Isoperimetric Graph Partitioning for Data Clustering and Image Segmentation

Leo Grady[1] and Eric L. Schwartz[2]

July 29, 2003

[1] L. Grady is with the Department of Cognitive and Neural Systems, Boston University, Boston, MA 02215, E-mail: lgrady@cns.bu.edu

[2] E. L. Schwartz is with the Departments of Cognitive and Neural Systems and Electrical and Computer Engineering, Boston University, Boston, MA 02215, E-mail: eric@bu.edu

Abstract

Spectral methods of graph partitioning have been shown to provide a powerful approach to the image segmentation problem. In this paper, we adopt a different approach, based on estimating the isoperimetric constant of an image graph. Our algorithm produces the high quality segmentations and data clustering of spectral methods, but with improved speed and stability.

1 Introduction

The application of graph theoretic methods to spatial pattern analysis has a long history, including the pioneering work of Zahn [1] on minimal spanning tree clustering, the development of connectivity graph algorithms for space-variant sensors by Wallace et al. [2], and the seminal work on image segmentation, termed “Ncuts” by Shi and Malik [3]. One reason for this interest is that the segmentation quality of Ncuts and other graph-based segmentation methods [4, 5, 6] is very good. However, there are several other important advantages of graph-based sensor strategies.

1.1 Motivation for using graph theoretic approaches in image processing

There are at least four distinct reasons to employ graph theoretic approaches to image segmentation:

1. *Local-global interactions* are well expressed by graph theoretic algorithms. Zahn [1] used a minimal spanning tree on a weighted graph to illustrate Gestalt clustering methods. The term “Gestalt” derives from early theories of visual psychology which attempted to relate local and global features of visual stimuli in terms of “rules” which may be best described as simple variational principles (e.g., “best completion”, etc.). Zahn’s results were impressive for the time, coming at the very beginnings of modern image processing and clustering. The central reason for this success, we believe, is that the minimal spanning tree defines a minimizing principle (e.g., a tree of minimal edge weights) which respects the global structure of the problem set, allowing a simple local rule (e.g., cut the graph at peak values of local density measure

[1]) to effectively cluster a set of feature vectors. Many graph theoretic approaches involve the use of global and local information, as will be made explicit later in the present paper. As Zahn originally pointed out, the important notion of Gestalt in image processing—the relationship of the whole to the part—seems to be an important ingredient in both biological and machine image processing.

2. *New algorithms* for image processing may be crafted from the large corpus of well-explored algorithms which have been developed by graph theorists. For example, spectral graph partitioning was developed to aid in design automation of computers [7] and has provided the foundation for the development of the Ncuts algorithm [3]. Similarly, graph theoretic methods for solving lumped, Ohmic electrical circuits based on Kirchhoff's voltage and current law [8, 9, 10, 11], form the basis for the method proposed in this paper for solving the isoperimetric problem.

3. *Adaptive sampling* and space-variant vision require a "connectivity graph" approach to allow image processing on sensor architectures with space-variant visual sampling. Space-variant architectures have been intensively investigated for application to computer vision for several decades [2, 12] partly because they offer extraordinary data compression. A sensor [12] employing a complex logarithmic visual sampling function can provide equivalent peak resolution to the workspace size of a constant resolution sensor with 10,000 times the pixel count [13]. However, even simple space-variant architectures provide significant challenges with regard to sensor topology and anti-aliasing. The connectivity graph was proposed by Wallace et al. [2] as a general algorithmic framework for processing the data output from space-variant sensors. This architecture describes sensor pixels by a specific neighborhood connectivity as well as geometric position. Using this approach, a variety of image processing algorithms were defined on an arbitrary pixel architecture, including spatial filtering via the graph Laplacian.

4. *New architectures for image processing* may be defined that generalize the traditional Cartesian design. Vision sensors are generally based on fixed size pixels with fixed rate clocks (i.e., they are space-invariant and synchronous). The space-invariant pixel design, as noted above, can lead to a huge inefficiency when compared to a spatially adaptive

sensor (e.g., a foveal architecture). A similar issue holds for temporal sampling, indicating that biological systems once again provide a counterexample to current engineering practice. Retinal sensors are not synchronous, but are based on an asynchronous “integrate and fire” temporal design. They integrate the locally available intensity, and fire when a fixed threshold is achieved. It is evident that this “just in time” strategy for temporal sampling will improve the average response speed of the ensemble of sensors. Presumably, a slight difference in response speed may translate to a significant difference in survival value. Just as in the spatial case, the temporal domain can (and does, in animals) exploit an adaptive, variable sampling strategy. In a computational context, this suggests the use of graph theoretic data structures, rather than pixels and clocks. In turn, the flexible data structures based on graphs, which are familiar in computer graphics, have been relatively unexplored in computer vision. The structure and algorithms of graph theory provide a natural language for space-time adaptive sensors.

