 0$. As k exclusive subsets V exist that share no edges then k orthogonal vectors exist such that $f_i(v) = \{0, 1\}$ given that $f^T L f = \sum_{(ij) \in E(G)} w_{ij}(f_i - f_j)^2 = 0$. It is worth noting that the vectors are trivially $D-$orthogonal, $f_i^T D f_j = 0$, and are also the minima of the Rayleigh quotient defined on matrix pencil L, D. This minimum value holds for $\mathcal{L}$ as well, consult Table 2.1 for the relationship between the eigenstructure of matrix pencil L, D and $\mathcal{L}$

The eigenvalue $\lambda_2(L)$ is useful in bounding the "isoperimetric number" of a graph. While the eigenvalue $\lambda_2(\mathcal{L})$ bounds the "conductance" and "normalized 2-cut" of the graph (see §2.3). The bound on conductance was first shown using the second largest eigenvalue of the transition matrix, $T = D^{-1}A$, by [JS89] (see Table 2.1 for the relationship between T and $\mathcal{L}$), and is connected to mixing times of random walks on G. For an incomplete graph $\lambda_2(\mathcal{L}) \leq 1$, and the maximum value of $\lambda_n(\mathcal{L}) = 2$. The eigenvalue condition, $\lambda_n(\mathcal{L}) = 2$, indicates that G is 2 colorable (*i.e.* bipartite). A detailed treatment of the relationship between the spectrum of $\mathcal{L}$ and the properties of G may be found [Chu92].

A Simple Example: the path graph $\mathcal{P}_3$

Let $\mathcal{P}_3 = (V, E, w)$ denote a path graph on three vertices connected along a straight line, thus $|V| = 3$, $|E| = 2$, and weighted by w (unit below). Recall the Laplacian associated with this graph is constructed from two matrices, the weighted adjacency matrix A and weighted degree D:

$$A \circeq \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad D \circeq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = Diag(A\mathbf{1}_3). \tag{2.2.1}$$

Thus

$$L \circeq \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} = \begin{bmatrix} d_1 & -w_{12} & 0 \\ -w_{12} & d_2 & -w_{23} \\ 0 & -w_{23} & d_3 \end{bmatrix} = D - A \tag{2.2.2}$$

The Laplacian amounts to a vectorization of the quadratic form $\sum_{(i,j)\in E} w_{ij}(f_i - f_j)^2$ on vertex valuations $f : V \to \mathbb{R}$, . For L as above

$$\begin{aligned}
f^T L f &= f_1(d_1 f_1 - w_{12} f_2) + f_2(-w_{12} f_1 + d_2 f_2 - w_{23} f_3) + f_3(-w_{23} f_2 + d_3 f_3) \\
&= w_{12}(f_1^2 - f_1 f_2) + f_2(-w_{12} f_1 + (w_{12} + w_{23}) f_2 - w_{23} f_3) + w_{23}(f_3^2 - f_2 f_3) \\
&= w_{12}(f_1^2 - f_1 f_2) + w_{12}(f_2^2 - f_1 f_2) + w_{23}(f_2^2 - f_2 f_3) + w_{23}(f_3^2 - f_2 f_3) \\
&= w_{12}(f_1^2 - 2 f_1 f_2 + f_2^2) + w_{23}(f_2^2 - 2 f_2 f_3 + f_3^2) \\
&= w_{12}(f_1 - f_2)^2 + w_{23}(f_2 - f_3)^2 \\
&= \sum_{(i,j)\in E} w_{ij}(f_i - f_j)^2
\end{aligned} \tag{2.2.3}$$

In getting from line 1 to line 2 the degrees are expanded out as $d_1 = w_{12}$, $d_2 = w_{12} + w_{23}$, and $d_3 = w_{23}$. The terms are reorganized yielding the final form in (2.2.3). For our path graph the expression simplifies to $\mathbf{f}^T L \mathbf{f} = (f_1 - f_2)^2 + (f_2 - f_3)^2$.

2.2.1 The Differentials of Laplacian Eigenstructures

The simple eigenvalues and eigenvectors of a matrix (or matrix pencil) are smooth functions of the matrix entries. Their differentiability follows from examining the characteristic polynomial $det\ C - \lambda B$ of the matrix pencil C, B. As the eigenvalues are the roots of a polynomial they vary smoothly with its coefficients, in this case the entries of the matrices C and B. Locally these roots are implicit functions of the coefficients, and therefore proving the existence of their differentials hinges on satisfying the requirements of the implicit function theorem. We quickly cover some results in this area for the matrix pencil L, D, a more detailed treatment of matrix differential calculus may be found in [Lax97, MN99].

We begin with the explicit formulae for the differentials of the implicit functions $\lambda(L)$ and

$f(L)$, obtained by differentiating $Lf = \lambda Df$ as follows:

$$(d\,L)f + L(d\,f) = (d\,\lambda)Df + \lambda(d\,D)f + \lambda D(d\,f). \tag{2.2.4}$$

Solving for $(d\,\lambda)$, at (λ_0, f_0), in (2.2.4) we obtain:

$$\begin{aligned} (f_0^T D_0 f_0)d\lambda &= f_0^T(L - \lambda D)(d\,f) - f_0^T((d\,L) - \lambda(d\,D))f_0 && (2.2.5)\\ &\equiv \\ (d\,\lambda) &= f_0^T(d\,L - \lambda_0(d\,D))f_0 && (2.2.6) \end{aligned}$$

pre-multiplying (2.2.4) by f_0^T grouping and canceling terms we arrive at (2.2.6). The key observations are that we may normalize f_0 as $f_0^T D_0 f_0 = 1$ and that $(d\,f)^T(L - \lambda_0 I)f_0 = 0$ as $f_0 \in Ker(L - \lambda I)$ as L is symmetric. The explicit formula for $(d\,f)$ follows from manipulation of (2.2.4).

$$\begin{aligned} (L_0 - \lambda D_0)d\,f &= ((d\,\lambda)D_0 + \lambda_0(d\,D) - d\,L)f_0 && (2.2.7)\\ (d\,f) &= (L_0 - \lambda D_0)^\dagger((d\,\lambda)D_0 + \lambda_0(d\,D) - d\,L)f_0 && (2.2.8) \end{aligned}$$

where $M^\dagger$ denotes the Penrose-Moore pseudo-inverse of M. Note, for large sparse systems it is far more efficient to solve for $(d\,f)$ with an approximate linear solver than to compute the dense matrix $(L_0 - \lambda_0 D_0)^\dagger$.

The second order differentials $(d^2\,\lambda)$ and $(d^2\,f)$ are quickly determined by applying the linear operator d to (2.2.4). Again we begin with the eigenvalue.

$$\begin{aligned} 2(dL) + L(d^2 f) &= (d^2\lambda)Df + \lambda D(d^2 f) + 2(d\lambda)(dD)f + (d\lambda)(dD)(df) && (2.2.9) \\ d^2\lambda &= f^T(2(dL) - (d\lambda)(dD))(df) - 2(d\lambda)(dD)(df) && (2.2.10) \\ &= f^T(2(dL) - (d\lambda)(dD))(L - \lambda D)^\dagger((d\lambda)D + \lambda(dD) - (dL)) - 2(d\lambda)(dD))f \end{aligned}$$

Note that the above shows that the second derivative can be written so as to be independent of the differential (df).

Existence of the Differentials

We prove that $\lambda(L, D)$ and $f(L, D)$ satisfy the implicit function theorem and are thereby differentiable functions of the matrix pencil L, D. In the following section λ is assumed simple, and D is assumed full rank. We will show that the derivative of a joint function over the generalized eigenvalues and eigenvalues is defined around the original eigenpair f_0, λ_0, and that the derivative is zero at this point.

Theorem 1. *The implicit functions $\lambda(L, D)$ and $f(L, D)$, satisfying $L(t)f(t) = \lambda D(t)f(t)$, are smooth in a local ball $\mathcal{N}$ centered at $(f_0, \lambda_0) \mid L_0 f_0 = \lambda_0 D_0 f_0$.*

Proof. To prove the existence of the differentials of $\lambda(L, D)$ and $f(L, D)$ we define a joint function and analyze its properties. For the generalized eigenproblem we form the function $\mathcal{F}$, about f_0, λ_0 as

$$\mathcal{F}(f_0, \lambda_0;\ L) \doteq \begin{pmatrix} A \\ b \end{pmatrix} \doteq \begin{pmatrix} (L - \lambda_0 D)f_0 \\ f_0^T D f_0 - 1 \end{pmatrix} \qquad (2.2.11)$$

and note that the eigenpair f_0, λ_0 is a zero of $\mathcal{F}$. To validate $\mathcal{F}$, and thereby satisfy the *IFT*, we must insure that the smooth open ball $\mathcal{N}(f_0, \lambda_0)$ exists. The first requirement that $\mathcal{F}(f_0, \lambda_0;\ L) = 0$ follows from the definition; that a function $\mathcal{G} = \dot{\mathcal{F}}$ exists, and that $\mathcal{G}(f_0, \lambda_0;\ L) =$

0 is all that remains. We start by constructing the Jacobian of (2.2.11) is:

$$J_{\mathcal{F}} \doteq \begin{pmatrix} \frac{\partial A}{\partial f} & \frac{\partial A}{\partial \lambda} \\ \\ \frac{\partial b}{\partial f} & \frac{\partial b}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} L - \lambda D & D^{1/2} f \\ \\ f^T D^{1/2} & 0 \end{pmatrix} \tag{2.2.12}$$

as λ is simple, the Jacobian, $J_{\mathcal{F}}$, is invertible and thus the function $\mathcal{G} = J_{\mathcal{F}}^{-1}\mathcal{F}$ exists and satisfies the requirements of the *IFT*. To show this, we observe that $J_{\mathcal{F}}$ is a bordered matrix[1] and its determinant can be written as $det\ J_{\mathcal{F}} = -f_0^T D^{1/2}(L - \lambda_0 D)^{\sharp} D^{1/2} f_0$, where $B^{\sharp}$ denotes the transpose of **cofactor matrix** of B. Recall that for a non-singular matrix B, $B^{-1} = \frac{1}{detB} B^{\sharp}$. As the $(L - \lambda D)$ has co-rank$= 1$, the cofactor matrix consists of scaled versions of its nullspace space. Therefore $det\ J_{\mathcal{F}} > 0$ and $\mathcal{G}$ exists. This follows $(D^{1/2} f)^T f > 0$, for a positive diagonal operator D, and as f is in the nullspace of $(L - \lambda D)$. Given that the conditions of the *IFT* are met by $\mathcal{G}$. The simple generalized eigenvalue $\lambda(L, D)$ is a differentiable function of the entries of L and D, which themselves are linear function of the graph weighting w. □

Entry-wise representation of λ and f

In this section the contribution of each edge to the total derivation of the eigenvalue $(d\ \lambda)$ is examined. In chapter 4, Spectral Rounding and Fractional Averages, we will return to this formulation in our discussion of the weight updates that optimize eigenvalues. While not covered here, the edge-contribution for entries of $(d\ f)$ can be deduced in an analogous fashion.

The standard differential $d\ \lambda$ is easily expressed as a sum of terms over the edges of the

[1] See Magnus and Neudecker [MN99] for details on the determinants of bordered matrices.

graph G. In the case of an undirected graph we rewrite $d\,\lambda$ as

$$\begin{aligned} d\,\lambda &= \frac{f^T(d\,L - \lambda\,d\,D)f}{f^T D f} & (2.2.13)\\ &= \sum_{(ij)\in E} d\,\lambda_{ij}. & (2.2.14)\end{aligned}$$

We obtain the contribution at each edge $d\,\lambda_{ij}$ by breaking apart (2.2.13).

$$\begin{aligned} d\,\lambda &= \sum_{(ij)\in E} 2l'_{ij} f_i f_j - \lambda \sum_{k\in V} d'_k f_k^2 & (2.2.15)\\ &= \sum_{(ij)\in E} -2a'_{ij} f_i f_j + \sum_{k\in V} d'_k f_k^2 - \lambda \sum_{k\in V} d'_k f_k^2 & (2.2.16)\\ &= \sum_{ij\in E} a'_{ij}(f_i - f_j)^2 - \lambda \sum_{(ij)\in E} a'_{ij}(f_i + f_j)^2 & (2.2.17)\end{aligned}$$

where $l'_{ij} = (d\,L)_{ij}$ is the ij^{th} entry in the Laplacian, and $d'_k = (dD)_{kk}$. Recall that without parameterization of G the edge derivative is $e'_{ij} = 1$. The individual contribution can now be isolated as

$$\begin{aligned} d\,\lambda_{ij} &= a'_{ij}\left((f_i - f_j)^2 - \lambda(f_i + f_j)^2\right) & (2.2.18)\\ &= (f_i - f_j)^2 - \lambda(f_i + f_j)^2 & (2.2.19)\end{aligned}$$

or, alternately, in terms of the underlying edge weights from $G = (V, E, w)$ the contribution simplifies to

$$d\,\lambda_{ij} = 2f_i f_j - \lambda(f_i^2 f_j^2). \qquad (2.2.20)$$

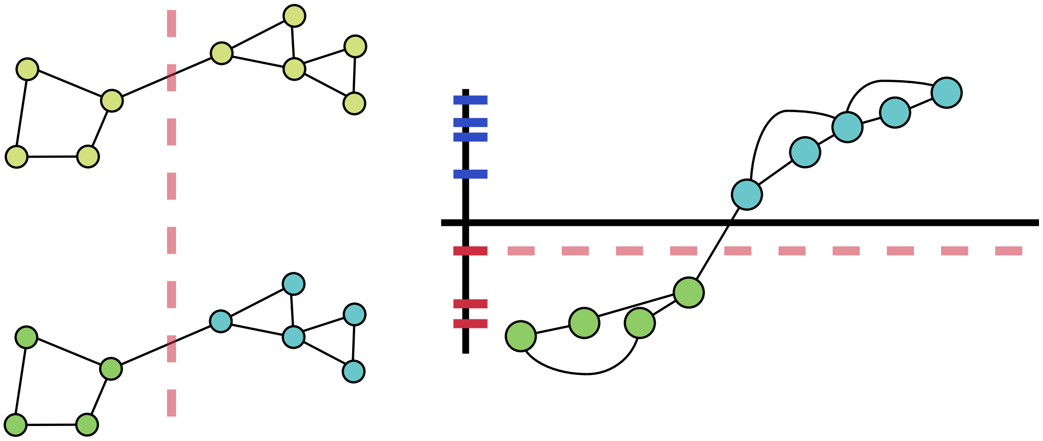


Figure 2.1: An illustration of the threshold cut algorithm studied in §2.3. The graph on the right is embedded on the real line according a vector $\mathbf{x} : V \rightarrow \mathbb{R}$. Each threshold is tested, the red starting from the bottom up and the blue from the top down, until the best threshold is found. The edge cut, illustrated with a red dashed line, is then returned separating the graph.

2.3 The Cheeger Inequality for the Normalized Cut

The normalize cut of a graph is a NP-hard combinatorial objective function over partitions of a graph. Ostensibly it seeks an equitable compromise between the cost the partitioning and the variation in volume of the individual partitions. This objective function can be written as a quadratically constrained quadratic program (QCQP) [AW00, YS03a].

Normalize Cut QCQP

The definition of the normalized cut (see definition 6) can be vectorized as an optimization over the feasible set $\mathcal{B}$ satisfying the constraints below:

$$\begin{aligned} \min_{B \in \mathcal{B}} \quad &: \quad \frac{1}{k} \sum_{i=1}^{k} \frac{B^T L B}{B^T D B} \\ s.t. \quad &: \quad B^T D B = I_k \\ &: \quad B_{ij} = \left\{ 0, \frac{1}{\sqrt{vol(V_j)}} \right\}. \end{aligned} \tag{2.3.1}$$

The two common paradigms for approximating such objective functions are 1) linear or semidefinite programming [LR88, XJ03, ARV04]. and 2) spectral methods [CGT94, Chu92, SM00, NJW02]. Most spectral methods consist of a two stage algorithm. In the first stage a small collection of, say k, eigenvectors with small eigenvalues are computed. While in the second stage these vectors are used to map the graph vertices into $\mathbb{R}^k$ and a geometric separator is then applied [CGT94].

This section provides an approximation bound on the Normalized Cut using the eigenvalues of the normalized Laplacian. The algorithm is a simplified version of the standard algorithm used by [CGT94, SM00, YS03a]. First a detailed proof of the bound on a normalized $2-$cut is given, followed by a simple sketch of the extension to $k-$way normalized cuts. The later being a $k-$fold application of the argument below.

We now provide an adaptation of Stephen Guattery and Gary Miller's unpublished note on the "best cut from a vector" using the the eigenvectors of the generalized graph Laplacian. The proof is adapted to the **normalized Laplacian** and the associated bound to one on the **normalized 2-cut** of the graph (rather than the "sparsest cut"). Some of the text in is included "as is" from their original note.

The algorithm employed in obtaining the bound uses a vector $\mathbf{x}$, called a valuation, which is a map from $\mathbf{x} : V \to \mathbb{R}$. The vertices are then embedded on the line according to $\mathbf{x}$ and a threshold is selected such that the cut cost is minimized (see Figure 2.1). Such a cut is referred to as a **threshold cut** of the graph G given $\mathbf{x}$.

Theorem 2. *Let G be a connected graph with positive edge weights on n vertices with generalized Laplacian L and weighted degree matrix D. For any vector $\mathbf{x}$ such that $\mathbf{x}^T D\vec{1} = 0$, let nc^* be the smallest normalized 2-cut over all threshold cuts based on $\mathbf{x}$. Then*

$$nc^* \leq \sqrt{2\frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T D\mathbf{x}}}.$$

Proof. Assume w.l.o.g. that the vertices of the graph are numbered such that the entries of $\mathbf{x}$

occur in non-increasing order: for $i < j$, $x_i \geq x_j$. Let P be the generalized sum Laplacian, defined as $P = D + A$.

We start with two facts about quadratic terms of generalized Laplacians and sum Laplacians. In the expressions below, let $\mathbf{z}$ be any real vector. First, the following fact is well known:

$$\mathbf{z}^T L \mathbf{z} = \sum_{(i,j) \in E(G)} w_{ij} \left(z_i - z_j\right)^2 \tag{2.3.2}$$

Second,

$$\begin{aligned} c\left(\mathbf{z}^T L \mathbf{z}\right)\left(\mathbf{z}^T P \mathbf{z}\right) &= c\left(\sum_{(i,j) \in E(G)} w_{ij} \left(z_i - z_j\right)^2\right)\left(\sum_{(i,j) \in E(G)} w_{ij} \left(z_i + z_j\right)^2\right) \\ &= c\left(\sum_{(i,j) \in E(G)} \left(\sqrt{w_{ij}}\,|z_i - z_j|\right)^2\right)\left(\sum_{(i,j) \in E(G)} \left(\sqrt{w_{ij}}\,|z_i + z_j|\right)^2\right) \\ &\geq c\left(\sum_{(i,j) \in E(G)} w_{ij} \left|z_i^2 - z_j^2\right|\right)^2, \end{aligned} \tag{2.3.3}$$

where $c = \frac{1}{(\mathbf{z}^T D \mathbf{z})^2}$, and the third line follows from the Cauchy-Schwarz inequality.

It is useful to give a high-level outline of the proof here before proceeding: we have just shown that the product $\left(\mathbf{x}^T L \mathbf{x}\right)\left(\mathbf{x}^T P \mathbf{x}\right)$ provides a connection between $\mathbf{x}^T L \mathbf{x}$ (which is the weighted sum of squares of differences across edges) and a weighted sum of differences of the squares of the values at the ends of edges. The second sum telescopes, and can be neatly divided up in terms of subintervals of the the interval from x_i to x_j. This will allow us to break an edge up into a number of pieces corresponding to the number of thresholds (and hence cuts) that it crosses. We will rewrite the last sum in (2.3.3) as a weighted sum of cut quotients to prove the theorem. First an issue must be addressed. Any edge that crosses zero is a potential problem for the application of telescoping. To resolve this we break the contribution of an edge into (positive) contributions over subintervals. For an edge (i, j) crossing the zero point, the sum of the contributions could be bigger than the difference $w_{ij}\ |x_i^2 - x_j^2|$. This could violate the inequalities used to show the upper bound. Therefore it is useful to make two changes: we shift the values of $\mathbf{x}$ so that $x_T = 0$ where T is the index into V such that $\sum_{i=1}^{T} d_i$ closest to $\frac{1}{2}vol(G)$; and we modify G by breaking any edge that crosses the zero point into two parts, one part from x_i to a vertex with value zero, and one part from the zero vertex to x_j; each of these parts is assigned weight w_{ij}. The next section shows that these changes don't affect the preceding upper bound much.

Let G' be the graph modified as specified in the previous paragraph; G' has Laplacian L'. Let $\mathbf{z}$ be any nonzero vector such that $z_i \geq z_j$ for all $i < j$ and $z_T = 0$. Then with respect to equation (2.3.2), $\mathbf{z}^T L'\mathbf{z}$ and $\mathbf{z}^T L\mathbf{z}$ differ only in the terms for edges that go from some vertex $i < T$ to some vertex $j > T$. Note that for each such edge we have

$$(z_i - z_j)^2 = z_i^2 + z_j^2 - 2z_i z_j > z_i^2 + z_j^2 = (z_i - 0)^2 + (0 - z_j)^2,$$

where the inequality holds because z_i and z_j have opposite signs by our restriction on the ordering of $\mathbf{z}$ (the edge weight has been factored out of each expression). As the total degree of G' is greater than G, $z^T D' z \geq z^T D z$, and thus we have

$$\frac{\mathbf{z}^T L'\mathbf{z}}{\mathbf{z}^T D'\mathbf{z}} \leq \frac{\mathbf{z}^T L\mathbf{z}}{\mathbf{z}^T D\mathbf{z}} \tag{2.3.4}$$

for any such vector.

Now consider the shifted version of $\mathbf{x}$: Let $\mathbf{y} = \mathbf{x} + \alpha\mathbf{1}$ where $\alpha = -x_T$. We have the following:

$$\frac{\mathbf{y}^T L\mathbf{y}}{\mathbf{y}^T D\mathbf{y}} = \frac{(\mathbf{x} + \alpha\mathbf{1})^T L(\mathbf{x} + \alpha\mathbf{1})}{(\mathbf{x} + \alpha\mathbf{1})^T D(\mathbf{x} + \alpha\mathbf{1})} = \frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T D\mathbf{x} + 2\alpha\mathbf{x}^T D\mathbf{1} + \alpha^2 \sum d_i} < \frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T D\mathbf{x}},$$

where the second equality follows from the restriction $\mathbf{x}^T D\mathbf{1} = 0$ from the theorem statement, and from the fact that $\mathbf{1}$ is the (simple) zero eigenvalue for any (generalized) Laplacian. Since $\mathbf{y}$ meets the restrictions on $\mathbf{z}$ in the preceding paragraph, we can combine this result with inequality (2.3.4) to get

$$\mathbf{y}^T L' \mathbf{y} \leq \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T D \mathbf{x}} \cdot \mathbf{y}^T D' \mathbf{y}. \tag{2.3.5}$$

We can perform a similar analysis for P', the sum Laplacian of G':

$$\frac{\mathbf{y}^T P' \mathbf{y}}{\mathbf{y}^T D' \mathbf{y}} \leq \frac{\mathbf{y}^T P \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} = \frac{\mathbf{y}^T (2D - L) \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} < \frac{\mathbf{y}^T (2D) \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} = 2. \tag{2.3.6}$$

thus

$$\mathbf{y}^T P' \mathbf{y} \leq 2 \mathbf{y}^T D' \mathbf{y}. \tag{2.3.7}$$

The first inequality follows from the fact that L' is positive semidefinite, and that $\mathbf{y}$ is not a multiple of the "all ones" vector, the only zero eigenvalue of L'. Combining inequalities (2.3.3), (2.3.5), and (2.3.6), we get

$$2 \cdot \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T D \mathbf{x}} \cdot \left(\mathbf{y}^T D' \mathbf{y}\right)^2 \geq \left(\mathbf{y}^T P' \mathbf{y}\right) \left(\mathbf{y}^T L' \mathbf{y}\right) \geq \left(\sum_{(i,j) \in E(G')} w_{ij} \, |y_i^2 - y_j^2| \right)^2 .$$

Since only nonnegative values are involved, we can take the square root of the terms above. Further, since no edges cross the zero point, we can rewrite the summation to eliminate the absolute value signs. This gives the following:

$$\sqrt{2 \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T D \mathbf{x}}} \cdot \left(\mathbf{y}^T D' \mathbf{y}\right) \geq \sum_{\substack{(i,j) \in E(G') \\ i < j \leq T}} w_{ij} (y_i^2 - y_j^2) \quad + \sum_{\substack{(i,j) \in E(G') \\ T \leq i < j}} w_{ij} (y_j^2 - y_i^2). \tag{2.3.8}$$

The rest of the proof essentially follows Mohar [Moh89]; for the application of the telescoping to work properly we must split the sum and process $\mathbf{x}$ inward from the endpoints. In this fashion the last term of a sum is $x_T = 0$, and thus only the degree terms appear in the sum. We'll actually only show the proof for the positive part of the vector, as the argument for the negative half is symmetric.

We need some notation before we can finish the proof. Note that the y_i's may not be distinct. Assume that there are k distinct values in the subvector consisting of entries y_1 through y_n, and denote them as $t_1 > t_2 > \ldots > t_{k-1} > t_k = 0$. Let δV_i be the total weight of the edges (k,l) in G' such that $y_k \geq t_i$ and $y_l < t_i$; that is, δV_i is the weight of the edges crossing the cut at threshold t_i. Let $V_i = \{j \in V(G') \mid y_j \geq t_i\}$ (for simplicity of notation below, let $V_0 = \emptyset$).

Consider the following calculation:

$$\sum_{(i,j)\in E(G')} \frac{w_{ij}\ (y_i^2 - y_j^2)}{\mathbf{y}^T D'\mathbf{y}} \quad \geq \quad \frac{1}{\mathbf{y}^T D'\mathbf{y}} \sum_{i=1}^{k-1} \delta V_i\ (t_i^2 - t_{i+1}^2) \tag{2.3.9}$$

The first step in deriving equation (2.3.9) is the application of telescoping: Let $y_i = t_l$ and $y_j = t_m$. Then $y_i^2 - y_j^2 = \sum_{i=l}^{m-1}(t_i^2 - t_{i+1}^2)$. This sum is regrouped with respect to the differences $t_i^2 - t_{i+1}^2$; each such difference is weighted by a factor equal to the weight of the edges crossing that threshold.

Recall the definition of the normalized 2-cut, and let nc_i be the normalized cut that separates V_i from the rest of the graph, and let nc^* be the minimum normalized cut produced by the vector $\mathbf{y}$. The cost of a threshold cut can thus be stated as follows:

$$nc_i = \frac{1}{2}\left(\frac{\delta V_i}{vol(V_i)} + \frac{\delta V_i}{vol(\bar{V}_i)}\right). \tag{2.3.10}$$

Note that, by the construction of G' and $\mathbf{y}$, the values for the nc_i's and nc^* are unchanged if the definitions are applied to G and $\mathbf{x}$.

Consider the following calculation:

$$\begin{aligned}
\frac{\sum_{i=1}^{k-1} \delta V_i \, (t_i^2 - t_{i+1}^2) + N}{\mathbf{y}^T D' \mathbf{y}} &= \frac{1}{\mathbf{y}^T D' \mathbf{y}} \sum_{i=1}^{k-1} nc_i \, \frac{2vol(V_i)vol(\bar{V}_i)}{vol(V_i) + vol(\bar{V}_i)} \, (t_i^2 - t_{i+1}^2) + \frac{N}{\mathbf{y}^T D' \mathbf{y}} \\
&= \sum_{i=1}^{k-1} nc_i \, \frac{2vol(\bar{V}_i)}{vol(V_i) + vol(\bar{V}_i)} \, \frac{vol(V_i) \, (t_i^2 - t_{i+1}^2)}{\mathbf{y}^T D' \mathbf{y}} + \frac{N}{\mathbf{y}^T D' \mathbf{y}} \\
&\geq \frac{1}{Z} \sum_{i=1}^{k-1} nc_i \, \frac{2vol(\bar{V}_i)}{vol(V_i) + vol(\bar{V}_i)} \, \frac{vol(V_i) \, (t_i^2 - t_{i+1}^2)}{\mathbf{y}^T D' \mathbf{y}} + \frac{N}{\mathbf{y}^T D' \mathbf{y}} \\
&= \sum_{i=1}^{t_m} nc_i \, Pr(nc_i; \mathbf{y}) \\
&\geq nc^* \sum_{i=1}^{t_m} \, Pr(nc_i; \mathbf{y}). \qquad (2.3.11)
\end{aligned}$$

We use N to denote the portion of the sum constituted by the negative half of $\mathbf{y}$. As the manipulations are symmetric the notation is simplified to reduce clutter. In the first line the normalized cut nc_i is substituted for the cut δV_i by multiplying through by a complicated instance of 1. The telescoping sum reappears as $\sum_{i=1}^{k-1} vol(V_i) \, (t_i^2 - t_{i+1}^2) = \sum_{i=1}^{k-1} y_i^2 d_i$ and thus the ratio $\frac{vol(V_i) \, (t_i^2 - t_{i+1}^2)}{\mathbf{y}^T D \mathbf{y}}$ sums to 1 over the entire vector $\mathbf{y}$. To preserve the inequality $Z \geq 1$ must hold, we see this obtains as $\frac{2vol(\bar{V}_i)}{vol(V_i)+vol(\bar{V}_i)} > 1$. This is accomplished by working from the positive and negative extreme valuations of $\mathbf{y}$ and then toward the origin. Thus the smaller shore of the cut is always represented by V_i and the larger as $\bar{V}_i$. Finally, by applying (2.3.8) we obtain the bound. □

Corollary 1. *The tightest bound on the normalized-cut $nc(G)$ of a connected graph, given L, comes from $\mathbf{y} \mid L\mathbf{y} = \lambda_2 D\mathbf{y}$, where $\lambda_1 = 0 < \lambda_2 \leq ... < \lambda_n$. Thus*

$$nc(G) \leq \sqrt{2\lambda_2}.$$

Quality of the Spectral Bound

Spectral methods are so named because the second smallest eigenvalue λ_2 of the **normalized Laplacian** bounds the best cut obtained from a continuous vector. As shown above, the associated eigenvectors provide a means of obtaining a discrete solution that satisfies the eigenvalue bound. The upper bound on $\Phi(G)$ is loose in general, as demonstrated by the pathological graphs constructed by Guattery and Miller in [GM98]. While, guaranteed $O(\frac{1}{\sqrt{n}})$ cut bounds were exhibited for planar graphs in [ST96].

Equation 2.3.1 has been studied in the context of image segmentation in the vision community [SM00, YS03a] and clustering in the learning community [NJW02, XJ03]. In all cases a standard spectral algorithm is used. The methods [SM00, YS03a, NJW02] differ primarily in how the eigenvectors are used to find a feasible solution satisfying the constraints in Equation 2.3.1.

2.3.1 Generalizing the Cheeger bound for nc_k

The proof of Theorem 2 can be extended to the case of a $k-$way normalized cut of the graph G. Here we sketch out a high level summary of our strategy. The proof hinges on $k-1$ applications of the Cheeger inequality to a series of sub-set restricted valuations of graph. The substitution of a simple edge sum for a $2-$cut (see equation 2.3.11) will be replaced with a busier term encapsulating the $k-$way cut prior. Much like Theorem 2 the tightest bound will correspond to using the $D-$orthogonal vectors with smallest Rayleigh quotient, *i.e.* first k generalized eigenvectors of L, D. The requirement that $k-1$ vectors be used, rather than $k-1$ successive threshold cuts of Fiedler vector, is necessary to insure that both a sufficient number of threshold exist and that balancing conditions are met.

Theorem 3. *Let G be a connected graph with positive edge weights on n vertices with generalized Laplacian L. Given a set of k vectors $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ such that $\mathbf{x}_i^T D \mathbf{x}_j = 0$ the* *smallest normalized $k-$cut nc_k^* over successive threshold cuts of $\mathcal{X}$ satisfies*

$$nc_k^* \leq \sum_{i=1}^{k} \sqrt{2\frac{\mathbf{x}_i^T L\mathbf{x}_i}{\mathbf{x}_i^T D\mathbf{x}_i}}$$

Proof Sketch. Apply the argument in the proof of the Cheeger constant k times. The critical difference is that volume term, substituted into the calculation (2.3.11) must be modified to the general form, for the i^{th} partition $\frac{k\prod_{j=1\neq i}^{k} vol(V_j)}{\sum_{j=1\neq i}^{k}\prod_{l<j,l\neq j} vol(V_j)vol(V_l)}vol(V_i)$. This term has multiple peaks at $\frac{1}{k}$ terms rather than the peak at $\frac{1}{2}$ shown in figure 2.3. The full set of eigenvectors, up to λ_k must be used to insure that a $k-$way cut is made, required by theorem 3.

2.3.2 Relating Isoperimetric Number and Normalized 2-Cut

In this section we relate the graph **conductance** and the **normalized 2-cut** of a graph. Below, **vol**, is used to denote the sum of weighted edges associated with a set.

We begin by illustrating how the $nc(G)$ and $\Phi(G)$ can be made to diverge in a graph dependent fashion. Is the partitioning induced by $\Phi(G)$ optimal for $nc(G)$? Obviously, the answer is no. A graph in which the $nc(G)$ partitioning differs from the $\Phi(G)$ partitioning is show in Figure 2.2.

Lemma 2. *The normalized 2-cut is bound by the conductance of the graph*

$$nc(G) \leq \Phi(G).$$

Proof. For a graph $G = (V, E, w)$ define the function $\Delta(U) = \frac{vol(U,V\setminus U)}{vol(U)}$ on the subsets of V. Let V_1 be the subset such that $\Phi(G) = \Delta(V_1)$, and $V_2 = V \setminus V_1$. That $\Phi(G)$ is an upper bound on nc_{opt} is entailed by the following observation $\Delta(V_1) \geq \Delta(V_2)$ as $vol(V_2) \geq vol(V_1)$. This yields the following bound

$$\Phi(G) = \frac{1}{2}\left(\Delta(V_1) + \Delta(V_1)\right) \geq \frac{1}{2}\left(\Delta(V_1) + \Delta(V_2)\right) \geq nc_{opt} \tag{2.3.12}$$

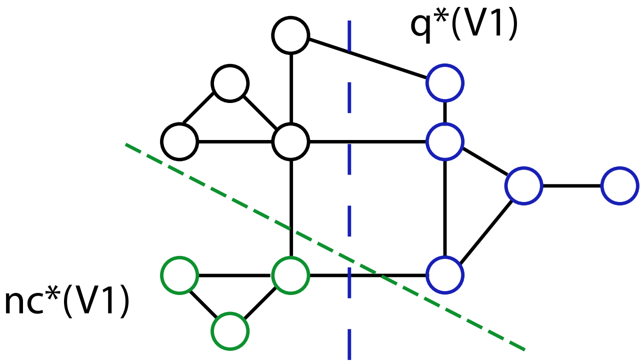


Figure 2.2: In the above graph $nc(G) = \frac{1}{2}(\frac{2}{5} + \frac{2}{14})$ and $\Phi(G) = \frac{3}{8}$. The green vertices above constitute the $nc_{opt}(V_1)$ while the blue vertices constitute $\Phi(G)(V_1)$. The normalized cut value associated with $\Phi(G)$ is $nc_{opt} < nc(\Phi(G)) = \frac{1}{2}(\frac{3}{8}+\frac{3}{11})$ and the $q-$value $\Phi(G) < q(nc_{opt}) = \frac{2}{5}$. Thus the two smaller partitions have no overlap and the conjecture is untrue.

as the optimal q-partition V_1, V_2 provides an upper bound on the NCut value of G □

Theorem 4. *There exists a* $\mathbf{c} \mid \frac{1}{2} \leq \mathbf{c} \leq 1$ *such that* $nc_{opt} = \mathbf{c} \cdot \Phi(G)$.

Proof. Part 1, $\mathbf{c} \leq 1$, follows from lemma 2. Part 2, $\frac{1}{2} \leq \mathbf{c}$, associate with $\Phi(G)$ the optimal partition of vertices into two sets V_1 and V_2. Given such a partition we assume that V_1 is the smaller without loss of generality. Define a subset of V_2, the interior $V_{\mathcal{I}}$, as the set of vertices that do not share an edge with V_1. By looking at definition 7 we see that edges may be added arbitrarily to $V_{\mathcal{I}}$ without effecting the value $\Phi(G)$, nor the optimal value, while lowering nc_{opt}. In limit this provides a lower bound on nc_{opt} as $\frac{1}{2}\Phi(G)$. This bound holds, as the a cheaper cut than $\frac{1}{2}\Phi(G)$ would imply that there exists a V_1' such that $\Delta(V_1') < \Delta(V_1)$, which would contradict the assertion that $\Phi(G)$ is optimal.

Corollary 2. *The best normalized cut value derived from a vector* $\mathbf{y}$ *associated with* λ_2, nc^*, *is bound by the second smallest eigenvalue* λ_2 *of the normalized laplacian as*

$$\frac{1}{2}\lambda_2 \leq nc^*(G) \leq \sqrt{2\lambda_2}$$

Proof. By applying lemma 2 we have $nc^* \leq \Phi^*$, we then apply Cheeger inequality to Φ to obtain $\Phi^* \leq \sqrt{2\lambda_2}$, thus $nc^* \leq \Phi^* \leq \sqrt{2\lambda_2}$. The lower bound is entailed as half the

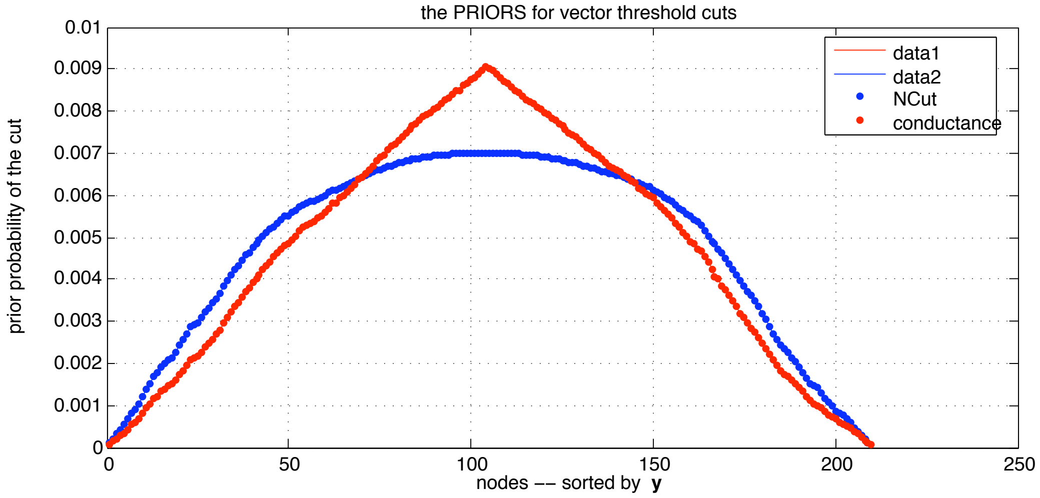


Figure 2.3: A comparison of priors, the blue contour indicates the prior over cuts induced by the normalized cut and the red that induced by the conductance.

lower bound on $\frac{1}{2}\lambda_2 \leq \Phi^*$. We tighten it to $\frac{1}{2}$ by applying Ky Fan's dominance theorem over contiguous eigenspaces. □

Comparing the Conductance and Normalized Cut

The distribution over cuts induced by a vertex valuation (vector) $\mathbf{y}$ for $nc_2(G; \mathbf{y})$ is given in (2.3.11), this differs from the distribution over ϕ given $\mathbf{y}$ which is $Pr(\phi_j \ \mathbf{y}) = \frac{vol(V_i)\Delta_i}{\mathbf{y}^T D\mathbf{y}}$. The derivation of these two distributions suggests a natural randomized algorithm for achieving the Cheeger bound (in expectation). The choice of a cut function acts like a prior over volume terms on the distribution of feasible cuts in $\mathbf{y}$. A comparative plot of the two volume priors is shown in figure 2.3.

For each of these cut functions we assign a probability that each edge $e \in E(G)$ is in the cut. Let $e \in E(G)$ and let $T(e)$ denote the set of indices for the threshold values that the edge e spans in $\mathbf{y}$. Thus the probability that an edge is in the "conductance cut" is computed as:

$$P(e \in \delta V; \phi, \mathbf{y}) = \sum_{k \in T(e)} Vol(V_k) \frac{\Delta_k}{\mathbf{y}^T D\mathbf{y}} \tag{2.3.13}$$

in the normalized cut as:

$$P(e \in \delta V; nc, \mathbf{y}) = \sum_{k \in T(e)} \frac{1}{Z} \frac{2vol(\bar{V}_k)vol(V_k)}{vol(V_k) + vol(\bar{V}_k)} \frac{\Delta_k}{\mathbf{y}^T D\mathbf{y}}. \qquad (2.3.14)$$

2.4 A Generalization of Fiedler's Theorem

The following lemma is attributed to both Colin de Verdière and van der Holst. The following bounds the number of **signed connected components** induced in a graph by eigenvectors of the Adjacency matrix. Here the same general lemma is provided for the generalized graph Laplacian $L(G)$.

Lemma 3. *Let $G = (V, E)$ be a connected graph, and λ an eigenvalue of $L(G)$ with multiplicity s such that there are r eigenvalues smaller than λ. Let $\mathbf{x}$ be an eigenvector in the eigenspace λ, and let a, b, and c denote the number of connected components spanned by $supp_+(\mathbf{x})$, $supp_-(\mathbf{x})$, and $supp(\mathbf{x})$ respectively.*
Then:

1. $c \leq s$
2. *L has at least $a + b$ eigenvalues $\lambda^{‘}|\lambda^{‘} \leq \lambda$.*
3. $a + b \leq r + c$
4. *if $\mathbf{x}$ has minimal support, of all vectors in the span of the λ eigenspace then $a+b \leq r+1$.*

See Lovàsz's notes for an Algebraic Graph Theory course offered at the University of Washington in the Spring of 2005 for a proof.

Chapter 3

Spectral Segmentation

Now we turn our attention to “spectral segmentation”, also dubbed somewhat incorrectly as *NCuts* in the vision community. We use spectral segmentation to denote the family of algorithms that seek a segmentation of the image by processing the eigenstructure associated with the image graph. These graphs are generally mesh-like in topology, as the pixel labels are expected to be spatially coherent.

The image segmentation problem, and more generally clustering, can be formulated as a set of m measurements $\{w_{ij}\}$ on n observables defining an undirected graph $G = (V, E, w)$ where $|V| = n$ and $|E| = m$. In this chapter we show that the spatially localized graph structure used in image processing applications is reflected in the graph’s eigenvalues and eigenvectors. All positive weightings of the graph topology share common properties in their eigenspaces. This is in part a consequence of theorem 3, which characterizes the number **sign connected components** an eigenvector of a generalized Laplacian may induce in the graph [Fie73, CdV90, GG01] - *i.e.* over all positive weightings of $E(G)$.

Given an image, we define n image sites, again these could be picture elements, or features associated with pixels, feature detection sites, or simply data-points in some space that we wish to cluster. Given such a collection there are $(n-1)n/2$ possible symmetric pairwise compar-

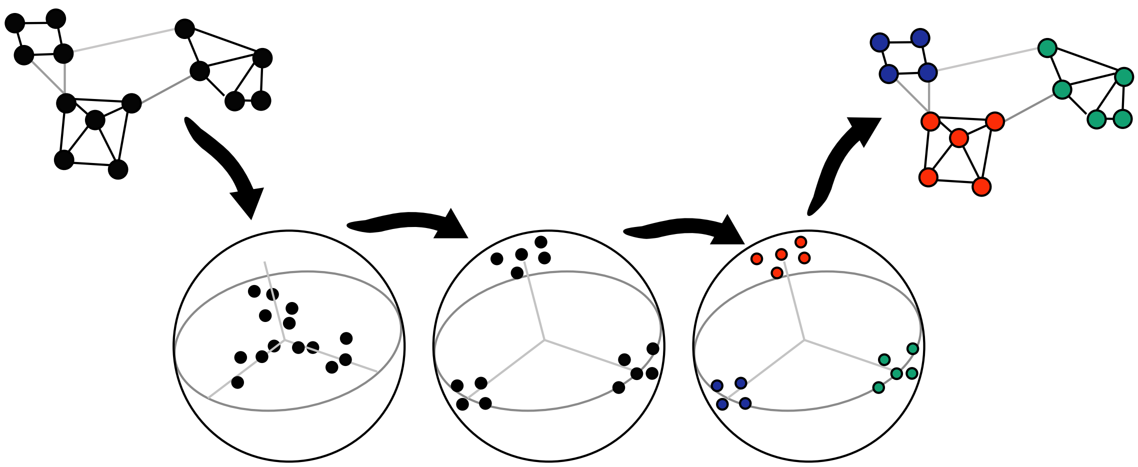


Figure 3.1: Compute a set of generalized eigenvectors of the Laplacian of the graph G. The row entries of these vectors are used to provide a coordinate for each vertex in the graph. The vertex-points are then projected onto the unit sphere. These coordinates are partitioned, on the sphere, using $k-$means or random hyperplanes (through the center of the sphere)

isons. For large vertex sets, such as those found in image segmentation, all $(n-1)n/2$ pairwise comparisons can not be represented in memory. The particular restricted set of comparisons we choose to make will determine both the cost structure of our combinatorial cut problem and may bias our relaxation toward particular cuts. In the following sections we will examine examples how these decisions affect both the optimal cut, with respect to measurement confidence, and the output of standard spectral algorithms.

3.1 Spectral Partitioning

Before beginning it is worth recalling the standard spectral algorithm (analogous to that used in [CGT94, SM00, NJW02, YS03a]). As a rule a small set of, say k, eigenvectors of the Laplacian (normalized or otherwise) with small eigenvalue provide an embedding of the graph vertices in $\mathbb{R}^k$. These points are then projected onto the unit sphere and clustered using a geometric heuristic such as $k-$means or random hyperplanes. An illustration of the process can be found in figure 3.1. A MATLAB-script description of the algorithm follows.