1.2 Overview of Graph partitioning

The graph partitioning problem is to choose subsets of the vertex set such that the sets share a minimal number of spanning edges while satisfying a specified cardinality constraint. Graph partitioning appears in such diverse fields as parallel processing [14], solving sparse linear systems [15], VLSI circuit design [16] and image segmentation [3, 17, 6, 5].

Methods of graph partitioning may take different forms, depending on the number of partitions required, whether or not the nodes have coordinates, and the cardinality constraints of the sets. In this paper, we use the term **partition** to refer to the assignment of each node in the vertex set into two (not necessarily equal) parts. We propose a partitioning algorithm termed **isoperimetric partitioning**, since it is derived and motivated by the equations defining the isoperimetric constant (to be defined later). Isoperimetric partitioning does not require coordinate information about the graph and allows one to find partitions of an “optimal” cardinality instead of a predefined cardinality. The isoperimetric algorithm most closely resembles spectral partitioning in its use and ability to create hybrids with other algorithms (e.g., multilevel spectral partitioning [18], geometric-spectral partitioning [19]), but requires the solution to a large, sparse system of equations, rather than solving the eigenvector problem for a large, sparse matrix. In this paper we will develop the isoperimetric algorithm, prove some of its properties, and apply it to problems in data clustering and image segmentation.

1.3 The Isoperimetric Problem

Graph partitioning has been strongly influenced by properties of a combinatorial formulation of the classic isoperimetric problem: *Find a boundary of minimum perimeter enclosing maximal area.*

Define the **isoperimetric constant** h of a manifold as [20]

$$h = \inf_S \frac{|\partial S|}{\mathrm{Vol}_S}, \tag{1.1}$$

where S is a region in the manifold, Vol_S denotes the volume of region S, $|\partial S|$ is the area of the boundary of region S, and h is the infimum of the ratio over all possible S. For a compact manifold, $\mathrm{Vol}_S \leq \frac{1}{2}\mathrm{Vol}_{\mathrm{Total}}$, and for a noncompact manifold, $\mathrm{Vol}_S < \infty$ (see [21, 22]).

We show in this paper that the set (and its complement) for which h takes a minimum value defines a good heuristic for data clustering and image segmentation. In other words, finding a region of an image that is simultaneously both large (i.e., high volume) and that shares a small perimeter with its surroundings (i.e., small boundary) is intuitively appealing as a “good” image segment. Therefore, we will proceed by defining the isoperimetric constant on a graph, proposing a new algorithm for approaching the sets that minimize h, and demonstrate applications to data clustering and image segmentation.

2 The Isoperimetric Partitioning Algorithm

A **graph** is a pair $G = (V, E)$ with vertices (nodes) $v \in V$ and edges $e \in E \subseteq V \times V$. An edge, e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. Let $n = |V|$ and $m = |E|$ where $|\cdot|$ denotes cardinality. A **weighted graph** has a value (typically nonnegative and real) assigned to each edge called a **weight**. The weight of edge e_{ij}, is denoted by $w(e_{ij})$ or w_{ij}. Since weighted graphs are more general than unweighted graphs (i.e., $w(e_{ij}) = 1$ for all $e_{ij} \in E$ in the unweighted case), we will develop all our results for weighted graphs. The **degree** of a vertex v_i, denoted d_i is

$$d_i = \sum_{e_{ij}} w(e_{ij}) \;\; \forall \, e_{ij} \in E. \tag{2.1}$$

For a graph, G, the **isoperimetric constant** [21], h_G is

$$h_G = \inf_S \frac{|\partial S|}{\mathrm{Vol}_S}, \tag{2.2}$$

where $S \subset V$ and

$$\mathrm{Vol}_S \leq \frac{1}{2}\mathrm{Vol}_V. \tag{2.3}$$

In graphs with a finite node set, the infimum in (2.2) becomes a minimum. Since we will be computing only with finite graphs, we will henceforth use a minimum in place of an infimum. The boundary of a set, S, is defined as $\partial S = \{e_{ij} | i \in S, j \in \overline{S}\}$, where $\overline{S}$ denotes the set complement, and