Algorithm §**3.1** *Given a weighted graph* $G = (V, E, w)$*, construct the matrices* $A,\ D,\ L$ *and return the partition indicator matrix* P*.*

1. $[G, \Lambda] = \texttt{eigs}(D^{-1/2} L D^{-1/2},\ k)$
2. $F = D^{-1/2} G \Lambda_{+}^{-1/2}$
3. $V_{coords} = F./\texttt{sqrt}((F^{.2})\mathbf{1}_k)\mathbf{1}_k^T$
4. $P = \texttt{round}(V_{coords},\ k)$

While this basic method lies at the heart of most spectral segmentation and clustering algorithms there are a number of variations on the theme. Should one use the eigenvalues or not? Some authors motivate the use of the $\mathbf{1}$s vector [YS03a], recall that $\mathbf{1}^T L \mathbf{1} = 0$, while most do not [CGT94, SM00, NJW02]. The routine `round` can be any number of geometric clustering tools – *e.g.* `k-means`, `alignment` [YS03a], or random hyperplanes. The number of partitions generated by each algorithm is generally determined by the engineer. However, there is a great deal of literature on methods to automatically estimate this parameter (and others) [RGDP01, TMVJ02, RLBB06]. Including multi-scale methods which provide a collection segmentations for each data-set (*e.g.* [AG06]). In chapter 4 we suggest a method for selecting eigenvectors and the number of clusters based on the differential structure of the eigenvalues of the normalized Laplacian (see 4.4.3). Notes on technical results on the behavior of **Algorithm 3.1** may be found in §2.3.

3.2 Spectral Segmentation of Images

The segmentation problem differs from traditional clustering in that the set of comparisons between points (pixels, feature sites, etc) is fixed beforehand. In clustering, comparisons are made in spatial neighborhoods of a metric feature space – *i.e.* determined by the data. In image processing the graph structure is meant to enforce local label consistency in the image plane. We will call this fixed set of edges, or comparisons, the topology of the graph.

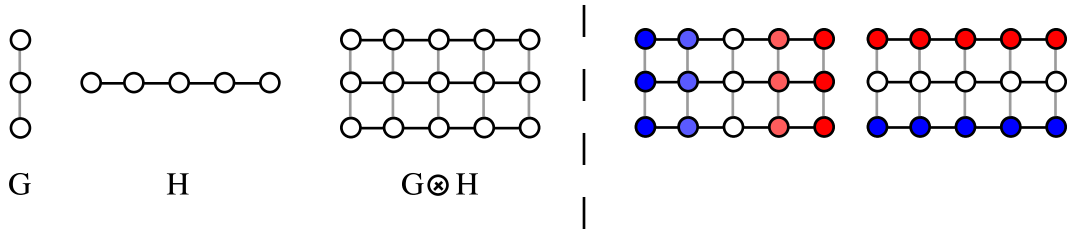


Figure 3.2: Left: the outer product graph, $M = G \otimes H$, of the graphs G and H. Right: the first two harmonic eigenfunctions of $L(M), D(M)$. Note, that each function is either a copy of an eigenfunction of G or H, (red and blue indicate positive and negative values respectively).

Thus the graphs used in image segmentation have mesh-like topology. That is, pixels are wired to neighboring pixels in a regular fashion (although perhaps with a random sparsity pattern). This highly localized topology can be contracted onto the mesh defined on a smaller number of vertices without loss of generality. Accordingly, we analyze the properties of weighted 4-connected meshes. Such mesh-like topologies affect both the optimal point of the *combinatorial problem* “normalized cut”, and the *relaxed instance* of partitioning with eigenvectors.

3.2.1 The Mesh

Imagine an image of size 220×400 pixels (roughly a $9 : 16$ ratio). As is standard practice, we’ll wire up pixels to their neighbors in a $4-$connected lattice – called the mesh. How are the low-frequency eigenspaces of the mesh affected by its topology? If we construct a mesh with this aspect ratio what cut properties can we expect?

Before answering these questions we quickly introduce **graph automorphisms** and **graph outer products** (covered in detail in [GG01]). These tools allow us to analyze the eigenstructure of the mesh in terms of path graphs $\mathcal{P}_n$. For simplicity we will concern ourselves with the eigenstructure of the combinatorial Laplacian, $L = D - A$, rather than that of the normalized Laplacian. This choice lessons the complexity of the relationships between outer product graphs and their eigenstructure. The results are essentially the same for the normalized case, *i.e.* solutions to $Lf = \lambda Df$ rather than $Lx = \sigma x$.

Definition 10. *For graph $G = (V, E)$ an* **automorphism** *is a permutation of the vertices $\pi : V \rightarrow V$ such that if the edge $(u, v) \in E(G)$ then $(\pi(u), \pi(v)) \in E(G_\pi)$.*

The rectangular mesh has two **automorphisms**, as it is symmetric about its horizontal and vertical axes. The square mesh has two additional automorphism, as the graph may be reflected across the diagonals without altering graph connectivity.

Definition 11. *The* **outer product** *of two graphs $G = (V_g, E_g)$ and $F = (V_f, E_f)$, $H = G \otimes F$, defines a graph H where $H = (V_h, E_h)$. The vertex set $V_h = V(G) \otimes V(F)$ and edge set is taken to be the union of all edges at vertex in G and F under the outer product $V(G) \otimes V(F)$.*

Analogous to the operation suggested by figure 3.2, our mesh may be constructed by taking the graph outer product of two path graphs $G = \mathcal{P}_{220} \otimes \mathcal{P}_{400}$. As the mesh maybe constructed in this fashion we might expect that it shares many properties with $\mathcal{P}_{220}$ and $\mathcal{P}_{400}$.

The graph H, $H = G \otimes F$, inherits two main properties from G and F. First the Laplacian eigenvalues of H consistent of, exactly, all unique pairwise combinations of the individual Laplacian eigenvalues of G and F [Fie73]. Further, the eigenvectors associated with these values inherit a great deal of structure as well. For example, the second smallest eigenvalue of the mesh, $\mathcal{M}_{nn'} = \mathcal{P}_n \otimes \mathcal{P}_{n'}$, is determined by the longer path graph $\lambda_2(\mathcal{M}) = min(\lambda_2(\mathcal{P}_n), \lambda_2(\mathcal{P}_{n'}))$. Assuming that $n' > n$, this will be $\lambda_2(\mathcal{P}_{n'})$. The eigenvector associated with this value consists of n copies of the eigenvector f such that $L(\mathcal{P}_{n'})f = \lambda_2 f$. The eigenvector associated with $\lambda_3(\mathcal{M})$ consists of n' copies of g such that $L(\mathcal{P}_n)g = \lambda_2 g$. These two eigenfunctions are depicted in figure 3.2.

The eigenfunctions of $\mathcal{M}_{nn'}$ are constrained by the automorphisms of both $P_{n'}$ and $\mathcal{P}_n$. In our case, the automorphisms are reflections across the centers of the two underlying path graphs. Note that applying such a permutation twice returns the vertices to their original ordering (*i.e.* a double reflection). In [GM98] the authors prove that the eigenvector of such a graph will be either symmetric or skew symmetric relative to every automorphism of the graph. Given an automorphism π every eigenvector $f \mid Lf = \lambda f$ behaves as $f_i = f_{\pi_i}$ or $f_i = -f_{\pi_i}$. For example, figure 3.2 the left valuation is same-signed over an "up-down" reflection and flips sign over a "left-right" reflection. Each of the eigenvectors of the mesh will possess this property.

Now that we known how the eigenstructure of the mesh is constrained by its underlying generating paths, we can ask how this set of comparisons affects spectral algorithms. From this point forward we'll return to normalized Laplacians and state approximate bounds on the behavior of the cut.

The Mesh: its Generalized Laplacian Eigenvalues and the Normalized Cut

The properties of the rectangular mesh $\mathcal{M}_{nn'}$, both in terms of eigenvalues and cuts can be expressed in terms of the aspect ratio $\frac{n}{n'}$, where $n' > n$. Given an unweighted mesh it is clear that the conductance of the graph $\Phi(\mathcal{M}) = \frac{n}{\frac{1}{2}vol(\mathcal{M})}$. In the following section we will analyze the cut Φ and eigenvalues associated with the $\mathcal{M}$ under a simple re-weighting of the edges.

Let $\mathcal{M}_{nn'}$ denote the mesh defined as $\mathcal{M}_{nn'} = \mathcal{P}_n \otimes \mathcal{P}_{n'}$. For simplicity assume that $n' > n$ and that both n and n' are even. Let $\mathcal{M}_{nn'}(\alpha)$ denote the weighted mesh, where the center row consists of n' edges taking the weight $\{\alpha \mid 0 < \alpha \leq 1\}$, and all other edges are unit. The conductance of $\mathcal{M}_{nn'}(\alpha)$ is given by

$$\Phi(\mathcal{M}_{nn'}(\alpha)) \;=\; min\left(\frac{n}{\frac{1}{2}vol(\mathcal{M}_{nn'}(\alpha))}, \frac{\alpha \cdot n'}{\frac{1}{2}vol(\mathcal{M}_{nn'}(\alpha))}\right). \tag{3.2.1}$$

Therefore the conductance Φ and nc_2 are determined by testing the inequality $\alpha < \frac{n}{n'}$.

Recall the Cheeger bound on the conductance of the graph $\Phi(G)$ given a threshold cut of the graph induced by a vector $\mathbf{y} : V(G) \rightarrow \mathbb{R}$, , see Chapter 2 theorem 2. The Cheeger inequality can be reversed to bound the Rayleigh quotient associated with a cut as

$$\frac{1}{2}\Phi(G;\ \mathbf{y})^2 \leq \frac{\mathbf{y}^T L\mathbf{y}}{\mathbf{y}^T D\mathbf{y}} \leq 2\Phi(G;\mathbf{y}) \tag{3.2.2}$$

for any vector $\mathbf{y}$ such that $\mathbf{y}^T D\mathbf{1} = 0$. For the mesh, we wish to examine the relationship between the two eigenfunctions $f|Lf = \lambda_2 Df$ and $g \mid Lg = \lambda_3 Dg$ (as pictured in figure 3.2). As eigenfunctions of the matrix pencil L, D both f and g satisfy the conditions on $\mathbf{y}$ in (3.2.2). In the case of the unit weighted rectangular mesh, we can compute the values $\Phi(G)$, $\Phi(G;\ f)$ and $\Phi(G;\ g)$ directly and use them to bound the eigenvalues λ_2 and λ_3. By the definitions of f and g, given above, we arrive at :

$$\frac{1}{2}\left(\frac{n}{\frac{1}{2}vol(G)}\right)^2 \leq \lambda_2(\mathcal{L}(G)) \quad\leq\quad 2\frac{n}{\frac{1}{2}vol(G)} \tag{3.2.3}$$

$$\frac{1}{2}\left(\frac{\alpha n'}{\frac{1}{2}vol(G)}\right)^2 \leq \lambda_3(\mathcal{L}(G)) \quad\leq\quad 2\frac{\alpha n'}{\frac{1}{2}vol(G)} \tag{3.2.4}$$

where $vol(G) = \frac{1}{(4nn'+2n+n'+\alpha n'-4)}$.

The normalized cut and conductance switch to the longer horizontal cut exactly when $0 \leq \alpha < \frac{n}{n'}$. To determine the behavior of the standard spectral algorithms we must examine the bounds on λ_2 and λ_3. By inspection the lower bounds and upper bounds on λ_2 and λ_3 intersect at $\frac{n}{n'}$. This verifies that the combinatorial solution exchanges to the $\alpha-$cut when $\alpha < \frac{n}{n'}$. The lower bound on λ_2 and the upper bound on λ_3 intersect at $\frac{n^2}{2vol(\mathcal{M}_{nn'})n'}$. This is the lowest possible $\alpha-$value at which the vectors f and g exchange position in the spectral index of L, D. Therefore the solution returned by the spectral algorithm may require that $\alpha < \frac{n^2}{2\ vol(\mathcal{M}_{nn'})n'}$ in order to match the combinatorial optimal cut. This is because it uses the “wrong” eigenvector to partition G as f only spans cuts that are orthogonal to the optimal cut. In practice, we observe that the spectral solution exchanges when $\alpha \lessapprox \frac{n^\theta}{n'^4}$, where $\theta = \left(\frac{n'}{n}\right)^2$.

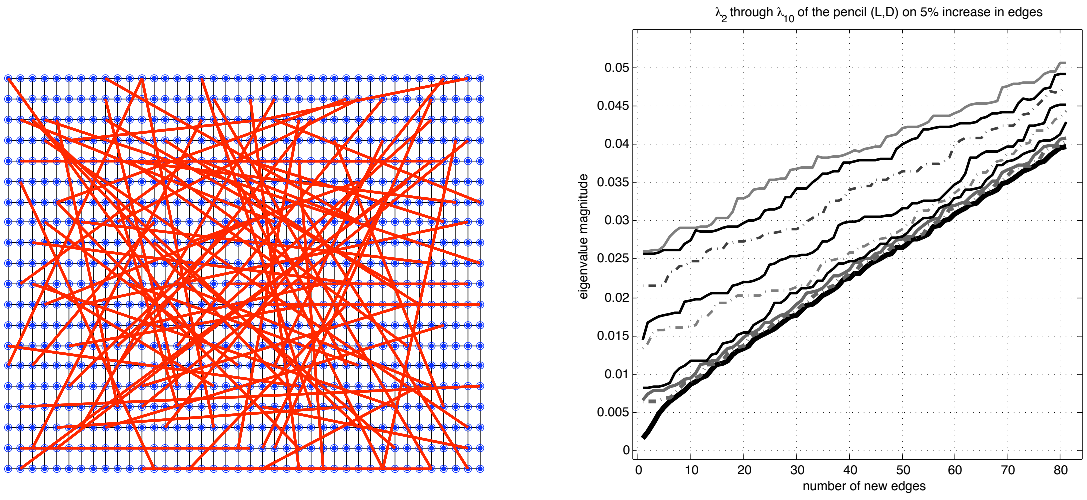
The lag between the spectral solution and the combinatorial solution has significant practical implications. First, we saw that the standard spectral algorithm may lag behind the true combinatorial optimum by a significant amount. For example, in the mesh $\mathcal{M}_{20,40}(\alpha)$, the combinatorial solution switches from short to long axis when $\alpha < \frac{1}{2}$, where as the spectral solution switches only when $\alpha < \frac{1}{16}$. In image processing applications this can be thought of as a statement on how sharp the image derived edge weights must be to get the spectral algorithm to match the NCuts criterion. This problem is illustrated by a seemingly trivial segmentation problem in figure 3.5. In the next section, we propose a method that mitigates the errors introduced through this topological bias in spectral segmentation. Further complications arise as mesh-like topologies produce sinusoidal functions in the eigenvector over regions of smooth texture in the image plane. This follows from our observations on how the eigenvectors of the graph are constrained by it's topology. Thus a constant region of the image will produce a unit weighted subgraph in the mesh – which tends toward a plane wave in the eigenvector (when viewed as an image).

3.2.2 Expansion Augmentation

Expansion augmentation attacks the topological eigen-gap problem directly by constructing sparse graph topologies with larger edge expansion. Intuitively, expanders are graphs that do not possess small cut-sets. Concretely, will we call a graph an expander if $\lambda_2(\mathcal{L})$ is sufficiently large. An example of such a graph, and the partitioning it induces is shown in Figure 3.5. As the topology contains no sensible spatial cuts a priori, the image derived data (in the form of edge weights) dominates the structure of the eigenvectors.

Recall that for a connected graph the generalized eigenvalues of $L(G), D(G)$ are bounded and ordered as $0 = \lambda_1 < \lambda_2 \leq \ \ldots\ \leq \lambda_n \leq 2$. Let $G_{\mathcal{M}4} = (V, E_{\mathcal{M}}, w)$ and $G_{\mathcal{T}} = (V, E_{\mathcal{T}}, \mathbf{1}_m)$ denote the weighted mesh and the unweighted augmentation graph respectively. In terms of our previous observations, as $\lambda_2(G_{\mathcal{T}})$ approaches the maximal possible value[1] of $\lambda_2 \leq 1$ the

[1] Recall that the sum of the eigenvalues is equal to the trace of the matrix. For the normalized Laplacian,



Augmented Mesh

Eigenvalue Compression

Figure 3.3: The original mesh is augmented with approximate 80 edges reducing the eigen-gap $\frac{\lambda_3-\lambda_2}{\lambda_2}$ by over 99%. The expansion augmentation edges are shown above in red, while the initial mesh edges are shown in blue. The final number of edges in the augmented graph is only about 5% more than in the mesh. The plot (right) shows the concentration in the first 9 (non-zero) eigenvalues as edges are added.

eigen-gap $\lambda_k - \lambda_2$ on the graph tend to its minimum *(*0 *if* $G_{\mathcal{T}}$ *is allowed to be the complete graph* K_n*)*. This differs from mesh-reducible graphs which possess cheap spectral cuts [ST96] and large eigen-gaps. In order to generate perceptually relevant segmentations large eigen-gaps in spectrum of the unweighted mesh must be overwhelmed by the image induced weighting w. Unfortunately, this is often too much to ask.

In practice the condition that $\lambda_2(G_{\mathcal{T}}) \approx 1$ is impractical in a sparse graph. Alternately, we may fix a normalized eigen-gap threshold $\frac{\lambda_k-\lambda_2}{\lambda_2} < \theta$. To obtain topologies that satisfy this criteria we apply the expansion augmentation algorithm:

Expansion Augmentation Iteration *Given a graph* $G = (V, E)$ *augment the edge set of* G *until* $\left(\frac{\lambda_k(\mathcal{L})-\lambda_2(\mathcal{L})}{\lambda_2(\mathcal{L})}\right) < \theta$.

```
While $\left(\frac{\lambda_k(\mathcal{L})-\lambda_2(\mathcal{L})}{\lambda_2(\mathcal{L})}\right) > \theta$, do
```

and associated co-spectral matrices, this value is n. A direct consequence of the definition of $\mathcal{L} = D^{-1/2}(D - A)D^{-1/2} = I_n - D^{-1/2}AD^{-1/2}$, thus trace$(\mathcal{L}) = $ trace$(I_n) = n$. It follows that no graph on n vertices has a $\lambda_2 > \frac{n}{n-1}$ by observing that $\sum_{i=1}^{n} \lambda_i = n$ and $\lambda_1 = 0$. Further if G is not complete $1 \geq \lambda_2$.

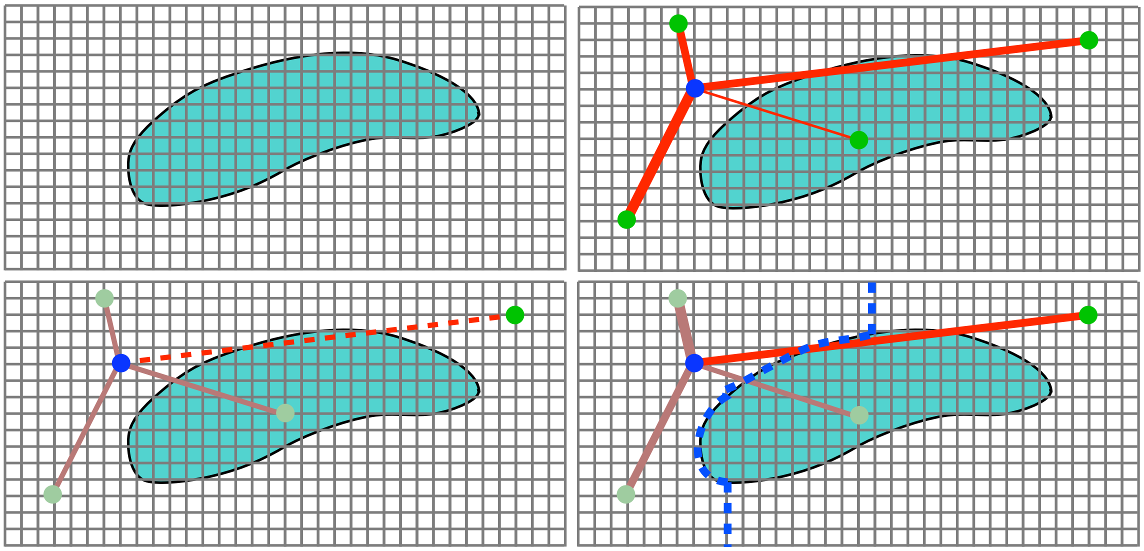


Figure 3.4: In clockwise order: the initial weighted mesh over the image pixels. An example of random edge augmentation. Such edges are added at until the target eigenvalue bound $\frac{1}{2} < \theta_\lambda \leq \lambda_2(G_\mathcal{T})$ is obtained. The edge weight is computed via a "source-sink" cut between the incident vertices (in dashed-blue) and normalized by a volume term. In contrast to the inaccurate weighting obtained intervening contour cue, as the random edges may be long range.

1. Compute the partial eigenstructure $\mathcal{L}G = G\Lambda_k$, and set $F = D^{-1/2}G$.

2. Choose a random subset $S \subset V(G)$.

3. Set $E(G) = E(G) \cup \underset{(u,v)\in S}{argmax} \; : \; ||F(u) - F(v)||_2$

end

We are guaranteed that $\lambda_2(\mathcal{L})$ increases monotonically under this iteration by Rayleigh's monotonicity principle [Bol02]. In the limit, as $G_\mathcal{T}$ tends toward K_n we must achieve our threshold, $\left(\frac{\lambda_k(\mathcal{L})-\lambda_2(\mathcal{L})}{\lambda_2(\mathcal{L})}\right) < \theta$, as the eigenvalues concentrate with each iteration. This follows by observing that the lower bound on the (non-zero) eigenvalues $\lambda_2(\mathcal{L})$ increases and that the sum of the eigenvalues is fixed. An example of the affect expansion augmentation procedure has on the mesh $\mathcal{P}_{20} \otimes \mathcal{P}_{40}$ eigenvalues is shown in figure 3.3.

Weighting the Expansion Edges

There are several possible ways to weight the new expansion edges. To maintain useful spatial structure in the cut we must derive the edge weights by processing the weights in the underlying mesh. The weighting procedure for the expansion edges breaks down into two steps. First a minimum weight path on G is found for each edge $e_{ij} \in E(G_{\mathcal{T}})$. The path cost is computed on a reflected set of edge weights $w'_{ij} = 1 - w_{ij}$, so that cheap paths correspond to heavy weight paths in G. The average of this path in G with weighting w is then used to weight the expansion edge. If the path cost $p(i, j;\ w') > \theta_p$ then a flow computation from $s : i \rightarrow t : j$ is computed in G. The threshold, θ_p, amounts to finding at least one lightweight edge on G in $p(i, j, w')$. We adopt this procedure for two reasons, the first is motivated by the approximation factor given in [KRV06] and the second is due to the inaccurate weights obtained by intervening contour computations on long range edges (see figure 3.4).

Expansion Augmentation: Two Interpretations

There are two straightforward interpretations of expansion augmentation. The first is that we are constructing an initial topology, defined over the image pixels, that has little preference over partitions of the pixels (*e.g.* compare the eigenvector and value in figure 3.5). This follows from the near multiplicity in the low frequency eigenvalues of $\mathcal{L}$:$\{\lambda_2, \ldots\ \ldots, \lambda_{k'}\}$ for $k' << n$ exhibited by augmented graphs. Thus the set of cuts spanned by the low frequency eigenspace is large relative that of the mesh.

The second is that we are computing a path embedding of the expander $G_{\mathcal{T}}$ into our image graph G. In this sense each expansion edge can be imagined as a query on the graph and its eigenvectors. As noted by [GM98, LR04] spectral algorithms confuse long paths with small cuts in the graph. Thus each expansion edge probes this very ambiguity. If a small cut exists between the two vertices of the edge, the weight will be small. If instead no cheap cut exists, the weight will be large and the long path in G eliminated.

The expansion augmentation in effect is a lazy version of the near optimal expander embedding algorithm of [KRV06]. In their work [KRV06] iteratively construct an expander for G with weights proportional the flow between the the sites v_i & v_j. They obtain an approximation factor of $O(log^2 n)$ for the "sparsest-cut" and the conductance Φ of the graph (which is worse than the best known of $O(\sqrt{log n})$ [ARV04] but better than the spectral bound). As shown in chapter 2 the conductance Φ and the normalized cut are closely related quantities and so strong approximations for Φ are likely to translate to the same for the normalized cut.

Computational Issues

The expander remedy faces several computational hurdles before becoming practical. The first is that the edge embedding may still involve a large number of Max-Flow computations. As there is regular geometric structure on the underling pixel graph efficient reuse of these computations is possible [BJ06]. In our current experiments the augmentation weighting adds, worst case, a computational cost of $O(|V|^2 \cdot |E|)$ – using a preflow-push algorithm. In the worst case, this is prohibitively expensive for standard image sizes. The second has to do with the sparse matrix representation of $G_{\mathcal{T}}$, as an expander it may have large fill, and is not well-ordered under the above sampling procedure (see [Dem97a] for details on sparse matrix fill). This implies that matrix operations will be slow and memory use high. Improved orderings can be sought for $G_{\mathcal{T}}$ by looking at the vector associated with $\lambda_2(G_{\mathcal{T}})$ prior to weighting. Further, given that most images fall into a small number of known sizes (or can be corrected to), it is feasible to pre-compute a small set of augmented graphs $\{G_{\mathcal{T}}\}$ with optimized fill orders to improve compute times. Then given an input image, select the appropriate expander from the set and perform the weight optimization. In the next section we hint at a complementary approach to addressing the topological bias problem.

3.2.3 Local vs. Global Geometry

The failure of the standard spectral algorithm hinges on the use of the eigenvector rows as global coordinates for the vertices. This follows from the insight that the eigenvectors effect the slowest possible oscillation on the graph. Expansion augmentation mitigates the affect of this decision, but does not "solve" the problem. Thus Algorithm §3.1 acts as if the eigenspace provides an embedding of the graph on a manifold. We know that such metricizations of the graph involve using all the eigenvectors of the Laplacian simultaneously (see Euclidean Commute Times in [Bol02]) or solving semi-definite programs with triangle inequality constraints (*e.g.* [ARV04, GW95]). Unfortunately both of these optimizations are prohibitively expensive for large graphs. As shown in 3.5 the partition boundaries tend to occur on vertices with near zero value in the eigenspace (gray is zero). This yields cuts in the smooth sign functions demonstrating that the global coordinate interpretation of the eigenvectors is dangerous (in general).

If instead we treat the the vertex coordinates as elements of local charts on some unknown embedding of the graph, defined by the topology at each vertex, their utility is apparent. Intuitively we see that slow oscillations result in locally smooth charts, while coherent edge drops result in locally structured charts, and that smooth oscillations around the origin are no different than those at the extreme values. In the next chapter we develop the Spectral Rounding algorithm as a means of using the eigenvectors in such a fashion.

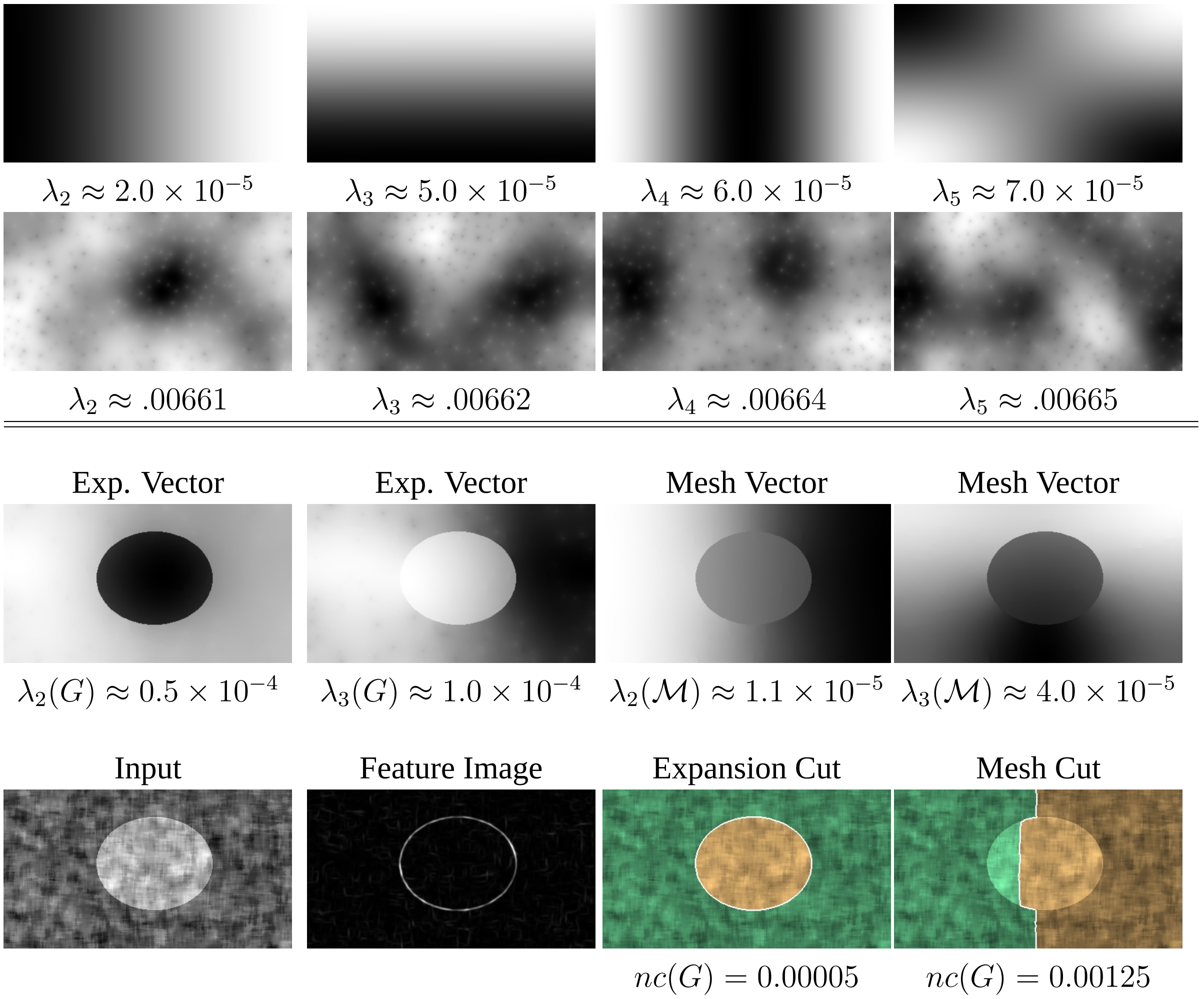

$\lambda_2 \approx 2.0 \times 10^{-5}$
$\lambda_3 \approx 5.0 \times 10^{-5}$
$\lambda_4 \approx 6.0 \times 10^{-5}$
$\lambda_5 \approx 7.0 \times 10^{-5}$
$\lambda_2 \approx .00661$
$\lambda_3 \approx .00662$
$\lambda_4 \approx .00664$
$\lambda_5 \approx .00665$
Exp. Vector
Exp. Vector
Mesh Vector
Mesh Vector
$\lambda_2(G) \approx 0.5 \times 10^{-4}$
$\lambda_3(G) \approx 1.0 \times 10^{-4}$
$\lambda_2(\mathcal{M}) \approx 1.1 \times 10^{-5}$
$\lambda_3(\mathcal{M}) \approx 4.0 \times 10^{-5}$
Input
Feature Image
Expansion Cut
Mesh Cut
$nc(G) = 0.00005$
$nc(G) = 0.00125$

Figure 3.5: The first row contains the first four harmonic eigenvectors of the unweighted mesh 220 by 400. The second row contains the first four harmonic eigenvectors of the unweighted expansion augmented mesh. The normalized eigen-gap, $\frac{\lambda_5-\lambda_2}{\lambda_2}$, for the mesh topology is 3.667 and 0.006 for the expansion augmented mesh topology. Approximately 200 edges were added to $E(G)$ to obtain this ratio (*i.e.* $|E_a| < \sqrt{|E|}$). The third row contains the second (used in partitioning) and third eigenvectors of the augmented mesh and mesh after edge weighting with image data. The forth row contains the input data and obtained cuts. The normalized cut values, $nc(G)$, are reported for the original image weighted mesh. The expansion augmented graph generates a superior cut on the underlying mesh. The intervening contour cue was used to weight the edge based on the data in the Feature Image using the authors' parameters [SM00, YS03a]. The weighted normalized eigen-gap ratios are 2.25 (mesh) and 0.13 (expansion augmented mesh).

Chapter 4

Spectral Rounding & Fractional Averages

The family of procedures, we term **spectral rounding** algorithms obtain discrete solutions to the graph partitioning problem through minimizing a subset of the generalized eigenvalues of graph Laplacians. In essence, spectral rounding reduces to the following, at each iteration a small number of eigenvectors, with small eigenvalue, are computed and used to determine a reweighting w' for the graph $G = (V, E, w)$.

We show the reweighting process induces a k-way multiplicity in the k smallest eigenvalues of L – *i.e.* $\lambda_i(L) = 0$ for $1 \leq i \leq k$. By obtaining a Laplacian with this nullspace property we guarantee that the matrix represents k disconnected subgraphs, whose vertex membership can be read off directly from the first k eigenvectors. In this way a feasible solution to the mathematical programming problem in Chapter 2 §2.3 (2.3.1) can be obtained directly from a spectral method without applying a standard geometric separator (as with the algorithm described in Chapter 3 §3.1).

The relationship between the Laplacian $\mathbf{co-rank}$ and the number of connected components in made concrete in the following lemma.

Lemma 4. *Given a graph $G = (V, E, w)$ with generalized Laplacian L, the number of connected components in G is equal to the $co-rank$ of L.*

Proof. Recall that the quadratic form $f^T L f$ for a vertex valuation $\{f \mid f : V \rightarrow \mathbb{R}\}$ can be written as $f^T L f = \sum_{i<j} w_{ij}(f(i) - f(j))^2$. By observation it is clear that the valuation $(\forall i)\ f(i) = 1$ is in the kernel of L. Now, assume that G has exactly two connected components. Call the two vertex sets associated with these connected components A and B. We may now define two valuations f and g such that $f(x) = 0$ if $x \notin A$ and 1 otherwise, and $g(x') = 0$ if $x' \notin B$ and 1 otherwise. The we see that two valuations f and g are orthogonal as $A \cap B = \emptyset$ by hypothesis. Further both f and g lie in the kernel of L. This follows from observing that no non-zero weight edge exists between the two components of G. The further generalization of this property to k is obvious. □

Corollary 3. *For L such that $k = co - rank(L) > 1$, k vectors may be chosen as a basis for $Ker(L)$ constituting a feasible point for the integer programming problem in §(2.3.1).*

Proof. As vectors are orthogonal and non-negative, they may be normalized to be $D-$orthogonal while still spanning the $Ker(L)$. This satisfies the orthogonality and normalization constraints in program (2.3.1). □

4.1 The `SR-Algorithm`

For a graph $G = (V, E, w^0)$ prescribe the number of partitions k that the edge cut is to yield. Given a valid **reweighting scheme**, iteration of the `SR-Step` produces a sequence of N weightings $\{w^{(N)}\}$ such that the graph $G^N = (V, E, w^N)$ is disconnected into k components by the weighting w^N.

Algorithm 1 `SR-Step(w:` $||w||_k > 0$`)`

Let $F_k = [f_1 \dots f_k]$ denote the k generalized eigenvectors of the matrix pencil $L(G; w), D(G; w)$ associated with the k smallest eigenvalues $\Lambda_k = diag([\lambda_1 \dots \lambda_k])$

```
1. compute $w_r = \mathcal{R}(F_k, \Lambda_k)$, set $\alpha = 1$ & $w' = w_r$
```

```
2. while $||w'||_k \geq ||w||_k$
       $\alpha \leftarrow \frac{1}{2}\alpha$, $w' = (1-\alpha)w + \alpha w_r$

3. return $w'$
```

The function $\mathcal{R}$ computes a new weighting of the graph given the first k eigenpairs of L, D. The norm $|| \cdot ||_k$ is taken over weightings of the graph, such that $||w||_k = 0$ *iff* the weighting w disconnects the graph into at least k pieces. A pair $\mathcal{R}$, $|| \cdot ||_k$ is called a **reweighting scheme** if the `SR-Step` converges in a finite number of iterations. We define Algorithm 2, the `SR-Algorithm`, as the iteration of Algorithm 1 until $||w^{(N)}||_k \approxeq 0$. In the following sections we propose $\mathcal{R}$s and corresponding norms $|| \cdot ||_k$ such that the `SR-Step` and `SR-Algorithm` converge in the desired fashion.

4.1.1 Examples of Reweighting Schemes

For the sake of readability, a collection of **reweighting schemes** are quickly defined for the `SR-Algorithm`. The properties of these update schemes depend upon results in successive sections, but a brief mention of the high-lights for each method is given below. In the following $\Psi_h(\cdot)$ denotes the function that returns the height, on the unit circle, of stereographic projection of the line onto the circle. The reweighting update is given for each edge $(ij) \in E(G)$.

The reweighting schemes are listed in order of simplicity. Each successive scheme can be thought of as augmenting its predecessor. The first two approaches and instances of the third can be shown for converge (formally). The last two are more powerful and consistently generate superior cuts when compared to the standard algorithm. The final reweighting scheme, FAMR, is shown to make progress at each step and converges rapidly in practice. This scheme, with the heuristic at the end of the chapter, was used to generate the results in Chapter 5.

Inverse Reweighting

For a $k-$way cut, the updated weights w_r and progress norm $||\cdot||_k$ of the **inverse reweighting** are defined as:

$$\begin{aligned} w_r(i,j) &= \Psi_h\left(\frac{f_i^2+f_j^2}{(f_i-f_j)^2}\right)\cdot w(i,j) \\ ||\cdot||_k &= \lambda_k \end{aligned}$$

where $Lf_k = \lambda_k Df_k$. In §4.3.1 the `SR-Algorithm`, under **inverse reweighting**, is shown to converge on graphs with $||\cdot||_k < 1$. In the case of a 2-cut this reduces to $\lambda_2(\mathcal{L}) < 1$. The class of graphs satisfying this spectral constraint is very general, excluding an uninteresting collection of graphs for our purposes (known as expander graphs).

Offset Inverse Reweighting

For a $k-$way cut, the updated weights w_r and progress norm $||\cdot||_k$ of the **inverse reweighting** are defined as:

$$\begin{aligned} w_r(i,j) &= \Psi_h\left(\frac{f_i^2+f_j^2+2\alpha^2}{(f_i-f_j)^2}\right)\cdot w(i,j) \\ ||\cdot||_k &= \lambda_k \end{aligned}$$

where $Lf_k = \lambda_k Df_k$. The convergence proof for **inverse reweighting** is adjusted to address this case in §4.3.2. The eigenvalue bound on λ_k is weakened to $\lambda_k \leq .5857$. This bound is many orders of magnitude higher than we observe in image processing and data mining applications.

Mixed-valuation Reweighting

This update constructs an update vector q from k eigenvectors as $q = \sum_{i=1}^{k} \omega_i f_i$ and $\omega \mid q^T D q = 1$ for $1 \leq i \leq k$, $L f_i = \lambda_i D f_i$. The vector q is then used in either an **inverse reweighting** or **offset inverse reweighting** update:

$$\begin{aligned} w_r(i,j) &= \Psi_h \left(\frac{q_i^2 + q_j^2}{(q_i - q_j)^2} \right) \cdot w(i,j) \\ || \cdot ||_k &= \sum_{i=1}^{k} \omega_i^2 \lambda_k . \end{aligned}$$

The properties of this update rule are given in §4.4.1. The convergence results for the previous methods can be directly applied to Mixed-valuation Reweighting, as it is a single vector update.

Fractional Average Multi-valuation Reweighting

For a $k-$way cut, the Fractional Average Multi-valuation Reweighting (FAMR) updated weights w_r and progress norm $|| \cdot ||_k$ are constructed as:

$$\begin{aligned} w_r(i,j) &= \Psi_h \left(\frac{\sum_{l=2}^{k} f_l^2(i) + f_l^2(j)}{\sum_{l=2}^{k} \frac{1}{\lambda_f} (f_l(i) - f_l(j))^2} \right) \cdot w(i,j) \\ || \cdot ||_k &= \frac{\sum_{l=1}^{k} f_l^T L f_l}{\sum_{i=1}^{k} f_l^T D f_l} . \end{aligned}$$

This update rule allows the information used in a suite of eigenvectors to be used to determine a single reweighting that drives down the fractional average of the eigenvalues. The vector f_l is taken to satisfy $L f_l = \lambda_l D f_l$ for the current matrix. At present global convergence results for this reweighting are strictly empirical.

4.2 Fractional Averages: a reweighting function

In this section we introduce the **fractional average**[1] (definition 12) of a set of fractions and demonstrate that the Rayleigh quotient $\frac{f^TLf}{f^TDf}$ is a fractional average. The existence of a reweighting of the fractions, composing a fractional average, that minimizes this average is shown in lemma 5. Such reweightings are connected to the eigenvalues of L, D when the reweighting is derived from an eigenvector in lemma 6. These results are extended to multiple eigenvectors in theorem 6 and lemma 13. The lemmas are then employed to prove that a simplified instance of the `SR-Algorithm` convergences in theorem 5 of §4.3. This basic convergence proof is repeated for more complex example of the `SR-Algorithm`.

By lemma 1 we saw that the Rayleigh quotient could be written as a sum of formal fractions where the numerators are added separately from the denominators. Define a **formal fraction** as a pair of real numbers $\frac{a}{b}$ and its **value** as the real number a/b. We call this average of a set of **formal fractions** the **fractional average**. We now prove a few simple but important facts about fractional averages.

Definition 12. *Given formal fractions*

$$\frac{a_1}{b_1}, \dots, \frac{a_n}{b_n}$$

the **fractional average** *is the formal fraction*

$$\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i}$$

where the a_i's and b_i's are reals.

We will simply call formal fractions fractions and only make a distinction between the formal

[1] The fractional average is also known as the generalized mediant of a set of fractions. While optimizing functions of the form $\frac{w^Ta}{w^Tb}$ falls into the category of *linear fractional programming* [Boy04].

fraction and its value when needed. In the case when the a_i's and b_i's are nonnegative we first observe that the fractional average is a convex combination of the fractions. That is we can rewrite the sum as

$$\sum_{i=1}^{n} \frac{b_i}{\bar{b}} \cdot \frac{a_i}{b_i}$$

where $\bar{b} = \sum_{i=1}^{n} b_i$. Thus fractional average lies between the largest and smallest fraction.

Possibly a more important interpretation is by viewing each fraction $\frac{a_i}{b_i}$ as the point $P_i = (b_i, a_i)$ in the plane and the value of the fraction is just its slope. The fractional average is just the vector sum of the points. Since we are only interested in the value of the fraction, the slope, we will think of the fractional average as the centroid of the points. If we multiply the numerator and denominator by a scalar w we shall say we **reweighted the fraction** by w. Geometrically, we are scaling the vectors or points P_i and then computing the centroid.

In the next lemma we show that we can control the slope of the fractional average by reweighting the formal fractions.

Lemma 5. *If* $\frac{a_1}{b_1} \leq \cdots \leq \frac{a_n}{b_n}$ *and* $w_1 \geq \cdots \geq w_n$ *then*

$$\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \geq \frac{\sum_{i=1}^{n} a_i w_i}{\sum_{i=1}^{n} b_i w_i}$$

The inequality is strict if for some pair $1 \geq i < j \leq n$ *we have that* $\frac{a_i}{b_i} < \frac{a_j}{b_j}$ *and* $w_i > w_j$.

Proof. It will suffice to show that

$$\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} - \frac{\sum_{i=1}^{n} a_i w_i}{\sum_{i=1}^{n} b_i w_i} \geq 0 \tag{4.2.1}$$

Multiplying the left hand side through by its denominators we get

$$\sum_{i,j} a_j b_i w_i - \sum_{i,j} a_j b_i w_j = \sum_{i,j} a_j b_i w_i - a_j b_i w_j \tag{4.2.2}$$

Observe that term where $i = j$ are zero. Thus we can write the sum as:

$$\sum_{i<j} a_j b_i (w_i - w_j) + a_i b_j (w_j - w_i) \tag{4.2.3}$$

Rearranging the last term in the sum gives:

$$\sum_{i<j} a_j b_i (w_i - w_j) - a_i b_j (w_i - w_j) \tag{4.2.4}$$

Finally we get:

$$\sum_{i<j} (a_j b_i - a_i b_j)(w_j - w_i) \tag{4.2.5}$$

By the hypothesis each term in the sum above is nonnegative which proves the inequality. The strict inequality follows when one of the pair of terms in the sum are both positive as prescribed in the hypothesis. □

We get an exact expression by observing that the only time we got an inequality was when we cleared the denominators. Thus we have the following equation.

$$\sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i w_i - \sum_{i=1}^{n} a_i w_i \sum_{i=1}^{n} b_i = \sum_{i<j} (a_j b_i - a_i b_j)(w_j - w_i) \tag{4.2.6}$$

To determine a reweighting, recall lemma 1, which shows that the Rayleigh quotient $\frac{f^T L f}{f^T D f}$ may be rewritten as the fractional average over the set of formal fractions given for each edge in G. One of the simplest ways to get weights satisfying the hypothesis of lemma 5, for such a system, is to pick $w_i = \frac{b_i}{a_i} = \frac{f(u_i)^2 + f(v_i)^2}{(f(u_i) - f(v_i))^2}$, if a_i is not zero. We shall call this **inverse reweighting**. This reweighting scheme gives very large values for small values of a_i. We have found that using the stereographic map to normalized the inverse fractions between zero and one works well.

Observation 1. *The stereographic projection $\Psi : \mathbb{R}^d \to \mathcal{S}^d$ preserves the order of points on the real line, mapping points at ∞ to 1 and points at 0 to 0. Thus the* **inverse weight ordering** *of the edge update values is preserved by the stereographic map.*

If we think of the Ψ as mapping points in $\mathbb{R}^d$ to $\mathbb{R}^{d+1}$, where we are only interested in the value in the $d+1$ dimension, then the images of $v \in \mathbb{R}^d$ is $\frac{v^T v}{v^T v + h} \geq 0$, $h > 0$. We use Ψ_h to denote the map which returns the value in this dimension (*i.e.* the "height" on the sphere). As the Rayleigh quotient $\frac{f^T L f}{f^T D f}$ is scale invariant the reweighting is only sensitive to the rank ordering of the edges which is preserved under the stereographic projection.

Corollary 4. *The fractional average of a set Rayleigh quotients $\left\{ \frac{f_i^T L f_i}{f_i^T D f_i} \right\}$ may be minimized by a single reweighting L'.*

Proof. The proof follows by noting that $\lambda_i = \frac{a_i}{b_i} = \frac{f_i^T L f_i}{f_i^T D f_i}$. Thus lemma 5 may be applied to these fractions to decrease the fractional average. Thus the fractional average of fractional averages may also be decreased. □

4.2.1 From Rayleigh Quotients to Eigenvalues

Above in §4.2 we showed how, given a valuation or set of valuations of a graph, to reweight the edges so as to reduce the Rayleigh quotient. In general this need not guarantee that if the valuation f is an eigenvector with eigenvalue λ of the old graph then the corresponding eigenpair f' and λ' of the new graph will have the property that $\lambda' \leq \lambda$.

Given a new edge weighting w' such that the fractional average of an eigenvector is decreased, we show that there is a linear combination of the weights of the form $w + t \cdot w'$ for $t > 0$ such that the associated eigenvalue is also decreased. This yields an algorithm which forces the target eigenvalue to zero. And motivates a matrix where the entries are linear functions of t and the following lemma.

Definition 13. *Given two weightings w and w' of G we define the* **matrix curve**, *a $1-$dimensional family of matrices, as:*

$$W(t) = W + tW'$$

for $t \geq 0$.