$$|\partial S| = \sum_{e_{ij} \in \partial S} w(e_{ij}). \tag{2.4}$$

In order to determine a notion of volume for a graph, a metric must be defined. Different choices of a metric lead to different definitions of volume and even different definitions of a combinatorial Laplacian operator (see [22, 23]). Dodziuk suggested [24, 25] two different notions of combinatorial volume,

$$\mathrm{Vol}_S = |S|, \tag{2.5}$$

and

$$\mathrm{Vol}_S = \sum_i d_i \;\; \forall\, v_i \in S. \tag{2.6}$$

The combinatorial isoperimetric constant based on equation (2.5) is what Shi and Malik call the "Average Cut" [3]. The matrix used in the Ncuts algorithm to find image segments corresponds to the combinatorial Laplacian matrix under the metric defined by (2.6). Traditional spectral partitioning [26] employs the same algorithm as Ncuts, except that it uses the combinatorial Laplacian matrix defined by the metric associated with (2.5). In agreement with [3], we find that the second metric (and hence, volume definition) is more suited for image segmentation since regions of uniform intensity are given preference over regions that simply possess a large number of pixels. Therefore, we will use Dodziuk's second metric definition and employ volume as defined in equation (2.6).

For a given set, S, we term the ratio of its boundary to its volume the **isoperimetric ratio**, denoted by $h(S)$. The **isoperimetric sets** for a graph,

G, are any sets S and $\overline{S}$ for which $h(S) = h_G$ (note that the isoperimetric sets may not be unique for a given graph). The specification of a set satisfying equation (2.3), together with its complement may be considered as a *partition* and therefore we will use the term interchangeably with the specification of a set satisfying equation (2.3). Throughout this paper, we consider a good partition as one with a low isoperimetric ratio (i.e., the optimal partition is represented by the isoperimetric sets themselves). Therefore, our goal is to maximize Vol_S while minimizing $|\partial S|$. Unfortunately, finding isoperimetric sets is an NP-hard problem [21]. Our algorithm is therefore a heuristic for finding a set with a low isoperimetric ratio that runs in polynomial time.

2.1 Derivation of Isoperimetric Algorithm

Define an indicator vector, x, that takes a binary value at each node

$$x_i = \begin{cases} 0 & \text{if } v_i \in \overline{S}, \\ 1 & \text{if } v_i \in S. \end{cases} \tag{2.7}$$

Note that a specification of x may also be considered as a partition.

Define the $n \times n$ matrix, L, of a graph as

$$L_{v_i v_j} = \begin{cases} d_i & \text{if } i = j, \\ -w(e_{ij}) & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

The notation $L_{v_i v_j}$ is used to indicate that the matrix L is being indexed by vertices v_i and v_j. This matrix is also known as the **admittance matrix** in the context of circuit theory or the **Laplacian matrix** (see, [27] for a review) in the context of finite difference methods (and in the context of [24]).

By definition of L,

$$|\partial S| = x^T L x, \tag{2.9}$$

and $\text{Vol}_S = x^T d$, where d is the vector of node degrees. If r indicates the vector of all ones, maximizing the volume of S subject to $\text{Vol}_S \leq \frac{1}{2}\text{Vol}_V = \frac{1}{2} r^T d$ may be done by asserting the constraint

$$x^T d = \frac{1}{2} r^T d. \tag{2.10}$$

Thus, the isoperimetric constant (2.2) of a graph, G, may be rewritten in terms of the indicator vector as

$$h_G = \min_x \frac{x^T Lx}{x^T d}, \tag{2.11}$$

subject to (2.10). Given an indicator vector, x, then $h(x)$ is used to denote the isoperimetric ratio associated with the partition specified by x.

The constrained optimization of the isoperimetric ratio is made into a free variation via the introduction of a Lagrange multiplier Λ [28] and relaxation of the binary definition of x to take nonnegative real values by minimizing the cost function

$$Q(x) = x^T Lx - \Lambda(x^T d - \frac{1}{2} r^T d). \tag{2.12}$$

Since L is positive semi-definite (see, [29, 30]) and $x^T d$ is nonnegative, $Q(x)$ will be at a minimum for any critical point. Differentiating $Q(x)$ with respect to x yields

$$\frac{dQ(x)}{dx} = 2Lx - \Lambda d. \tag{2.13}$$

Thus, the problem of finding the x that minimizes $Q(x)$ (minimal partition) reduces to solving the linear system

$$2Lx = \Lambda d. \tag{2.14}$$

Henceforth, we ignore the scalar multiplier 2 and the scalar Λ since, as we will see later, we are only concerned with the relative values of the solution.