As a direct consequence of the scale invariance of the Rayleigh quotient $\frac{f^T Lf}{f^T Df}$, any linear combination $W(t) = W + tW'$ may be expressed as a convex combination $W(\alpha) = (1-\alpha)W + \alpha W'$ on $0 \leq \alpha \leq 1$ (*i.e.* $\alpha = \frac{t}{t+1}$). The eigenstructure of normalized Laplacians defined on $W(\alpha)$ and $W(t)$ are identical by the scale invariance of the Rayleigh quotient.

Lemma 6. *Given a weighted graph $G = (V, E, w)$, matrices L and D, the simple eigenpair $(f, \lambda) \mid Lf = \lambda Df$, and a new weighting w' such that $\frac{f^T L' f}{f^T D' f} < \frac{f^T Lf}{fDf} = \lambda$ then the derivative of the eigenvalue function*[2] *$\lambda(t)$ on the matrix curve $W(t) = W + tW'$ is well defined for small t and*

$$\frac{d\lambda(t)}{dt} < 0$$

at $t = 0$.

Proof. For a simple eigenpair (f, λ), recall that $\frac{f^T Lf}{f^T Df} = \lambda$, as $W(0) = W$ and thus $L(0) =$

[2] The proof that $d\lambda$ exists follows from properties of the characteristic polynomial of L, D and relies on the implicit function theorem. Details can be found on the differentiability of λ and f in Chapter 2, or in Lax [Lax97] chapter 9.

$L,\ D(0) = D$ by definition. We deduce the bound on $\frac{d\lambda}{dt}$ with a simple calculation.

$$
\begin{aligned}
\frac{d\lambda(t)}{dt} &= \frac{f^T\left(\frac{dL_{(t)}}{dt} - \lambda\frac{dD_{(t)}}{dt}\right)f}{f^T D f^T} && (4.2.7)\\
&= \frac{f^T(L' - \lambda D')f}{fDf^T} \quad \left\{\frac{dL(t)}{dt} = L'\right\} && (4.2.8)\\
&= f^T(L' - \lambda D')f && (4.2.9)\\
&= f^T L' f - \lambda f^T D' f && (4.2.10)\\
&= \left(\frac{f^T L' f^T}{f^T D' f} - \lambda\frac{f^T D' f}{f^T D' f}\right) f^T D' f && (4.2.11)\\
&= \left(\frac{f^T L' f}{f^T D' f} - \lambda\right) f^T D' f && (4.2.12)\\
&= \left(\frac{f^T L' f}{f^T D' f} - \frac{f^T L f}{f^T D f}\right) f^T D' f && (4.2.13)
\end{aligned}
$$

The bound is entailed by observing that 1) the term $f^T D' f = \sum_{v \in V} f^2(v) d'(v)$ is positive, and 2) thus the sign of $\frac{d\lambda(t)}{dt}$ is strictly negative due to the inequality $\frac{f^T L' f}{f^T D' f} < \frac{f^T L f}{fDf}$. The above derivation demonstrates that the slope $\frac{d\lambda(t)}{dt}$ is negative at $t = 0$. By the continuity of λ there exists a $t > 0$ such that $\lambda(t) < \lambda(0)$. This insures that the eigenvalue λ associated with f can be decreased algebraically using the procedure in §4.2 on fractional averages. □

To handle multiple eigenvectors one hope might be to simultaneously bound the derivatives of the target eigenvalues $\{\lambda_2, ..., \lambda_k\}$ of $L(t), D(t)$. To do this one arrives at the update criteria for the re-weighting w'; select a w' such that $\frac{f_i^T L' f_i}{f_i^T D' f_i} < \frac{f_i^T L f_i}{f_i^T D f_i}$ insuring that $\lambda_i(L(t)) < \lambda_i(L)$ for $2 \le i \le k$. But in general this may be unrealistic since we must use the same weight per edge for all the eigenvectors f_i. In the case where the aforementioned inequalities do not hold, we determine w' so as to decrease the fractional average of the Rayleigh quotients. The average of the target eigenvalues tends to zero as a consequence of the decrease in the fractional average.

4.3 Convergence and Termination of Reweighting Schemes

In this section we introduce two simplified reweighting schemes and demonstrate that they converge to disconnected graphs. The target eigenvalue is taken to be the k^{th} smallest eigenvalue of the matrix. Recall that when $\lambda_k = 0$ the weighted graph has k connected components. The first scheme is the **inverse reweighting** for a single vector, associated with the target eigenvalue. The iteration is shown to terminate when the eigenvalue reaches zero thus cutting the graph into k pieces. The second scheme **offset inverse reweighting** combines two valuations in a single update, the constant vector $\mathbf{1} \mid L\mathbf{1} = 0$ and the target eigenvector $f \mid Lf = \lambda_k Df$.

4.3.1 Inverse Reweighting

We begin by demonstrating convergence for the simplest update rule that which applies **inverse reweighting** directly. Under such a scheme the pair $\mathcal{R}$ and $|| \cdot ||_k$ are specified as follows:

$$w_r(i,j) = \Phi_h \left(\frac{f_i^2 + f_j^2}{(f_i - f_j)^2} \right) \cdot w(i,j) \tag{4.3.1}$$

$$|| \cdot ||_k = \sum_{i=1}^{k} \lambda_k \tag{4.3.2}$$

in this section the **inverse reweighting** scheme is shown to converge for a broad class of graphs. In subsequent sections we often refer to functions of the fraction associate with each edge $\frac{(f_i - f_j)^2}{f_i^2 + f_j^2}$ as a δ_{ij} update associated with the edge $(ij) \in E(G)$.

Termination and Convergence

In order to prove that the `SR-Algorithm` converges to a k-way partition we first need to show that each step of `SR-step` terminates. Then we use this termination to show convergence. To simplify the discussion we only consider one eigenvector. The `SR-Algorithm` has two majors steps. In the first step given a valuation f it computes a reweighting w_r. We claim with a reasonable assumption about the Rayleigh quotient that

$$\frac{f^T L f}{f^T D f} > \frac{f^T L_r f}{f^T D_r f}. \tag{4.3.3}$$

at each step. Thus by lemma 5 we know that equation 4.3.3 is true as long as not all the fractions $\delta_{ij} = (f_i - f_j)^2/(f_i^2 + f_j^2)$ are equal. We show that if the fractions are all equal the Rayleigh quotient is at least one, contradicting how we construct affinity graphs.

Lemma 7. *Let $G = (V, E, w)$ be a connected graph and f a valuation such that all the fractions $\delta_{ij} = (f_i - f_j)^2/(f_i^2 + f_j^2)$ are equal then $\frac{f^T L f}{f^T D f} \geq 1$*

Proof. Let f be a valuation of G with Rayleigh quotient λ. Suppose all $\delta_{ij} = (f_i - f_j)^2/(f_i^2 + f_j^2)$ over all edges of G are the same value. Observe that: 1) $\delta_{ij} > 1$ if the sign of f_i and f_j differ, 2) $\delta_{ij} = 1$ if $f_i f_j = 0$, 3) $\delta_{ij} < 1$ if $f_i f_j > 0$. If we are not case 3) then the value of each fraction is at least one. Thus by lemma 1 the Rayleigh quotient is at least one. □

Lemma 8. *Algorithm SR-step terminates for graphs such that $\lambda_k < 1$*

Proof. Follows from lemma 6 and Lemma 7. Observe that eigenvectors of L, D satisfy $f^T D\mathbf{1} = 0$, and thus f changes sign over at least one edge in G. This eliminates case *3* in lemma 7. □

By lemma 8 we know that each step of procedure SR-step produces a new weighted graph such that the norm $||\ ||_k$ is strictly decreasing. We show that in fact the limit norm must be zero, *i.e.* the `SR-Algorithm` converges.

Again for simplicity of the exposition we only consider the case of a single eigenvector. Let G^i, D^i, λ^i, and f^i be the ith graph, its degree matrix, its second eigenvalue, and eigenvector. That is $L^i f^i = \lambda^i D^i f^i$. We also assume that each f^i is a unit length vector. Thus all the eigenvectors all belong to a compact region in $\mathbb{R}^n$, the unit sphere. The Bolzano-Weierstrass theorem dictates such a space contains an accumulation point, say, f. Let $\{f^{(i)}\}$ be a subsequence of eigenvectors that converge to f, and let G',L', D', and λ' be the corresponding graph, Laplacian, degree matrix and eigenvalue. The eigenvectors converge to f and the eigenvalues converge to some value λ. To insure that the graph also converges to a unique graph we apply the Bolzano-Weierstrass theorem again to the weighted graphs which belong to a compact set in $\mathbb{R}^m$. Thus, we may also assume the graphs and degree matrix converge to G', L', D', and λ' such that $L'f' = \lambda' D' f'$.

Theorem 5. *The limit of the sequence of $\{\lambda^{(i)}\}$, as defined above, converges to $\lambda = 0$*

Proof. Suppose that $\lambda' > 0$. We know by lemma 8 that if we run `SR-step` on G' we will get a new graph with $\lambda'' < \lambda'$. Let $\epsilon = \lambda' - \lambda''$. We can also run `SR-Step` on each $G^{i\prime}$ getting a new eigenvalue $\lambda^{i\prime\prime}$. Let $\epsilon^i = \lambda^{i\prime} - \lambda^{i\prime\prime}$. Since `SR-step` is a continuous function in $G^{i\prime}$ and $f^{i\prime}$ we get that the ϵ^i converge to ϵ. For sufficiently large i it follows that $\epsilon^i \geq \epsilon/2$. But this implies that $\lambda^{i\prime}$ goes to $-\infty$ which contradicts the fact the they a bounded below by zero. □

Random Walks and Convergence

The eigenvalue bound entailed by lemma 8 has a natural interpretation in terms of random walks on weighted graphs. Given that $\lambda_2(\mathcal{L}) = 1$ is the largest value a less than complete graph can take, this is a very small set of weighted graphs. Recall the Cheeger bound on the normalized cut from Chapter 2, $\lambda_2(\mathcal{L}) \leq 2\Phi(G)$. In table 2.1 (Chapter 2) the eigenvalues of the normalized Laplacian $\mathcal{L}$ are shown to be reflected and shifted eigenvalues of the transition matrix $T = D^{-1}A$. For non-bipartite graphs, the mixing time on T is a proportional to $\lambda_{n-1}(T) = 1 - \lambda_2(\mathcal{L})$. So as $\lambda_2(\mathcal{L})$ approaches 1 the mixing time tends to zero. In terms of a cut, we see that there exists an optimal subset of vertices V_* associated with the conductance number $\Phi(G)$ and thus

$$\begin{aligned} \lambda_2(\mathcal{L}) &\leq 2\Phi(G) && (4.3.4)\\ 1 &\leq 2\frac{\delta V_*}{vol(V_*)} && (4.3.5)\\ vol(V_*) &\leq 2\delta V_* && (4.3.6)\\ \delta V_* + \iota V_* &\leq 2\delta V_* && (4.3.7)\\ \iota V_* &\leq \delta V_* && (4.3.8) \end{aligned}$$

where ιV_* denotes the volume of the internal edges of V_*. In terms of a random walker, this inequality indicates that every subset of V with less than half of the transition probability mass has an emittence probability of at least $\frac{1}{2}$. Therefore a random walker leaves any such subset roughly half the time at each step. Graphs with large λ_2 values are termed expander graphs and pop-up in the study of communication networks, rapidly mixing times of markov chains, as well as a host of other applications.

4.3.2 Offset Inverse Reweighting

We now state an alternative δ_{ij} update rule and prove convergence and termination of the resulting reweighting scheme. Recall that if we wish to break the graph into k pieces we must drive the k^{th} eigenvalue to zero. We begin by fixing the following **reweighting scheme**, define $f \mid Lf = \lambda_k Df$ where λ_k is the k^{th} smallest generalized, let

$$\begin{aligned} w_r(i,j) &= \Phi_h\left(\frac{f_i^2 + f_j^2 + 2\alpha^2}{(f_i - f_j)^2}\right) \cdot w(i,j) && (4.3.9)\\ ||\cdot||_k &= \lambda_k. && (4.3.10) \end{aligned}$$

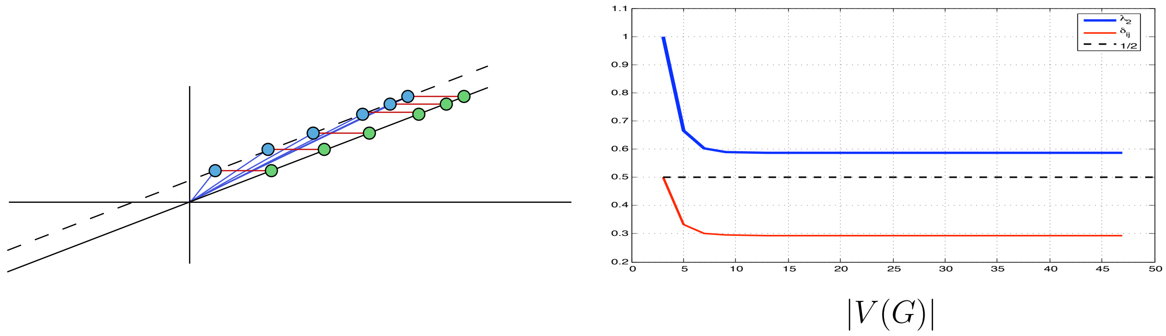


Figure 4.1: Left: an illustration of shifted formal fractions producing a constant slope (update value). The offset $2\alpha^2$ is given in red and maps the slopes corresponding to the blue points to the green co-linear points (*i.e.* the green points produce a constant update fraction δ_{ij} for all edges). Right: the eigenvalue $\lambda_2(\mathcal{L})$ with respect to the length of **iso-δ valuation** weighted line. As $|V(G)| \to \infty$ the eigenvalue is bound as $\lambda_2 > .5857$.

The term α is taken to be $1/\sqrt{vol(G)}$, which can thought of as an entry in the normalized $\mathbf{1}$s vector (associated with $\lambda = 0$). We now show that this scheme converges, in the desired sense, on a large set of graphs (in essence we must exclude a larger class of expanders than in §4.3.1).

At a high level it's worth motivating the use of (4.3.9) over (4.3.1). In particular what deficit in (4.3.1) does it correct? It is clear that reweighting according to (4.3.1) will most aggressively down-weight edges that span 0 in the valuation f (*i.e.* $f_i f_j < 0$). Immediately a number of graphs come to mind where this property is undesirable. For example, a symmetrically weighted odd-length path graph with a heavy-weight edge at the center will yield a suboptimal 2-cut under (4.3.1) but not (4.3.9). This is because λ_2 eigenvector of the path graph crosses zero at the center edge for such a weighting. The update rule in (4.3.9) biases toward cutting edges with large gap $(f_i - f_j)^2$ and high magnitude $f_i^2 + f_j^2$.

The following lemmas and definitions will be combined to supplant lemma 7, and allow the remaining technology in the previous section to be employed to demonstrate convergence. Their purpose is to establish the greatest lower bound of $\lambda_k = 0$ on the `SR-Algorithm`. To do so we establish that the update rule only admits constant offset fractions, over all δ_{ij} as show in figure 4.1, for connected graphs with large eigenvalue $\lambda > .5857$. We will call functions of the form $f : V(G) \rightarrow \mathbb{R}$, **vertex valuations** and those that produce a constant update δ_{ij} at each edge **iso**$-\delta$ **valuations** of the graph. These valuations are exactly the fixed points of the spectral rounding algorithm.

Corollary 5. *The Rayleigh quotient* $\frac{f^TLf}{f^TDf}$ *is invariant to a uniform rescaling of the weighting.*

Proof. A direct consequence of the lemma 1 in Chapter 2 □

With the above corollary, we may choose a global scale for w such that $\sum_{i=1}^{n} d_i = 2$, and by a simple calculation verify that $2\alpha^2 = 1$, simplifying the reweighting scheme. The second degree of freedom that we fix in (4.3.9) is the norm of $f \mid f^TDf = 1$. While this has no effect on the Rayleigh quotient $\frac{f^TLf}{f^TDf}$, this scaling has a dramatic effect on (4.3.9).

Having fixed α and the scale of f it is possible to derive an **iso**$-\delta$ **valuation** $f : V \rightarrow \mathbb{R}$, such as that shown in Figure 4.1, for a weighted path graph. Given such an f, a weighting of the path graph G may be constructed satisfying $L(G)f = \lambda_2 D(G)f$. We will show that the weights drop off exponentially as one walks from the center to end of the line, dictating that the weighted path is an expander graph. Further, from the properties of f we can bound the eigenvalue λ_2 for paths or arbitrary length.

Constructing an iso$-\delta$ valuation for the path graph

Claim 1. *Given the update equation* δ_{ij} *with* $2\alpha^2 = 1$ *a vertex valuation* f *can be constructed such that the fraction* δ_{ij} *is constant for all edges in the graph.*

For the weighted path graph on n vertices it is possible to construct an **iso**$-\delta$ **valuation** f insuring that for all edges (i, j) the update $\delta_{ij} = c$. For simplicity we assume that n is odd and number the vertices, over the integers $-\lfloor \frac{n}{2} \rfloor$ to $\lfloor \frac{n}{2} \rfloor$, as the path is odd length $f_0 = 0$. Given f_0 there must be an offset ϵ so that $f_1 = f_0 + \epsilon$. Thus we can write down the first update value, to the right, as

$$\epsilon = \pm \frac{\sqrt{2\alpha^2}\sqrt{\delta}}{\sqrt{1-\delta}} \tag{4.3.11}$$

or more generally

$$\delta_{i,i+1} = \frac{(f_i - f_{i+1})^2}{f_i^2 + f_{i+1}^2 + 1} = \frac{\epsilon_{i+1}^2}{f_i^2 + (f_i + \epsilon_{i+1})^2 + 1} \tag{4.3.12}$$

the valuation f_{i+1} can be written as

$$f_{i+1} = f_i + \epsilon_{i+1} = f_i \pm \frac{\delta f_i + \sqrt{\delta\,(1 + 2f_i^2 - (1 + f_i^2)\delta)}}{|\delta - 1|} \tag{4.3.13}$$

an **iso**$-\delta$ **valuation** can be constructed by applying this recurrence, starting at the center of the line and moving toward the end points. We note that the **iso**$-\delta$ **valuation** grows exponentially as the above recurrence is strictly greater than $f_i > \epsilon_1(i + 2\epsilon_1^2)(1 + 2\epsilon_1^2)^{i-1}$, for $i > 0$, which grows exponentially with the length of the path.

Claim 2. *A weighting of the path graph, with Laplacian L and degree matrix D, can be derived from a vector f such that $f \mid Lf = \lambda_2 Df$*

Again, for simplicity assume the path is odd length. The weighting can be computed directly from f as the equation $Lf = \lambda Df$ fully constrains the weighting of the line. The valuation f is minimal, over the set $\{f \mid f^T D\mathbf{1} = 0\}$, as it is the unique vector that breaks the path into exactly two pieces. Let $w_1 = 1$ denote the edge linking the center vertex v_0 and it's neighbor f_1. Starting at the center we can work out the form of w_2 as $w_2 = w_1 \cdot \frac{f_1(1-\lambda) - f_0}{f_2 - f_1 + \lambda f_2}$, which can be generalized as the recurrence

$$w_i = w_{i-1} \cdot \frac{(1-\lambda)f_i - f_{i-1}}{f_{i+1} - (1-\lambda)f_i} \tag{4.3.14}$$

as the term $\frac{(1-\lambda)f_i - f_{i-1}}{f_{i+1} - (1-\lambda)f_i}$ is positive, it is less than 1, and thus the weights exponentially decay with the length of the line.

Proposition 1. *An* **iso** $\delta-$**valuation** *weighted line graph has an eigenvalue* $\lambda_2 > .5857$*, independent of length.*

The above proposition was determined from a numerical optimization (see table 4.1 for an illustration of the asymptote). The bound hinges on the relationship between two free parameters 1) $vol(G) = 2$ and 2) $f^T Df = 1$. Given a fixed graph volume and vector normalization a suitable slope δ_{ij} must be determined such that all edge weights are strictly positive and the valuation f has proper norm. Under these assumptions, for paths $\mathcal{P}_n$ where $n > 7$, $\delta > .2928$ to produce a valid weighting and thus the $\lambda_2 > .5857$ as $vol(G) = \frac{\delta_{ij}}{\lambda}$.

From paths to general graphs

We now define the relationship between **iso-**δ **valuations** of general graphs and such valuations of the path. This following statements relate a graph which admits an **iso-**δ **valuation** to the path graph and equates the eigenvalues of the former with latter.

Lemma 9. *Given an* **iso**$-\delta$ **valuation** f *for a connected graph* G *the following properties hold*

I . for all vertices $(u, v) \in E(G)$ *where* $f(u) > f(v)$ *then* $f(u) = f(v) + \epsilon(f(v))$*,*

II . for two vertices v *and* u *where* $(u, v) \in E(G)$ *then* $f(u) \neq f(v)$*,*

III . graphs with cliques of size 3 *or greater do not admit iso*$-\delta$ *valuations.*

Proof. We prove the properties for the $i^{th} + 1$ term, which is assumed to satisfy $f_{i+1} \geq f_i$ *w.l.o.g.* Property *I* follows from the observation that equation 4.3.12 admits a unique positive solution for ϵ_{i+1} in terms of the known quantities f_i and δ. This property dictates that every edge is between vertices spanning exactly one threshold in f. Property *II* is simple to observe, as all the fractions would equal 0 and as G the orthogonality conditions $fD\mathbf{1} = 0$ prevents this. Property *III* depends on *I* and *II*. By P*II* we have that each valuation in the clique must be unique. By P*I* we establish a contradiction. Label the vertices of clique as a, b, and c using P*I* and P*II* we may assume that $f(a) < f(b) < f(c)$. The unique monotonicity of the edge equations contracts this ordering relationship, demonstrating *III* for the minimal case. □

The next lemma shows that an **iso-δ valuation** of a graph G provides a mechanism by which the graph can be collapsed onto the path graph without loss of generality. Further, that the eigenvalue of the line is a lowerbound on the eigenvalue of the graph.

Lemma 10. *Given an iso$-\delta$ valuation f of a graph G such that $Lf = \lambda Df$ then a line graph G' may be constructed such that $L'f' = \lambda D'f'$.*

Proof. We begin by demonstrating that the thesis holds for a contraction of two vertices with the same value in f. *w.l.o.g.* assume that these vertices are associated with the least valued vertices of f and order the graph accordingly, generality holds as this is accomplished by permuting the associated matrices L and D. We expand $Lf = \lambda Df$ to clarify our assumptions

$$\begin{pmatrix} d_1 & 0 & -w_{1,i} & . & . & . \\ 0 & d_2 & -w_{2,j} & . & . & . \\ . & & . & & & \\ . & & & . & & \\ . & & & & . & \\ . & & & & & \end{pmatrix} \begin{pmatrix} a \\ a \\ b_1 \\ . \\ . \\ . \end{pmatrix} = \lambda \begin{pmatrix} d_1 \cdot a \\ d_2 \cdot a \\ d_3 \cdot b_1 \\ . \\ . \\ . \end{pmatrix} . \qquad (4.3.15)$$

For the first vertex, rewrite the constraint equation $Lf = \lambda Df$ as

$$\begin{aligned} d_1 \cdot a - \lambda d_1 - \sum_{v_i \sim v_1} w_{1i} b_i &= 0 && (4.3.16) \\ (a - \lambda a) \sum_{v_i \sim v_1} w_{1i} - \sum_{v_i \sim v_1} w_{1i} b_i &= 0 \text{ (def. of } d_1\text{)} && (4.3.17) \\ (a - \lambda a) \sum_{v_i \sim v_1} w_{1i} - b \sum_{v_i \sim v_1} w_{1i} &= 0 \text{ (iso}-\delta \text{ hyp.)} && (4.3.18) \\ (a - \lambda a - b) \sum_{v_i \sim v_1} w_{1i} &= 0 && (4.3.19) \end{aligned}$$

analogously the linear constraint for v_2 can be written as $(a - \lambda a - b) \sum_{v_j \sim v_2} w_{2j} = 0$. The replacement of the b_is with b is a direct consequence of *P I* in Lemma 9. Thus we see that the constraint applies only to the weighted degrees, *i.e.* $\sum_{v_i \sim v_1} w_{1i} = \sum_{v_j \sim v_2} w_{2j}$. Define L' and D' as the graph with v_1 and v_2 contracted to v_1', removing double edges. The contraction reduces to adding the first two rows and dividing by two, which preserves the total weight constraint and thus the eigenvector equation $L'f' = \lambda D'f'$ preserves λ.

We now handle the next contraction, along the order induced by f. The only novel term introduced is the backward edge to the previously contracted layer. We drop the $'$ notation to simplify the expressions. We have

$$(b - \lambda b - c) \sum_{v_i \sim v_2} w_{2i} - a \cdot w_{21} = (b - \lambda b - c) \sum_{v_j \sim v_3} w_{3j} - a \cdot w_{31} \qquad (4.3.20)$$

as the weights w_{21} and w_{31} may be taken to be equal the lower diagonal terms cancel and the previous argument applies. This operation is repeated along the ordering induced by f, producing a line graph with matrices L' and D' such that if $Lf = \lambda Df$ then $L'f' = \lambda D'f'$. □

Tying it all together

The results in this section for **offset inverse reweighting** demonstrate that graphs with diameter greater than 7 and eigenvalues such that $\lambda_k < .5857$ must converge to a $k-$way cut under the `SR-Algorithm`. Further lemma 9 demonstrates that *connected* graphs containing the K_p minor with $p > 2$ have no fixed points under the reweighting scheme. Thus we may use the lemmas in this section of supplant lemma 5 and apply the remaining technology in §4.3.1 to prove convergence without modification.

4.4 Multiple Eigenvectors

Combining multiple eigenvectors into a single update is appealing as one can imagine that lower frequency data is important in regularizing the cuts. In lemma 13 we proved that the fractional average of fractional averages drives the sum of the target eigenvalues down. This update rule can be applied to reweight the graph. In §4.4.3 we detail a heuristic for choosing a subset of eigenvectors with which to reweight the graph, based on the differential and structural properties.

4.4.1 Mixed-valuation Reweighting

Mixed-valuation Reweighting updates (MVR) constructs an update vector q where $q = \sum_{i=1}^{k} \omega_i f_i$ and $||\omega|| = 1$. The vector q is then used in either an **inverse reweighting** or **offset inverse reweighting** update:

$$\begin{aligned} w_r(i,j) &= \Psi_h\left(\frac{q_i^2 + q_j^2}{(q_i - q_j)^2}\right) \cdot w(i,j) \\ ||\cdot||_k &= \sum_{i=1}^{k} \omega_i^2 \lambda_k. \end{aligned}$$

Given a valuation q such that $\frac{q^T Lq}{q^T Dq} \leq \frac{1}{k}\sum_{i=1}^{k}\lambda_k$ – *i.e.* $q = \sum_{i=1}^{n}\omega_i f_i$ and $||\omega|| = 1$. We can obtain a reweighting L' using the standard rules for fractional averages such that

$$\frac{q^T L' q}{q^T D' q} < \frac{q^T Lq}{q^T Dq}. \tag{4.4.1}$$

The existence of a such a reweighting following from lemma 5. As in previous sections (see lemma 6) we define a matrix curve $A(t) = A + t \cdot A'$ and connect its eigenvalues to those of $L(0) = D - A, D(0) = D$. We now show that an $\omega-$weighted average of the eigenvalues must decrease for sufficiently small t, bounding their derivative by 0.

Theorem 6. *Given a weighted graph $G = (V, E, w)$ matrices L and D, a set of $k-$eigenpairs $Lf_i = \lambda_i D f_i$ such that $\lambda_i \neq \lambda_j$, and a new reweighting w' such that $\frac{q^T L' q}{q^T D' q} < \frac{q^T Lq}{q^T Dq} = \sum_{i=1}^{k}\omega_i^2\lambda_i$ where $q = \sum_{i=1}^{k}\omega_i f_i$ and $||\omega|| = 1$. Then there exists a mixture ω, such that the derivative of $\sum_{i=1}^{k}\omega_i^2\lambda_i$ along the matrix curve $W(t) = W + tW'$ is bound as*

$$\sum_{i=1}^{k}\omega_i^2\frac{d\lambda_i(t)}{dt} < 0$$

for small t.

Proof. To begin we note that,

$$\frac{q^T(t)L(t)q(t)}{q^T(t)D(t)q(t)} = \sum_{i=1}^{n}\omega_i^2\frac{f_i^T(t)L(t)f_i(t)}{\omega^T\omega} \tag{4.4.2}$$

$$= \sum_{i=1}^{k}\omega_i^2\lambda_i(t) \tag{4.4.3}$$

where we assume that $f_i^T(t)D(t)f_i(t) = 1$ *w.l.o.g.* – the above reduction directly follows from lemma 11. Given 4.4.3 we may apply the derivative operator and obtain

$$\frac{d}{d\,t}\left[\frac{q^T(t)L(t)q(t)}{q^T(t)D(t)q(t)}\right] = \sum_{i=1}^{k}\omega_i^2\frac{d\,\lambda_i(t)}{d\,t} \tag{4.4.4}$$

$$= \sum_{i=1}^{k}\omega_i^2\left(\frac{f_i^TL'f_i}{f_i^TD'f_i} - \frac{f_i^TLf_i}{f_i^TDf_i}\right)f_i^TD'f_i \tag{4.4.5}$$

$$= \sum_{i=1}^{k}\left(\omega_i^2\frac{f_i^TL'f_i}{f_i^TD'f_i} - \frac{q^TLq}{q^TDq}\right)f_i^TD'f_i\,. \tag{4.4.6}$$

The bound follows as there must exist an ω such that $\sum_{i=1}^k \omega_i^2\frac{f_i^TL'f_i}{f_i^TD'f_i} < \frac{q^TLq}{q^TDq}$, given that $\frac{q^TL'q}{q^TD'q} < \frac{q^TLq}{q^TDq}$. For example, any of the indicator vectors $\omega = e_i$ will suffice by lemma 6 as will perturbations about those vectors by continuity. Thus we may take the average difference in 4.4.6 to be negative and the bound obtains. As all the eigenvalues are continuous along the matrix curve (by theorem 7) there must exist an t such that $\sum_{i=1}^k \omega_i^2\lambda_i(t) < \sum_{i=1}^k \omega_i^2\lambda_i(0)$. □

Lemma 11. *If $q = \sum_{i=1}^k \omega_i f_i$, where all f_i satisfy $Lf_i = \lambda_i Df_i$, $D^{-1}L$ has simple structure and $||\omega|| = 1$, then*

$$\frac{q^TLq}{q^TDq} = \sum_{i=1}^{k}\omega_i^2\lambda_i \tag{4.4.7}$$

Proof. From lemma 12 (4.4.7) simplifies to $\frac{\sum_{i=1}^k \omega_i^2 f_i^TLf_i}{\sum_{i=1}^k \omega_i^2 f_i^TDf_i}$ as the basis of q is both $L-$ and $D-$orthogonal. We may choose $f_i^TDf_i = 1$ for all i w.l.og. simplifying (4.4.7) to $\sum_{i=1}^k \omega_i^2 f_i^TLf_i = \sum_{i=1}^k \omega_i^2\lambda_i$ verifying the hypothesis. □

Lemma 12. *Given vectors f and g, satisfying $Lf = \lambda Df$ and $Lg = \lambda' Dg$ where $\lambda \neq \lambda'$, then*

$$g^TLf = f^TLg = 0,\; g^TDf = f^TDg = 0.$$

Proof. The conditions $f^TLg = g^TLf$ and $f^TDg = g^TDf$ follow from the symmetry of L and

D. Their $L-$orthogonality and $D-$orthogonality is shown in the following calculation

$$f^T Lg = \lambda' f^T Dg, \qquad g^T Lf = \lambda g^T Df \quad \text{(by hyp.)} \tag{4.4.8}$$

$$\lambda f^T Dg = \lambda' g^T Df \quad \text{(by sym of } L\text{)} \tag{4.4.9}$$

$$\lambda f^T Dg = \lambda' f^T Dg \quad \text{(by sym of } D\text{)} \tag{4.4.10}$$

For the final equality to hold $f^T Dg = g^T Df = 0$ by hypothesis as $\lambda \neq \lambda'$. Given that $f^T Dg = 0$ this implies that $f^T Lg = 0$ as $f^T Lg = \lambda' f^T Dg = 0$. □

We include "Theorem 12" from [MN99] on the derivatives of multiple eigenvalues. The result is attributed to Lancaster, and pertains to the eigenvalues of matrix curves $A(t) \in \mathbb{R}^{n \times n}$. Three assumptions are made on the curve $A(t)$: 1) the elements of $A(t)$ are analytic functions of in some neighborhood of $t = 0$, 2) the matrix $A(t)_{t=0}$ has **simple structure**[3], and 3) if $\lambda(t)$ is an eigenvalue of $A(t)$ then $\underset{t \to 0}{limit}\, \lambda(t) = \lambda(0)$

Theorem 7. *If $\frac{d^q A(t)}{dt}$ is the first non-vanishing derivative of $A(t)$, then the n eigenvalues $\lambda(t)$ of $A(t)$ are differentiable at least q times at $t = 0$ and their first $q - 1$ derivatives all vanish at $t = 0$.*

See [MN99] for a proof of Theorem 7.

Convergence of Mixture Valuation Schemes

Assuming the generalized Rayleigh quotient $\frac{q^T Lq}{q^T Dq}$ satisfies the eigenvalue constraints in §4.3.1 on **inverse reweighting** then the remaining proof technology may be applied. Alternatively, we my apply **offset inverse reweighting** with assured convergence if the conditions in §4.3.2 are met. The convergence proof is further complicated if we allow ω to change over the sequence of iterations. By the Bolzano-Weierstrauss theorem a sequence$\{\omega_N\}$ converges as $||\omega|| =$

[3] A matrix has **simple structure** if all of its eigenvalues are simple. The eigenvalues have only linear elementary divisors.

1. In the next section we describe two possible ways to update ω between iterations of the `SR-Algorithm`.

Optimizing the mixture coefficients of q

An intuitively appealing update for the entries of ω is the reciprocals of the corresponding eigenvalues. That is, let $\hat{\omega}_i = \frac{1}{\sqrt{\lambda_i}}$ and $\omega = \frac{1}{||\hat{\omega}||_2}\hat{\omega}$. A potentially more powerful update to the entries of ω comes from working with the differentials of $q^T Lq$ with respect to ω_i.

The differentials of $\mathcal{O}(q) = \frac{1}{2}((q^T D'q)(q^T Lq) - q^T L'q)$ w.r.t ω_i, subject to $||\omega|| = 1$ can be written as:

$$\begin{aligned}\frac{\partial \mathcal{O}_{uv}}{\partial \omega_i} &= w_{uv}\left(\sum_{i=1}^{k}\omega_i^2(f_i(u)-f_i(v))^2 + \omega_i(f_i(u)-f_i(v))(q^2(u)+q^2(v))\right)\\ &\quad - \left((f_i(u)-f_j(v))(q(u)-q(v)) + \frac{\partial w'_{uv}}{\partial \omega_i}(f_i(u)-f_j(v))(q(u)-q(v))\right)\end{aligned} \tag{4.4.11}$$

recall that $w'_{uv} = \frac{\hat{w}'_{uv}}{\hat{w}'_{uv}+1}$, where $\hat{w}'_{uv} = (q^2(u)+q^2(v))/(q(u)-q(v))^2$. With the second order terms (which we omit) a standard Gauss-Newton iteration can be employed, optimizing the mixture coefficients ω_i so as to maximize $\mathcal{O}$. By maximizing $\mathcal{O}$ we move in the direction of steepest descent with respect to the mixture of the eigenvalues $\sum_{i=1}^{k}\omega_i^2\lambda_i(t)$. By lemma 5 and lemma 7 the $\mathcal{O}(q)$ is strictly positive and bounded.

4.4.2 Fractional Averages of Fractional Averages

For a $k-$way cut, the updated weights w_r and progress norm $||\cdot||_k$ are constructed as:

$$\begin{aligned} w_r(i,j) &= \Psi_h\left(\frac{\sum_{l=2}^k f_l^2(i)+f_l^2(j)}{\sum_{l=2}^k \frac{1}{\lambda_f}(f_l(i)-f_l(j))^2}\right)\cdot w(i,j) \\ ||\cdot||_k &= \frac{\sum_{i=1}^k f_i^T L f_i}{\sum_{i=1}^k f_i^T D f_i}. \end{aligned}$$

This update rule allows the information in a suite of eigenvectors to be used to determine a single reweighting that drives down the fractional average of the eigenvalues. The above notation is a bit overloaded, as f_i is taken to satisfy $Lf_i = \lambda_i D f_i$ for the current matrix. At present global convergence for this reweighting is not proven. We note that the updating is analogous to (MVR) where the numerator and denominator are allowed to have different mixture coefficients. In practice we observe that this method converges to a disconnected graph in a small number of iterations (less than 10 for most image derived graphs). This reweighting scheme was used in to obtain the results in Chapter 5 (see figure 5.3 for the effect of an iteration on the eigenvectors). The following lemma demonstrates that the above reweighting scheme drives the target eigenvalues down.

Lemma 13. *For a weighted graph $G = (V,E,w)$ with matrices L and D and simple eigenpairs $(f,\lambda_f) \mid Lf = \lambda_f Df$ and $(g,\lambda_g) \mid Lg = \lambda_g Dg$, given a reweighting w' such that*

$$\frac{\frac{1}{\lambda_f}f^T L' f + \frac{1}{\lambda_g}g^T L' g}{f^T D' f + g^T D' g} < \frac{\frac{1}{\lambda_f}f^T L f + \frac{1}{\lambda_g}g^T L g}{f^T D f + g^T D g} = 1 \tag{4.4.12}$$

then

$$\frac{d\lambda_f(t)}{dt} + \frac{d\lambda_g(t)}{dt} < 0$$

at $t = 0$.

Proof. We begin by stating a related quantity of interest, the derivative of the fractional average of Rayleigh quotients on f and g for the matrix curve $w = w + t \cdot w'$ as:

$$\frac{d}{dt}\left[\frac{\frac{1}{\lambda_f}f^T(t)L(t)f(t) + \frac{1}{\lambda_g}g^T(t)L(t)g(t)}{f^T(t)D(t)f(t) + g^T(t)D(t)g(t)}\right] \tag{4.4.13}$$

and examine its derivative centered at $t = 0$. First we must fix the scale of the eigenvectors $f(t)$ and $g(t)$, we choose $f(t)^T D(t) f(t) = g(t)^T D(t) g(t) = 1$ *w.l.o.g.* Thus equation 4.4.13 simplifies to

$$\frac{d}{dt}\left[\frac{\frac{1}{\lambda_f}f^T(t)L(t)f(t) + \frac{1}{\lambda_g}g^T(t)L(t)g(t)}{1+1}\right] = \frac{1}{2}\left(\frac{1}{\lambda_f}\frac{d}{dt}\lambda_f(t) + \frac{1}{\lambda_g}\frac{d}{dt}\lambda_g(t)\right)$$

by the linearity of the derivative. Substitute the functional form of $\frac{d\lambda(t)}{dt}$ and obtain

$$\begin{aligned} \frac{1}{2}\left(\frac{1}{\lambda_f}\frac{d}{dt}\lambda_f(t) + \frac{1}{\lambda_g}\frac{d}{dt}\lambda_g(t)\right) &= \\ \frac{1}{2}\left(\frac{1}{\lambda_f}f^T(L' - \lambda_f D')f + \frac{1}{\lambda_g}g^T(L' - \lambda_g D')g\right) & \end{aligned} \tag{4.4.14}$$

assume the bound holds on equation 4.4.14, thus

$$\begin{aligned} \frac{1}{2}\left(\frac{1}{\lambda_f}f^T L' f - \frac{\lambda_f}{\lambda_f}f^T D' f + \frac{1}{\lambda_g}g^T L' g - \frac{\lambda_g}{\lambda_g}g^T D' g\right) &< 0 \\ \left(\frac{1}{\lambda_f}f^T L' f - f^T D' f + \frac{1}{\lambda_g}g^T L' g - g^T D' g\right) &< 0 \end{aligned} \tag{4.4.15}$$

arriving at

$$\frac{1}{\lambda_f}f^T L' f + \frac{1}{\lambda_g}g^T L' g \quad < \quad f^T D' f + g^T D' g$$

which is equivalent to the hypothesis in equation 4.4.12. The remainder of the proof follows the continuity argument in lemma 6. □

In lemma 13 we proved that the fractional average of fractional averages drives the sum of the target eigenvalues down. This update rule can be applied to reweight the graph, and is highly effective (as shown in Chapter 5). At present, we observe that the method convergences to a disconnected graph empirically. If an intermediate fixed point were to arise, any of the previously covered methods could be applied for an iteration to break the symmetry across the edges.

4.4.3 Heuristics for Choosing $k < k' << n$

In this section we detail a heuristic for choosing a subset of eigenvectors with which to reweight the graph, based on their differential and structural properties. The problem of choosing which eigenvectors to use is common in spectral image processing and data mining applications. An ambiguity arises when the eigengap $\lambda_{k+1} - \lambda_k$ is small, or when two eigenvectors decompose the graph into the same number of **sign connected components** (see §2.4 lemma 3). Under such conditions, what criteria can we rely upon to select a "good" suite of eigenvectors for partitioning. An intuitive answer comes in the form of the derivatives of the generalized eigenvalues along the matrix curves constructed in the previous sections.

Recall that the number of **sign connected components** (SCCs) an eigenvector decomposes the graph into is, in part, determined by the eigenvector's spectral index. If the vector f is associated with the i^{th} smallest eigenvalue, then it may decompose the graph into (at most) i connected subgraphs. This provides a natural mechanism for grouping eigenvectors into equivalence classes – *i.e.* those vectors that decompose the graph into the same number of SCCs. For example, given a rectangular mesh the second and third eigenvectors both decompose the graph into two pieces. To motivate the use of the updated eigenvalues, recall the mesh example in Chapter 3 §3.2.1. Where the eigen-gap between λ_2 and λ_3 maybe quite small, but the derivative $\frac{d\,\lambda_3(t)}{d\,t}$ is far steeper than that of λ_2. In breaking the graph in two it appears sensible to choose the vector that will do so most rapidly.

To make the above intuition concrete we specify the following vector selection heuristic.

SCCs Heuristic:

1. compute $LF_{k'} = DF_{k'}\Lambda_{k'}$
2. group $F_{k'}$ into $k \mid k \leq k'$ sets where f_i and f_j are grouped if $\mathbf{SCCs}(f_i) = \mathbf{SCCs}(f_j)$
3. return the steepest vector from each set (combined for a $k-$way update).

The (approximate) vector of steepest descent in a population of vectors can be determined in a number of fashions. In the case where we employ a multi-valuation updating scheme we process the vectors in a coarse to fine fashion (small order **SCCs** first). That is, we construct aggregate reweighing matrix curves by augmenting the current reweighting (obtained in previous steps) by each of the candidate vectors independently to determine the affect on the eigenvalues. Given this assumption the most of obvious approximation is a partial Taylor expansion of the eigenvalues $\lambda(t) \approx \lambda(0) + t\frac{d\lambda}{dt}(0) + \left(\sum_{i=2}^{k} \frac{t^i}{i!} \cdot \frac{d^i\lambda}{d^i\lambda}(0)\right)$ along the matrix curves. In chapter 2 the first and second order derivatives of an eigenvalue were given in terms of the entries of the adjacency matrix A. In practice this may be extremely expensive as the higher order derivatives involve solving large linear systems. Accordingly we choose to perform a small number of powering operations to estimate the change in eigenvalues.

To automatically determine the number of segments we use a strategy much like the eigen-gap heuristic (see [AG06] for a recent treatment), but based on the effects each increment in the number of segments has on the eigen-gap after update reweighting. Therefore assuming that k' is an upper bound on desired the number of connected components – we may use the eigen-gap heuristic to choose a $k \mid k \leq k'$ – in effect breaking the graph into segments that are highly predicted by the eigenvectors. The notion that these are predictions is easily derived from the vector induced distributions over cuts shown chapter 2 (2.3.14).

Chapter 5

Results

This chapter presents quantitative and qualitative comparisons between the **spectral rounding** algorithm (SR) and the standard spectral algorithm. Each problem is formulated as the minimization of the Normalized Cut over partitionings of the graph. The proceeding evaluation focuses on the model-free segmentation of natural and medical images. The results indicate that reductions in cut cost correspond to increased overlap with human segmentations.

Spectral rounding compares favorably to recent spectral approximations for the Normalized Cut criterion (NCut). The evaluation compared (*SR*) to the method proposed by Yu and Shi [YS03a], (*EIG*), as it returned stable cuts with respect to parameter perturbation and initialization. This is in contrast to methods that employ K-means which tend to be highly sensitive to initialization – *e.g.* [NJW02]. In the following sections results are given in geometric clustering, and medical and natural image segmentation. Natural image segmentations results are given for an image collection obtained from Google image search and the Corel images used in Berkeley Hand Segmentation Database [MFTM01].

The remainder of this section introduces two quantitative measures of segmentations that form the basis of comparison. §5.1 compares the SR-algorithm and EIG on random geometric graphs. In §5.2.1 the SR-algorithm and EIG are compared on a collection of images extracted from Google image search. §5.2.2 evaluates SR and EIG on a simple medical image segmentation task. Model-free segmentations of the left ventricle were generated for a collection of NMR images over a 40 subject population. The segmentations were compared with those generated by an expert. Finally in §5.3 the SR-algorithm and the EIG-algorithm are compared on the Berkeley Segmentation Database. The variation in information is reported for both methods with respect to the human hand segmentations when appropriate. For the medical and Berkeley data sets, an improvement in normalized cut correlates with an increase in the mutual information between human and machine segmentations.

The Problem and Analysis

In all cases, the same graph $G = (V, E, w_o)$, is passed to both algorithms. The normalized cut cost is reported on the initial weighting of the graph w_o. The affinity graphs used on medical image evaluation and Google search collection were constructed using the Intervening Contour [MBLS01]. In the Berkeley Segmentation dataset the affinity graph was constructed using Martin and Fowlkes' "probability of a boundary" detector [MFTM01].

To compare the partitioning we report the expected change in NCut value, on the initial graph $G = (V, E, w)$, and the **variation of information** between clusterings. The expected change can be expressed as a positive scaler **c** such that $nc(SR) \approxeq \mathbf{c} \cdot nc(EIG)$ on average. In the case where $\mathbf{c} < 1$ *spectral rounding* finds an improved cut on average.