Unfortunately, the matrix L is singular: all rows and columns sum to zero (i.e., the vector r spans its nullspace), so finding a unique solution to equation (2.14) requires an additional constraint.

We assume that the graph is connected, since the optimal partitions are clearly each connected component if the graph is disconnected (i.e., $h(x) = h_G = 0$). Note that in general, a graph with c connected components will correspond to a matrix L with rank $(n - c)$ [29]. If we arbitrarily designate a node, v_g, to include in S (i.e., fix $x_g = 0$), this is reflected in (2.14) by removing the gth row and column of L, denoted by L_0, and the gth row of x and d, denoted by x_0 and d_0, such that

$$L_0 x_0 = d_0, \tag{2.15}$$

which is a nonsingular system of equations.

Solving equation (2.15) for x_0 yields a real-valued solution that may be converted into a partition by setting a threshold (see below for a discussion of different methods). In order to generate a clustering or segmentation with more than two parts, the algorithm may be recursively applied to each partition separately, generating subpartitions and stopping the recursion if the isoperimetric ratio of the cut fails to meet a predetermined threshold. We term this predetermined threshold the `stop` parameter and note that since $0 \leq h(x) \leq 1$, the `stop` parameter should be in the interval $(0, 1)$. Since lower values of $h(x)$ correspond to more desirable partitions, a stringent value for the `stop` parameter is small, while a large value permits lower quality partitions (as measured by the isoperimetric ratio). In Appendix 1 we prove that the partition containing the node corresponding to the removed row and column of L must be connected, for any chosen threshold i.e., the nodes corresponding to x_0 values less than the chosen threshold form a connected component.

2.2 Circuit analogy

Equation (2.14) also occurs in circuit theory when solving for the electrical potentials of an ungrounded circuit in the presence of current sources [10]. After grounding a node in the circuit (i.e., fixing its potential to zero), determination of the remaining potentials requires a solution of (2.15). Therefore, we refer to the node, v_g, for which we set $x_g = 0$ as the **ground node**. Likewise, the solution, x_i, obtained from equation (2.15) at node v_i, will be referred to as the **potential** for node v_i. The need for fixing an $x_g = 0$ to constrain equation (2.14) may be seen not only from the necessity of grounding a circuit powered only by current sources in order to find unique potentials, but also from the need to provide a boundary condition in order to find a solution to Laplace's equation, of which (2.14) is a combinatorial analog. In our case, the "boundary condition" is that the grounded node is fixed to zero.

Define the $m \times n$ edge-node **incidence matrix** as

$$A_{e_{ij}v_k} = \begin{cases} +1 & \text{if } i = k, \\ -1 & \text{if } j = k, \\ 0 & \text{otherwise,} \end{cases} \tag{2.16}$$

for every vertex v_k and edge e_{ij}, where e_{ij} has been arbitrarily assigned an orientation. As with the Laplacian matrix, $A_{e_{ij}v_k}$ is used to indicate that the incidence matrix is indexed by edge e_{ij} and node v_k. As an operator, A may be interpreted as a combinatorial gradient operator and A^T as a combinatorial divergence [31, 11].

Define the $m \times m$ **constitutive matrix**, C, as the diagonal matrix with the weights of each edge along the diagonal.

As in the familiar continuous setting, the combinatorial Laplacian is equal to the composition of the combinatorial divergence operator with the combinatorial gradient operator, $L = A^T A$. The constitutive matrix defines a weighted inner product of edge values i.e., $\langle y, Cy \rangle$ for a vector of edge values, y [11, 10]. Therefore, the combinatorial Laplacian operator generalizes to the combinatorial Laplace-Beltrami operator via $L = A^T CA$. The case of a uniform (unit) metric, (i.e., equally weighted edges) reduces to $C = I$ and $L = A^T A$. Removing a column of the incidence matrix produces what is known as the **reduced incidence matrix**, A_0 [32].