Throughout this section we report the **variation of information** described in [Mei03] to measure the difference between two partitionings, P_a and P_b of the vertex set V, defined as

$$\mathcal{D}_{vi}(P_a, P_b) = H(P_a) + H(P_b) - 2I(P_a, P_b) \tag{5.0.1}$$

with $I(P_a, P_b) = -\sum_{i=1}^{p} \sum_{j=1}^{p} \frac{n_{i,j}}{n} \log \frac{n_{i,j}}{n}$, $H(P) = -\sum_{k=1}^{p} \frac{n_k}{n} \log \frac{n_k}{n}$, where n_k is number of vertices in the k^{th} partition, $n_{i,j}$ the size of the overlap between the i^{th} partition in P_a and

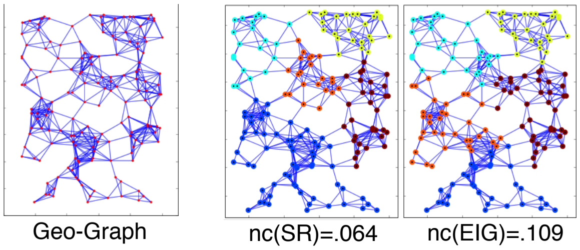


Figure 5.1: A ($|V| = 300$) geometric graph, and two 5-way cuts.

j^{th} partition in P_b, and $n = |V|$. The entropy term $H(P)$ can be thought of as the number of bits required to represent the distribution of partition sizes. Where as $I(P_a, P_b)$ is the mutual information between the two partitionings P_a and P_b. And so, $\mathcal{D}_{vi}$ can be thought of as the number of bits required to encode the aggregate difference between the overlapping partitions of P_a and P_b. As expected $\mathcal{D}_{vi}(P_a, P_a) = 0$.

5.1 Random Geometric Graphs

We compare *SR* and *EIG* in the expected partition cost on a collection of random geometric graphs. The vertices of $G = (V, E, w)$ are associated with uniformly distributed coordinates in $\mathbb{R}^d$. The edge set of $E(G)$ is then constructed using the following rule, for $\{u, v \in V | u \neq v\}$, $(u, v) \in E \iff dist(u, v) < r$. We sampled 10000 graphs with 1000 vertices and chose the radius r such that the expected degree of each vertex was approximately $log(|V|)$. As shown in Figure 5.1 such graphs afford a large number of inexpensive cuts. Table 5.1 contains the improvement factor, and the cluster divergence. We note that the divergence distance, relative to partition entropy $H(SR)$, highlights that the NCut improvements are not due to a small number of boundary vertex exchanges, but rather that SR and EIG return significantly different subgraphs.

	$\mathcal{D}_{vi}(SR, Eig)$	$nc(SR) = \mathbf{c} \cdot nc(EIG)$
geo-graph	$0.910 \pm .219$	$\mathbf{c} = .690 \pm .113$

Table 5.1: Comparison between spectral rounding *SR* and the multi-way cut algorithm of Yu and Shi [YS03a] *EIG*. The partition entropy for SR was $H(SR) \approxeq 1.935$.

5.2 Image Segmentation

The parameters used in constructing a weighted graph from an image were fixed for all the results presented in this section. The graph $G = (V, E, w)$ represents an image as follows. For each pixel in the image a vertex in V is assigned. If two pixels are connected in E a weight in w is determined based on the image data. The graph connectivity, E, was generated by connecting pixels to 15% of their neighboring pixels in a 5 pixel radius. The initial weighting w of the graph $G = (V, E, w)$ was determined using the *Intervening Contour* cue described in [MBLS01]. This cue assigns small weights to pixels which lie on opposite sides of a strong image boundary, and large weights otherwise.

5.2.1 Natural Image Segmentation (google dataset)

We compiled a set of a 100 images from Google Images using the keywords *farm, sports, flowers, mountains, & pets*. Examples from this data set, and segmentations can be found in Figure 5.9. Again, we note that changes in the cut value often correlate with large changes in the co-membership relationships on the image pixels. To quantitatively compare the methods on natural images we report the divergence distance and NCut improvement factor **c**. The effect of the SR iteration at an intermediate iteration can be found in figure 5.3.

	$\mathcal{D}_{vi}(SR, Eig)$	$nc(SR) = \mathbf{c} \cdot nc(EIG)$
natural	$1.23 \pm .160$	$\mathbf{c} = .536 \pm .201$

Table 5.2: Comparison between spectral rounding *SR* and the multi-way cut algorithm of Yu and Shi [YS03a] *EIG* on segmentations of natural images. The average cluster entropy over *SR*-segmentations of the image collection is $1.62 \pm .4$.

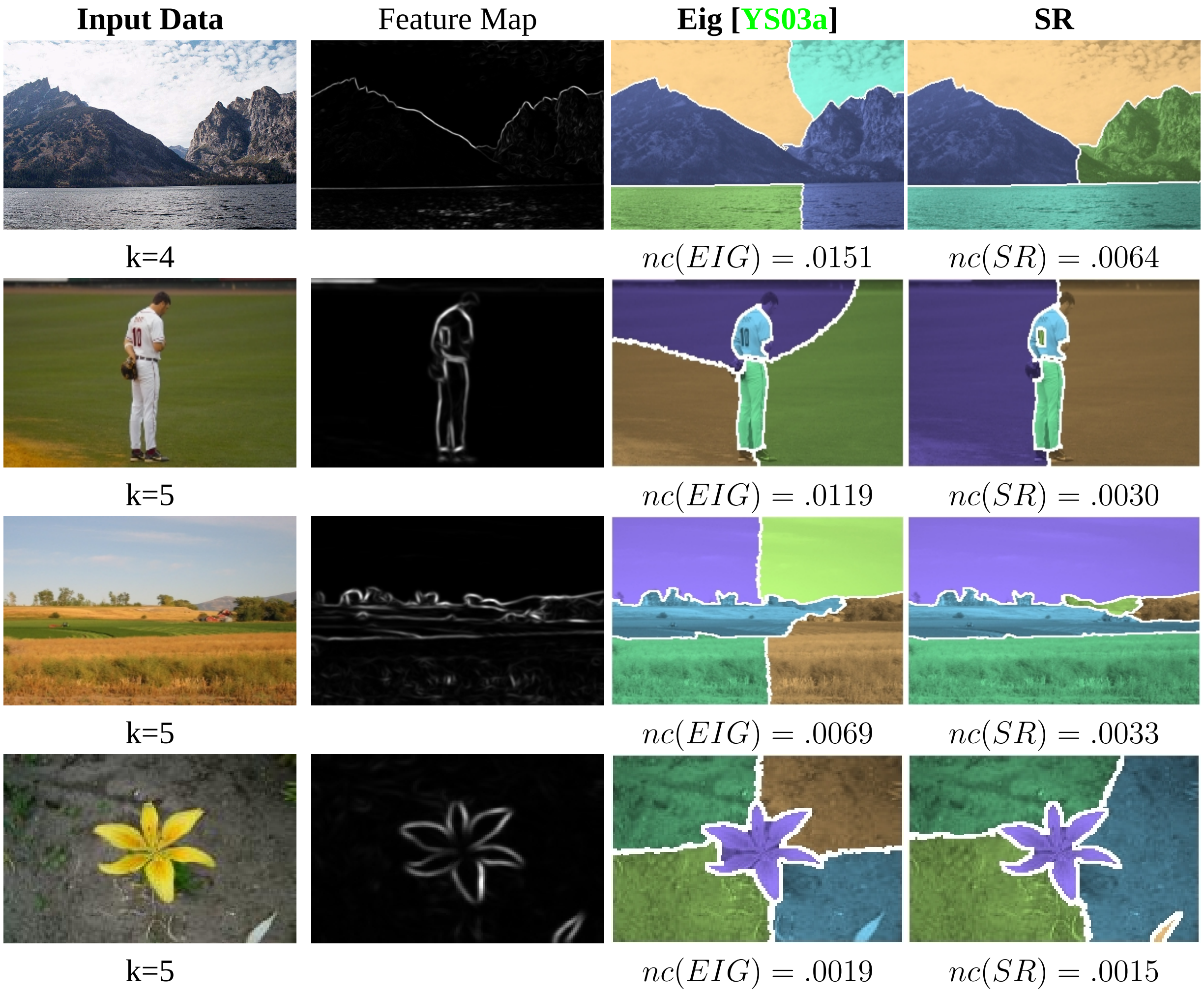
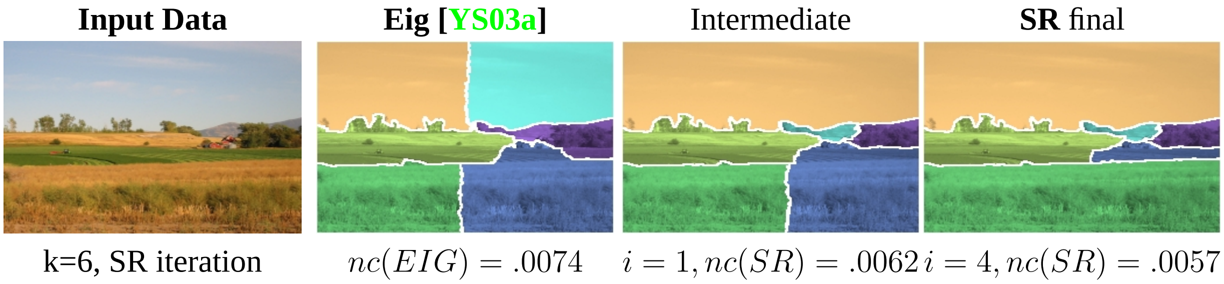


Figure 5.2: The first four rows provide qualitative examples of the improvements in NCut value for natural images. Column three contains segmentations generated by the published code of Yu and Shi [YS03a]. Column four contains results for *Spectral Rounding*. The number of segments k is fixed for each comparison. We emphasize that the cut cost was evaluated on identical combinatorial problems (graphs).

5.2.2 Medical Image Segmentation

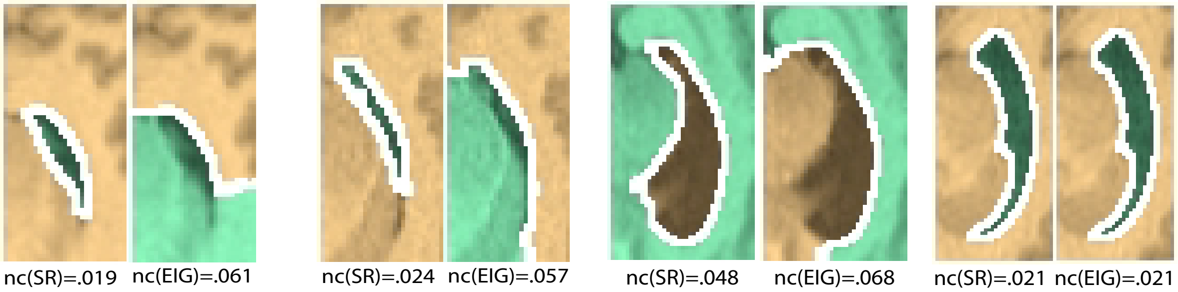
To a degree, clustering methods are only successful in that are useful in servicing a particular task. We have selected a simple medical task, segmenting out the left ventricle (a fluid sack located in the brain), as it is well defined – *i.e.* the boundary of the percept is agreed upon by experts. While this task appears to be relatively easy, a successful automatic method represents



Input Data | **Eig [YS03a]** | Intermediate | **SR** final

k=6, SR iteration | $nc(EIG) = .0074$ | $i = 1, nc(SR) = .0062$ | $i = 4, nc(SR) = .0057$

Figure 5.3: A sequence of iterations projected onto the feasible set, starting left with solution from Yu's method and ending with the fourth and final *SR* iteration on the right. Notice that the large cuts in the sky and field shift to small cuts in the area around the farm.



nc(SR)=.019 nc(EIG)=.061 | nc(SR)=.024 nc(EIG)=.057 | nc(SR)=.048 nc(EIG)=.068 | nc(SR)=.021 nc(EIG)=.021

Figure 5.4: Examples of the left ventricle, and qualitative results for the *SR* and *EIG* algorithms. Segmentations required approximately 1.2 seconds for *EIG* and 1.9 seconds for *SR*.

a significant reduction in human effort for a common labeling task.

The evaluation was performed on a collection of 200 NMR images containing the left ventricle. The collection was built by taking 5 slices each from scans of 40 individuals. Images were selected randomly from slices containing the left ventricle. As shown in Figure 5.4 the appearance of the ventricle varies substantially in shape and size.

The comparison of segmentations obtained from spectral rounding and the eigenvector method of Yu and Shi [YS03a] with the human labels is given in Table 5.3. The divergence distance and expected cut improvement are given in Table 5.4. The average cluster entropy for *SR* was $0.611 \pm .131$. As this is a two-class problem, this suggests that one of the segments tends to be much smaller than the other. This is due to the often small size of the ventricle in the image.

	$nc(SR)$	$nc(EIG)$[YS03a]
$Pr(v \in T(Im))$	$.95 \pm .04$	$.37 \pm .12$

Table 5.3: The value $Pr(v \in T(Im))$ is reported over the population of images, where $T(Im)$ is the expert's hand segmentation and $Pr(v \in T(Im))$ is the probability that a pixel v in a segment is also contained in $T(Im)$ – this statistic was computed for the segment with the majority overlap with $T(Im)$. Change this to a D_{vi} statistic for $(EIG, human)$ and $(SR, human)$.

	$\mathcal{D}_{vi}(SR, Eig)$	$nc(SR) = \mathbf{c} \cdot nc(EIG)$
medical	$1.856 \pm .192$	$\mathbf{c} = .598 \pm .237$

Table 5.4: The divergence and expected value improvement for the medical image data set. The average cluster entropy for *SR* segmentations on the medical data set was $0.611 \pm .131$.

5.3 Human Segmentation and SR

The Berkeley Human Segmentation Database (BHSD) [MFTM01] provides a means of evaluating the Normalized Cut as a measure of image segment salience. At first blush the question seems obvious, do human segmentations consistently cost less than machine generated cuts? Or, perhaps, do human segmentations span a large set of low cost cuts? Given the means of obtaining improved cuts automatically, does this decrease in NCut value predict an increase the overlap with a human segmentations? In this section we employ spectral rounding to improve the cut value over images in the BHSD dataset and demonstrate that cut improvements increase the mutual information between human segmentation and automatic segmentation.

Human segmentations often contain a large number of segments, resulting in the partitions with high cost. However, a coarsening of the segmentation, using the human selected partitions as a basis results in very low cost low order cuts (*i.e.* small k). In the following section we detail segmentation for the normalized cut over multiple orders grouping human segments to generate low-cost low-order cuts. These cut values and co-membership structures are used to compare the *SR* algorithm and standard spectral algorithm.

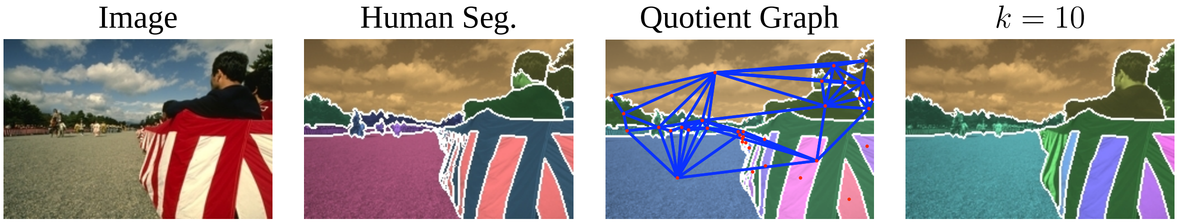


Figure 5.5: The input image and human segmentations are combined into a quotient graph representation of the image.

Boundary Correction

The human segmentation boundaries in the BHSD are often near, but slightly removed from, the locally optimal partition boundary. To amend this a local minimum cut boundary optimization is performed. The procedure is as follows, for each boundary. For adjacent image segments A and B a region of pixels is selected. We use $\delta(A)$ and $\delta(B)$ to denote the sets of vertices with edges in the "human" cut. The nearest set of pixels outside these regions are added to the sets. These vertex sets are labeled source and sink respectively. The minimum S-T cut of $\delta(A) \cup \delta(B)$ is computed. If the value of S-T cut is significantly less than that of the human generated boundary it is transplanted as the boundary between the two regions. In practice, this condition fails when an two image regions is too small, and therefore every pixel with in it is adjacent to the cut.

The best k-way human cut

We report segmentation comparisons for $k = 2, ..., 20$. Given a human, or set of human segmentations, we collapse the human segmentation down to each setting of k. For a given setting of k a "best human cut" is determined by searching over planar cuts in quotient graph defined by the fine grained human segmentation. This is illustrated in figure 5.5.

Judging significance

The BHSD contains 290 images, split into 197 train and 93 test images. Table 5.5 contains statistics on the reduction in **variation of information**, $\mathcal{D}_{vi}$, when using `SR` over `EIG` with respect to human segmentation. The table reports bootstrap estimates of standard errors and 70^{th} percentile sample confidence intervals. To determine if the improvement in $\mathcal{D}_{vi}$ is significant, we consider each algorithm as generating samples from a continuous random variable of (VoI) bits. The two distributions are highly skewed (asymmetric) and have dramatically different variances. First we employ the Kolmogorov-Smirnov two-sample test [Was04] to test the Null Hypothesis that the sample sets are drawn from the same underlying distribution. The test rejects the Null Hypothesis with a significance of $\alpha < .00005$, which verifies that the distributions are distinct. Thus the improvement (*i.e.* reduction) in variation of information is statistically significant by comparison of means.

We determined that the distributions over $\mathcal{D}_{vi}$ induced by the segmentation methods were invariant of *test-train* split of the image collection. A bootstrap Kolomogorov-Smirnov 2 sample-test was applied for each algorithm comparing the sample distributions for *test* and *train*. In both cases the Null Hypothesis was accepted indicating that results on the *test* and *train* sets were drawn over the same distribution over bits for each algorithm. Therefore we employ all 293 images in our analysis.

	$\mathcal{D}_{vi}(SR, Eig)$	$\mathcal{D}_{vi}(Eig, H)$	$\mathcal{D}_{vi}(SR, H)$	NCut
Std.	1.13 ± 0.30	$1.43 \pm .79$	$.634 \pm 0.32$	$\mathbf{c} = .58 \pm 0.2$
C.I.	NA	$1.47 : \{-.63, +0.53\}$	$0.61 : \{-0.25, +0.13\}$	NA

Table 5.5: The **variation of information**, $\mathcal{D}_{vi}$, for: the SR versus EIG, EIG versus Human, and the SR versus Human. Statistics are shown with standard errors and confidence intervals. The Kolmogorov-Smirnov two-sample test [Was04] verifies that the distributions over $\mathcal{D}_{vi}$ induced by the `SR` and `EIG` are distinct with a significance of $\alpha < .00005$. The forth column reports the expected improvement in normalized cut $nc(SR) = \mathbf{c} \cdot nc(EIG)$.

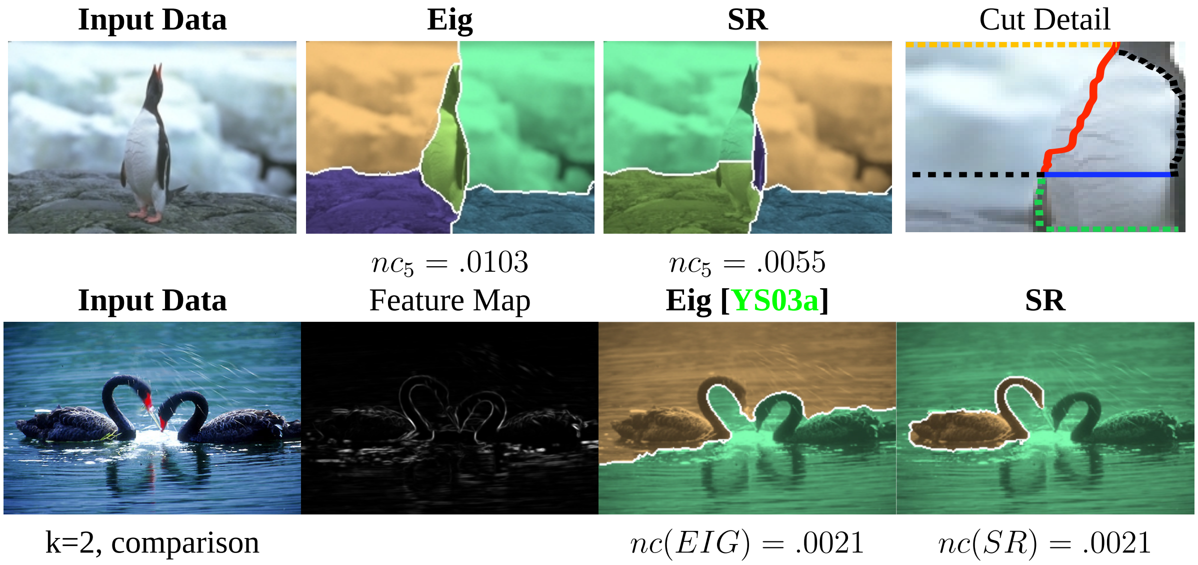


Figure 5.6: The first row contains a segmentation that does not match the common human perception of the foreground (a penguin). In the forth column, the lines in the image illustrate various node types in an *s-t cut* computation. The yellow line indicates the sink, the green the initial source, and the black lines denotes the boundary. The blue line is the minimum cut, the red line is the "forced" minimum cut with the blue line used as source. The second row illustrates a 2-way cut in which the NCut values are nearly identical, but which support very different percepts.

The Normalized Cut & Human Segmentations

As mentioned in the introduction, and demonstrated above, a superior Normalized Cut while positively correlated with perceptually relevant segmentations is not guaranteed to be match the common human perception of the image contents (percept). The images in figure 5.6 demonstrate that the pixel affinity graph may contain ambiguous segmentations, or may miss the target percept (see the second row).

The image containing the penguin, row 1 figure 5.6, is example of a case in which the cut that closely matches our perception of the foreground is not contained in the affinity graph. However, as illustrated in the forth column, the red edge cut does not actually trace the contour of the foreground, but rather is fortuitously close to it. In fact the cut, indicated in red, is more expensive than that indicated indicated in blue.