With this interpretation of the notation used above, the three fundamental equations of circuit theory (Kirchhoff's current and voltage law and Ohm's law) may be written for a grounded circuit as

$$\begin{aligned} A_0^T y &= f \quad \text{(Kirchhoff's Current Law)}, & (2.17)\\ Cp &= y \quad \text{(Ohm's Law)}, & (2.18)\\ p &= A_0 x \quad \text{(Kirchhoff's Voltage Law)}, & (2.19) \end{aligned}$$

for a vector of branch currents, y, current sources, f, and potential drops (voltages), p. Note that there are no voltage sources present in this formulation. These three equations may be combined into the linear system

$$A_0^T CA_0 x = L_0 x = f, \tag{2.20}$$

since $A^T CA = L$ [29].

In summary, the solution to equation (2.15) in the isoperimetric algorithm is provided by the steady state of a circuit where each edge has a conductance equal to the edge weight and each node is attached to a current source of magnitude equal to the degree (i.e., the sum of the conductances of incident edges) of the node. The potentials that are established on the nodes of this circuit are exactly those which are being solved for in equation (2.15). An example of this equivalent circuit is displayed in Figure 2.1.

One final remark on the circuit analogy to (2.15) follows from recalling Maxwell's principle of least dissipation of power: A circuit with minimal

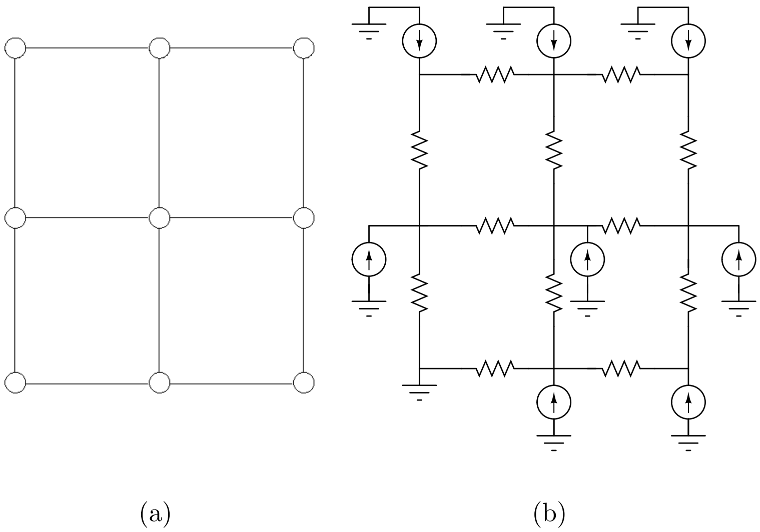

(a) (b)

Figure 2.1: An example of a simple graph and its equivalent circuit. Solving equation (2.15) (using the node in the lower left as ground) for the graph in (a) is equivalent to connecting the circuit in (b) and reading off the potential values at each node.

power dissipation provides a solution to Kirchhoff's current and voltage laws [33]. Explicitly, solving equation (2.15) for x is equivalent to solving the dual equation for $y = CAx$. The power of the equivalent circuit is $P = I^2R = y^TC^{-1}y$ subject to the constraint from Kirchhoff's law that $A^Ty = f$. Therefore, the y found by $y = CAx$ also minimizes the above expression for y [11, 34]. Thus, our approach to minimizing the combinatorial isoperimetric ratio is identical to minimizing the power of the equivalent electrical circuit with the specified current sources and ground [11].

2.3 Algorithmic details

Choosing edge weights

In order to apply the isoperimetric algorithm to partition a graph, the position values (for data clustering) or the image values (for image segmentation) must be encoded on the graph via edge weights. Define the vector of data changes, c_{ij}, as the Euclidean distance between the scalar or vector fields (e.g., coordinates, image RGB channels, image grayscale, etc.) on nodes v_i and v_j. For example, if we represent grayscale intensities defined on each node with vector b, then $c = Ab$. We employ the weighting function [3]

$$w_{ij} = \exp\left(-\beta c_{ij}\right), \tag{2.21}$$

where β represents a parameter we call `scale`. In order to make one choice of β applicable to a wide range of data sets, we have found it helpful to normalize the vector c.

Choosing Partitions from the Solution

The binary definition of x was extended to the real numbers in order to solve (2.15). Therefore, in order to convert the solution, x, to a partition, a subsequent step must be applied (as with spectral partitioning). Conversion of a potential vector to a partition may be accomplished using a threshold. A **cut value** is a value, α, such that $S = \{v_i | x_i \leq \alpha\}$ and $\overline{S} = \{v_i | x_i > \alpha\}$. The partitioning of S and $\overline{S}$ in this way may be referred to as a **cut**. This thresholding operation creates a partition from the potential vector, x. Note that since a connected graph corresponds to an L_0 that is an M-matrix [30], and is therefore monotone, $L_0^{-1} \geq 0$. This result then implies that $x_0 = L_0^{-1}d_0 \geq 0$.