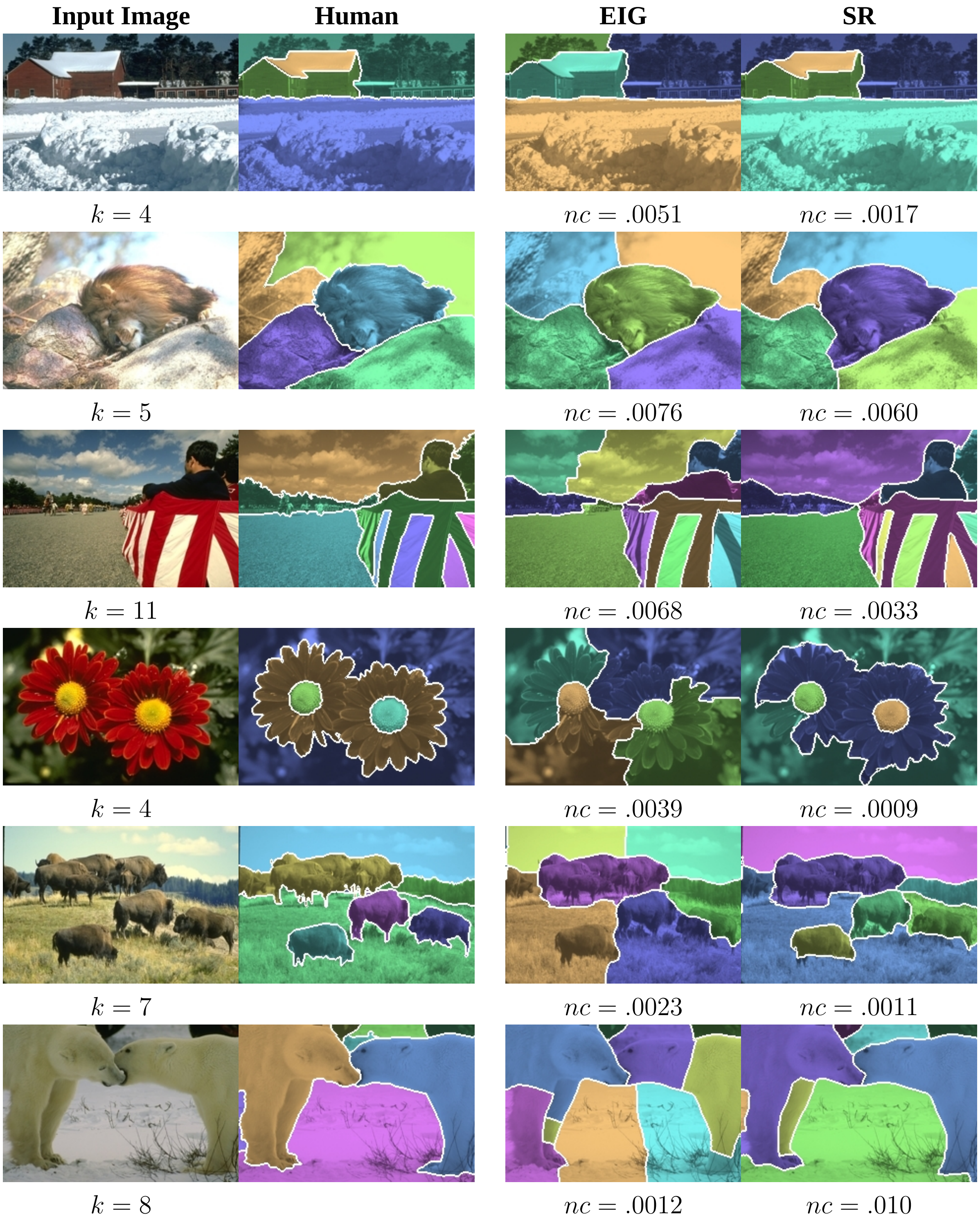


Figure 5.7: Example segmentations from the Berkeley Hand Segmentation Database. Image results comparing the $k-$way cut generated from hand segmentation (column 2), the standard spectral algorithm (column 3), and spectral rounding with expansion edges and the derivative heuristic (column 4). For each image, the number of segments was fixed for both the spectral rounding algorithm **SR** and the standard algorithm **Eig**. Each method was initialized with the same weight matrix, and the reported cut costs are given on the original weighted graph (*i.e.* affinity matrix).

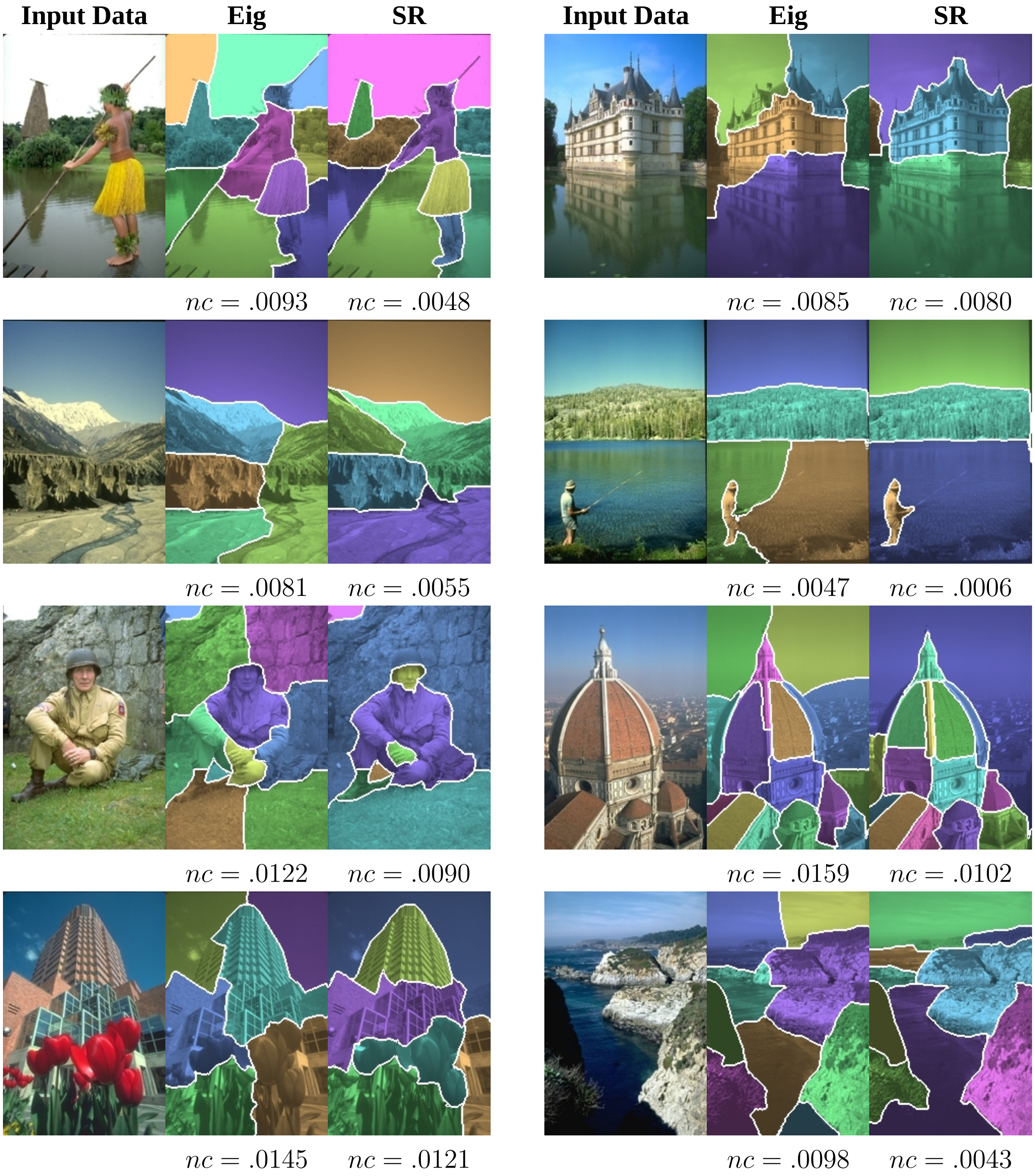


Figure 5.8: Example Images

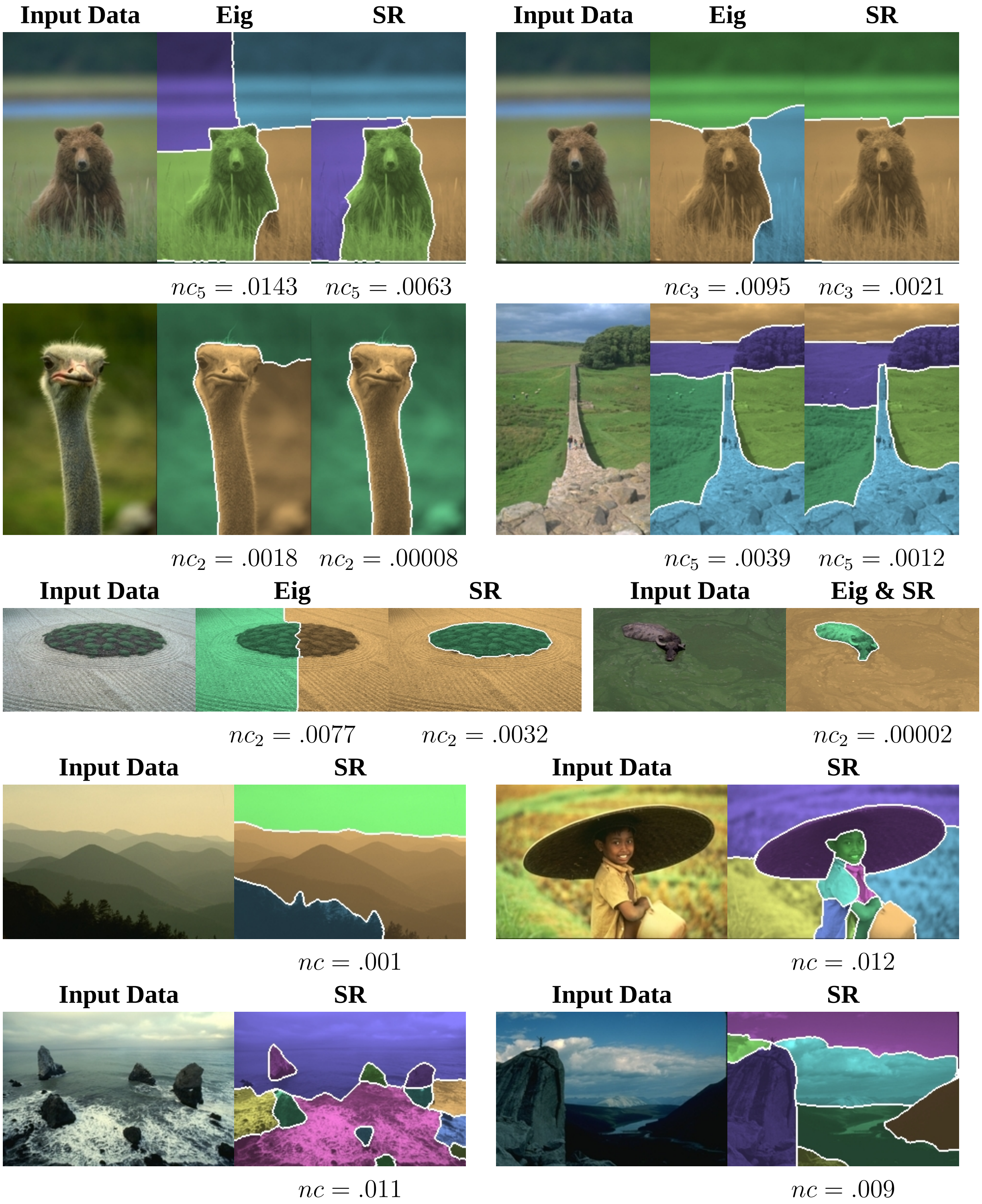


Figure 5.9: Example Images

Chapter 6

Moving Forward

Efficiency

Spectral graph theory has seen wide application in construction of efficient PDE solvers, basic algorithm construction, machine learning, and computer vision. However, the computational complexity of the underlying sparse eigencomputation remains a major barrier against the integration of spectral techniques into large scale data-mining problems and vision. At present the best known [KW92, AHK05] average case time bound on $\epsilon-$approximate estimation of an extremal eigenpair is $\tilde{O}(m\sqrt{\frac{n}{\epsilon}})$, for generalized Laplacians with $n = |V|$ and $m = |E|$.

Fortunately, empirical evidence and recent theoretical results suggest the possibility of nearly-linear time eigencomputation. In [TCB05] we demonstrated that rudimentary multi-level methods are effective for rapid approximate eigenvector computation. This is in part due to the smoothness of the systems arising from image data. Spielman and Teng [ST04] furnish a multi-level method for solving linear systems with a theoretical bound of near linear-time. Their work has been further improved to a linear-time algorithm in the planar graph case (a common case in vision applications) by Koutis and Miller. Such work suggests that $\epsilon-$approximate eigencomputation may soon fall into same time complexity, as iterative linear system solving is central to several techniques computing eigenvectors and eigenvalues. For example, the inverse iteration and Rayleigh quotient iteration [Dem97b] both compute eigenvectors by repeatedly solving linear systems that converge upon an eigenpair of a matrix. The Rayleigh quotient iteration exhibits cubic convergence yeilding a set of iterations k where $k < log(m)$. However, the deflation step required in the Rayleigh quotient iteration must be addressed in order to successfully adapt [ST04] to eigenvector computation. As [ST04] assume that the matrix is diagonally dominant – which will not hold true for a deflated matrix. Another promising direction of inquiry involves employing Fiedler's theorem directly in an eigensolver. We are currently exploring both strategies for efficient Laplacian eigenvector computation.

An improved theoretical bound

We have presented a new rounding technique for the eigenvectors of graph laplacians in order to partition a graph. The method was shown to converge and demonstrated empirical improvements over the standard rounding strategy on a variety of problems. In ongoing work we are seeking a theoretical bound for variants of the `SR-Algorithm` in terms of the Normalized Cut.

As the `SR-Algorithm` is exactly a spectral algorithm we have no way of improving upon the Cheeger bound without modifying the approach. This sad state of affairs is a direct consequence of the work of Guattery and Miller [GM98], in which they construct families of graphs where the required eigenvectors (for a good cut) are given arbitrarily large spectral index. However, as shown in Chapter 3 modifying the graph, so as to correct some of the spectral methods short comings may provide a route to such improvements. The expansion augmentation strategy put forward in Chapter 3 provides one possible to path two achieving such an improved bound. The plausibility of such an improvement is high due to the success of [KRV06] with their expansion embedding algorithm for bounding the conductance number. For practical algorithms, the issue with graph augmentation is speed – to this end we are seeking a strictly spectral version of expansion augmentation coupled with the `SR-Algorithm`.

Guided Cuts

In many domains, such as medical image processing, we know the type of structure we wish to segment in the image or volume. Alternately, in graphics and image editing applications the user is often able to provide a sparse set of inputs that suggest a particular target segmentation. Accordingly, developing a mechanism for guiding the optimization of the graph cut with side is highly desirable.

In previous work, [TMC05] we put forth a joint spectral partitioning and shape estimation optimization that improved the over all segmentation and shape estimates relative of human hand segmentation. We observed that in optimizing shape based segmentations, eigenvector methods possess desirable properties compared to flow based optimization. Primarily this follows from examining the intermediate solutions generated by both techniques. For spectral methods there is a great deal of global information about the geometry of the cut contained in eigenvectors of the graph. Accordingly, they can be used to update estimates of shape parameters. This differs from the intermediate solutions in the flow optimization are not necessarily geometrically meaningful. However, flow-based methods support statistical parameter estimation over populations as shown in the spatially coherent clustering method proposed by Zabih [ZK04].

In [TMC05] we incorporate model data into a segmentation by aligning the target eigenvectors with a vector (or subspace) encoding the current estimate of the shape. The shape estimate is then updated by fitting regions in the Fiedler vector that are likely to contain the cut. This resulted in an alternating gradient ascent algorithm switching between shape parameters estimation (with respect to the current eigenvectors) followed by an alignment step in which matrix entries are updated to align the eigenvector to the current shape estimate.

In ongoing work we use “the probability of an edge” (in the cut) from Chapter 2 and the probability of a contour passing through an image region (which can also be expressed as a

$Pr((uv) \in \delta(S, \bar{S}))$ to provide strictly probabilistic shape optimization. This has the advantage of providing a strictly probabilistic algorithm for optimization shape and cut with respect to matrix eigenstructure. Further, human data can be natural incorporated into such a scheme as constraints on the distribution over cuts.

Bibliography

[AG06] Arik Azran and Zoubin Ghahramani. Spectral methods for automatic multiscale data clustering. In *Computer Vision and Pattern Recognition*, pages 190–197, 2006. 37, 78

[AHK05] S. Arora, E. Hazan, and S. Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. *FOCS*, pages 339–348, 2005. 93

[ARV04] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *STOC*, pages 222–231, 2004. 3, 24, 46, 47

[AW00] K. Anstreicher and H. Wolkowicz. On lagrangian relaxation of quadratic matrix constraints. *SIAM Journal Matrix Analysis Applications*, 22(1):44–55, 2000. 23

[Bes86] Julian Besag. On the statistical analysis of ditry pictures. *Journal of the Royal Statistical Society*, 3:259–302, 1986. 4

[Bis95] Christopher Bishop. *Neural Networks for Pattern Recognition*. 1995. 4, 5

[BJ06] Yuri Boykov and Olivier Juan. Active graph cuts. volume 1, pages 1023–1029, 2006. 46

[BKS02] A. Brook, R. Kimmel, and N. A. Sochen. Variational segmentation of color images. *IJCV*, ?(?):?–?, 2002. 5

[Bol02] Béla Bollobàs. *Modern Graph Theory*, volume 184. springer, 2002. 44, 47

[Boy04] K. Boyd. *Convex Optimization*. Cambridge University Press, 1st edition, 2004. 54

[BU02] E. Borenstein and S. Ullman. Class-specific top-down segmentation. *ECCV*, LNCS(2351):109–122, 2002. 8

[BVZ98] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *CVPR*, pages 648–655, 1998. 7

[CdV90] Y. Colin de Verdiére. On a new graph invariant and a criterion for planarity. *Graph Structure Theory, Contemporary Mathematics*, AMS:137–147, 1990. 35

[CGT94] Tony F. Chan, John R. Gilbert, and Shang-Hua Teng. Geometric spectral bisection. Manuscript, July 1994. 3, 24, 36, 37

[Chu92] F. R. K. Chung. *Spectral graph theory*. Number 92 in Regional Conference Series in Mathematics. Amer. Math. Soc, Providence, 1992. 3, 16, 17, 24

[CM99] Dorian Comaniciu and Peter Meer. Mean shift analysis and applications. *International Conference on Computer Vision*, pages 1197–1203, 1999. 4, 5

[CM02] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. In *PAMI*, volume 24, pages 603–619, 2002. 4

[CV01] T. F. Chan and L. A. Vese. Active contours without edges. *Image Processing*, 10(2):266–277, 2001. 6

[Dem97a] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1st edition, 1997. 46

[Dem97b] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1st edition, 1997. 94

[DS80] E. Diday and J.C. Simon. Clustering analysis. In *Digital Pattern Recognition*, pages 47–94, 1980. 4

[FF62] L. Ford and D. Fulkerson. *Flow in Networks*. Princeton University Press, 1962. 6

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematics Journal*, 98(23), 1973. 35, 39

[GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *PAMI*, volume 6, pages 721–741, 1984. 4, 6

[GG01] C. Godsil and R. Gordon. *Algebriac Graph Theory*. Springer, 2001. 35, 38

[GM98] S. Guattery and G. L Miller. On the quality of spectral separators. *Matrix Analysis & Applications*, 19(3), 1998. 30, 39, 45, 94

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *ACM*, 1995. 47

[HEH05] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Proc. 10th Int'l Conf. Computer Vision*, volume 2, 2005. 1

[HEH06] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *Conf. on Computer Vision and Pattern Recognition*, 2006. 1

[Ish03] H. Ishikawa. Exact optimization for markov random fields with convex priors. In *PAMI*, volume 25, pages 1333–1336, 2003. 7

[JS89] M. Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989. 16, 17

[Kle03] J. Kleinberg. An impossibility theorem for clustering. *Advances in Neural Information Processing Systems*, 2003. 2

[KRV06] Rohit Khandekar, Satish Rao, and Umesh Vasirani. Graph partitioning using single commodity flows. In *STOC*, pages 385–390, 2006. 45, 46, 94

[KW92] J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalue by the power and lanczos algorithms with a random start. *Matrix Analysis & Applications*, 4(13):1094–1122, 1992. 93

[KWT87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1987. 5

[KWT01] M. Kass, A. Witkin, and D. Terzopoulos. Image segmentation by data driven markov chain monte carlo. In *ICCV*, 2001. 7

[KZ04] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004. 7

[Lax97] P. D. Lax. *Linear Algebra*. Pure and Applied Mathematics. Wiley-Interscience, 1st edition, 1997. 18, 58

[Li01] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag: Tokyo, 2001. 4

[LR88] F. T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *FOCS*, pages 422–431. IEEE, October 1988. 3, 24

[LR04] K. Lang and S. Rao. A flow-based method for improving the expansion or conduectance of graph cuts. *Integer Programming and Combinatorial Optimization*, pages 325–337, June 2004. 45

[MBLS01] J. Malik, S Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. In *IJCV*, volume 43(1), pages 7–27, June 2001. 80, 82

[Mei03] M. Meilă. Comparing clusterings by the variation of information. *COLT*, pages 173–187, 2003. 80

[Mei05] M. Meilă. Comparing clusterings – an axiomatic view. *ICML*, pages 577–584, 2005. 2

[MFTM01] 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'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 79, 80, 85

[MN99] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus: with applications in statistics and econometrics*. John Wiley and Sons Ltd., revised edition, 1999. 18, 21, 73

[Moh89] B Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, 47:274–291, 1989. 27

[MS01] M. Meilă and J. Shi. A random walks view of spectral segmentation. In *AISTATS*, 2001. 16

[NJW02] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002. 2, 16, 24, 30, 36, 37, 79

[PD99] N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. *ICCV*, 2:926–932, 1999. 6

[RGDP01] Tibshirani R., Walther G., Botstein D., and Brown P. Cluster validation by prediction strength. In *Technical Report, Department of Statistics, Stanford University*, September 2001. 37

[RLBB06] Andrew Rabinovich, Tilman Lange, Joachim Buhmann, and Serge Belongie. Model order selection and cue combination for image segmentation. In *Computer Vision and Pattern Recognition*, 2006. 37

[SBB00] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *CVPR*, pages 1070–1077, November 2000. 16

[Set96] J.A. Sethian. *Level Set Methods*. Cambridge University Press, 1996. 4

[SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. In *PAMI*, 2000. vi, 2, 3, 4, 7, 16, 24, 30, 36, 37, 48

[ST96] D. Spielman and S.H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *FOCS*, 1996. 30, 43

[ST04] D. Spielman and S.H. Teng. Nearly linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. *STOC*, pages 81–90, 2004. 93, 94

[SZHW03] N. Shental, A. Zomet, T. Hertz, and Y. Weiss. Learning and inferring image segmentations with the gbp typical cut algorithm. *ICCV*, 2003. 2

[TCB05] D. A. Tolliver, R. T Collins, and S. Baker. Multilevel spectral partitioning for efficient image segmentation and tracking. In *WACV*, 2005. 93

[TCYZ05] Z. Tu, A. Chen, A. L. Yuille, and S.C Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *Proc. 10th Int'l Conf. Computer Vision*, volume 2, pages 113–140, 2005. 1

[TM06] D. A. Tolliver and G. L. Miller. Graph partitioning by spectral rounding: applications in image segmentation and clustering. *Computer Vision and Pattern Recognition*, June 2006. 4

[TMC05] D. A. Tolliver, G. L. Miller, and R. T Collins. Corrected laplacians: closer cuts and shape integration. In *Conf. on Computer Vision and Pattern Recognition*, 2005. 9, 95

[TMVJ02] Lange T., Braun M., Roth V., and Buhman J. Stability-based model selection. In *NIPS*, 2002. 37

[Vek00] O. Veksler. Image segmentation by nested cuts. In *Computer Vision and Pattern Recognition*, pages 339–344, June 2000. 4, 7

[Was04] Larry Wasserman. *All of Statistics*. Springer, 2004. iii, 87

[XJ03] E. P. Xing and M. I. Jordan. On semidefinte relaxtion for normalized k-cut and connections to spectral clustering. *TR-CSD-03-1265*, University of California Berkeley, 2003. 3, 24, 30

[YS02] S. X. Yu and J. Shi. Grouping with bias. In *NIPS*, 2002. 7

[YS03a] S. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, October 2003. iii, vi, vii, 2, 4, 23, 24, 30, 36, 37, 48, 79, 82, 83, 84, 85, 88

[YS03b] S. X. Yu and J. Shi. Object specific figure ground segregation by graph partitioning. *CVPR*, pages 16–22, 2003. 8

[ZK04] R. Zabih and V. Kolmogorov. Spatially coherent clustering with graph cuts. *CVPR*, 2:437–444, 2004. 7, 8, 95

[ZLY95] S.C. Zhu, T.S. Lee, and A. Yuille. Region competition: unifying snakes, region growing and mdl for image segmentation. In *ICCV*, volume 24, pages 416–425, 1995. 4