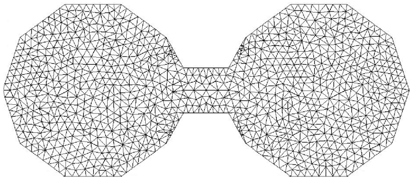


Figure 2.2: Dumbbell graph with uniform weights

Employing the terminology of [35], the standard approaches to cutting the indicator vector in spectral partitioning are to cut based on the median value (the **median cut**) or to choose a threshold such that the resulting partitions have the lowest available isoperimetric ratio (the **ratio cut**). The ratio cut method will clearly produce partitions with a lower isoperimetric ratio than the median cut. Unfortunately, because of the required sorting of x, the ratio cut method requires $\mathcal{O}(n \log(n))$ operations (assuming a bounded degree). The median cut method runs in $\mathcal{O}(n)$ time, but forces the algorithm to produce equal sized partitions, even if a better cut could be realized elsewhere. Despite the required sorting operation for the ratio cut, the operation is still very inexpensive relative to the solution of equation (2.15) for the range of n we focus on (typically 128×128 to 512×512 images). Therefore, we have chosen to employ the ratio cut method.

Ground node

We will demonstrate that, in the image processing context, the ground node may be viewed from an attentional standpoint. However, in the more general graph partitioning context it remains unclear how to choose the ground. Anderson and Morley [36] proved that the spectral radius of L, $\rho(L)$, satisfies $\rho(L) \leq 2d_{\max}$, suggesting that grounding the node of highest degree may have the most beneficial effect on the conditioning of equation (2.15). Empirically, we have found that as long as the ground is not along the ideal cut, a partition with a low isoperimetric ratio is produced.

Figure 2.3 illustrates this principle using the dumbbell shape (in Figure 2.2) discussed in Cheeger's seminal paper [20] on the relationship of the isoperimetric constant and the eigenvalues of the Laplacian on continuous manifolds. The left column (i.e., (a), (c), (e), and (g) in Figure 2.3) shows the potentials, x, solved for using (2.15). The brightest node on the graph represents the ground node. For the rest of the nodes, bright nodes are closer to ground (i.e., have lower potentials) and dark nodes are further from ground. The right column (i.e., (b), (d), (f), and (h) in Figure 2.3) shows the post-threshold function where the ratio cut method has been employed. The top two rows indicate a random selection of ground nodes and the bottom two represent pathological choices of ground nodes. Of the two pathological cases, the third row example (i.e., (e) and (f) in Figure 2.3) uses a ground in the exact center of the neck, while the last row takes ground to be one node over from the center. Although the grounding in the exact center produces a partition that does not resemble the known ideal partition, grounding one node over produces a partition that is nearly the same as the ideal, as shown in the fourth row example (i.e., (g) and (h) in Figure 2.3). This illustrates that the solution is largely independent of the choice of ground node, except in the pathological case where the ground is on the ideal cut. Moreover, it is clear that choosing a ground node in the interior of the balls is better than choosing a point on the neck, which corresponds in some sense to our above rule of choosing the point with maximum degree since a node of high degree will be in the "interior" of a region, or in an area of uniform intensity in the context of image processing.

Solving the System of Equations

Solving equation (2.15) is the computational core of the algorithm. It requires the solution to a large sparse system of symmetric equations where the number of nonzero entries in L will equal $2m$.

Methods for solving a system of equation fall generally into two categories: direct and iterative methods [37, 38, 30]. The former are generally based on Gaussian elimination with partial pivoting while for the latter, the method of conjugate gradients is arguably the best approach. Iterative procedures have the advantage that a partial answer may be obtained at intermediate stages of the solution by specifying a limit on the number of iterations allowed. This feature allows one to trade speed for accuracy in finding a solution. An additional feature of using the method of conjugate gradients to solve equation (2.15) is that it may lend itself to efficient parallelization [39, 40]. In this work, we used the sparse matrix package in $^{\text{TM}}$MATLAB [41] to find direct solutions.

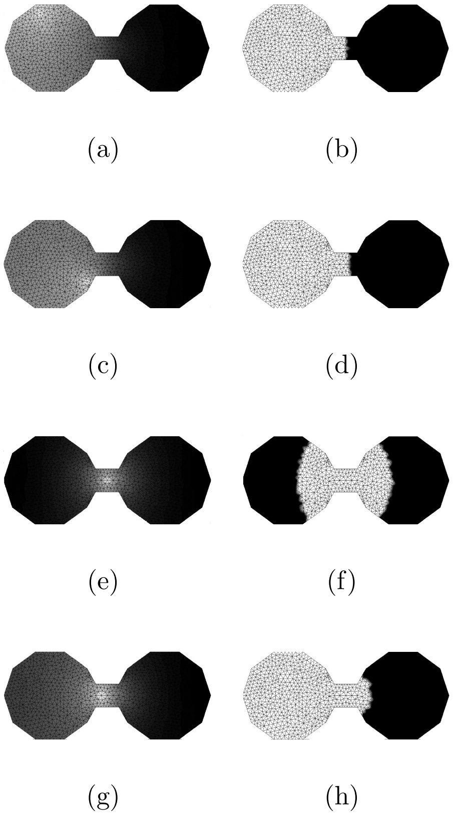


Figure 2.3: An example of the effects on the solution with different choices of ground node for a problem with a trivial optimal partition. The left column shows the potential function (brightest point is ground) for several choices of ground while the right column shows thresholded partitions. Uniform weights ($\beta = 0$) were employed.

Time Complexity

Running time depends mainly on the solution to equation (2.15). A sparse matrix-vector operation depends on the number of nonzero values, which is, in this case, $\mathcal{O}(m)$. If we may assume a constant number of iterations required for the convergence of the conjugate gradients method, the time complexity of solving (2.15) is $\mathcal{O}(m)$. Cutting the potential vector with the ratio cut requires a $\mathcal{O}(n \log(n))$ sort. Combined, the time complexity is $\mathcal{O}(m + n \log n)$. In cases of graphs with bounded degree, then $m \leq n d_{\max}$ and the time complexity reduces to $\mathcal{O}(n \log(n))$. If a constant recursion depth may be assumed (i.e., a consistent number of "objects" in the scene), the time complexity is unchanged.

Summary of the algorithm

Applying the isoperimetric algorithm to data clustering or image segmentation may be described in the following steps:

1. Find weights for all edges using equation (2.21).
2. Build the L matrix (2.8) and d vector.
3. Choose the node of largest degree as the ground node, v_g, and determine L_0 and d_0 by eliminating the row/column corresponding to v_g.
4. Solve equation (2.15) for x_0.
5. Threshold the potentials x at the value that gives partitions corresponding to the lowest isoperimetric ratio.
6. Continue recursion on each segment until the isoperimetric ratio of the subpartitions is larger than the `stop` parameter.

2.4 Relationship to Spectral Partitioning

Building on the early work of Fiedler [42, 43, 44], Alon [45, 46] and Cheeger [20], who demonstrated the relationship between the second smallest eigenvalue of the Laplacian matrix (the **Fiedler value**) for a graph and its isoperimetric constant, spectral partitioning was one of the first successful graph partitioning algorithms [7, 26]. The algorithm partitions a graph by finding the eigenvector corresponding to the Fiedler value, termed the **Fiedler vector**, and cutting the graph based on the value in the Fiedler vector associated with each node. Like isoperimetric partitioning, the output of the spectral partitioning algorithm is a set of values assigned to each node, which require cutting in order to generate partitions.

Spectral partitioning may be used [26] to minimize the isoperimetric ratio of a partition by solving

$$Lz = \lambda z, \tag{2.22}$$

with L defined as above and λ representing the Fiedler value. Since the vector of all ones, r, is an eigenvector corresponding to the smallest eigenvalue (zero) of L, the goal is to find the eigenvector associated with the second smallest eigenvalue of L. Requiring $z^T r = 0$ and $z^T z = n$ may be viewed as additional constraints employed in the derivation of spectral partitioning to circumvent the singularity of L (see, [47] for an explicit formulation of spectral partitioning from this viewpoint). Therefore, one way of viewing the difference between the isoperimetric and the spectral methods is in terms of the choice of an additional constraint that allows one to circumvent the singular nature of the Laplacian L.

In the context of spectral partitioning, the indicator vector z is usually defined as

$$z_i = \begin{cases} -1 & \text{if } v_i \in \overline{S}, \\ +1 & \text{if } v_i \in S, \end{cases} \tag{2.23}$$

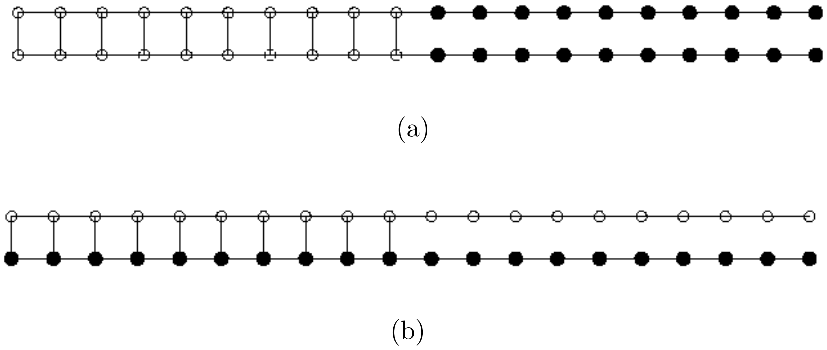
such that z is orthogonal to r, for $|S| = \frac{1}{2}|V|$. The two definitions of the indicator vector (equations (2.7) and (2.23)) are related through $x = \frac{1}{2}(z+r)$. Since r is in the nullspace of L, the definitions are equivalent up to a scaling.

The Ncuts algorithm of Shi and Malik [3] is essentially the spectral partitioning algorithm, except that the authors implicitly choose the metric of [25] to define a combinatorial Laplacian matrix rather than the metric of [24] typically used to define the Laplacian in spectral partitioning. Specifically, the Ncuts algorithm requires the solution of

$$D^{-\frac{1}{2}} L D^{-\frac{1}{2}} z = \lambda z, \tag{2.24}$$

where $D = \text{diag}(d)$. Therefore, although the spectral and Ncuts algorithms produce different results when applied to a specific graph, they share many theoretical properties.

Despite the remarkable success of spectral partitioning [26], it has been pointed out that there are some significant problems. Guattery and Miller



(a)

(b)

Figure 2.4: The "roach" graph (n = 20) illustrated here is a member of a family of graphs for which spectral partitioning is known to fail to produce a partition with low isoperimetric ratio. Uniform weights were used for both algorithms. (a) Solution using isoperimetric algorithm. Ratio = 0.1. (b) Solution using spectral algorithm. Ratio = 0.5.

[48] proposed families of graphs for which spectral partitioning fails to produce the best partition. One of these is the "roach" graph shown in Figure 2.4. This graph will always be partitioned by the spectral method into two symmetrical halves (using the median cut), which yields a suboptimal partition relative to the minimum isoperimetric ratio criterion. For a roach with an equal number of "body" and "antennae" segments, the spectral algorithm will always produce a partition with $|\partial S| = \Theta(n)$ (where $\Theta()$ is the function of [49]) instead of the constant cut set of two edges obtained by cutting the antennae from the body. Teng and Spielman [35] demonstrated that the spectral approach may be made to correctly partition the roach graph if additional processing is performed. The partitions obtained from the spectral and isoperimetric algorithms when applied to the roach graph are compared in Figure 2.4. The solution for the spectral method was obtained from the MESHPART toolbox written by Gilbert, Miller and Teng [50]. This example demonstrates that the isoperimetric algorithm performs in a fundamentally different manner from the spectral method and, at least in this case, outperforms it significantly.

A second difference is that the isoperimetric method requires the solution of a sparse linear system rather than the solution to the eigenvalue problem required by spectral methods of image segmentation [3, 5, 4]. The Lanczos algorithm provides an excellent method for approximating the eigenvectors corresponding to the smallest or largest eigenvalues of a matrix with a time complexity comparable to the conjugate gradient method of solving a sparse system of linear equations [37]. However, solution to the eigenvector problem is less stable to minor perturbations of the matrix than the solution to a system of linear equations, if the desired eigenvector corresponds to an eigenvalue that is very close to other eigenvalues (see, [37]). In fact, for graphs in which the Fiedler value has algebraic multiplicity greater than one the eigenvector problem is degenerate and the Lanczos algorithm may converge to any vector in the subspace spanned by the Fiedler *vectors* (if it converges at all). A square lattice with uniform weights is an example of a graph for which the Fiedler value has algebraic multiplicity greater than unity, as is the fully connected graph with uniform weights (see Appendix 3). The authors of [51] raise additional concerns about the Lanczos method. Appendix 2 formally compares the sensitivity of the isoperimetric, spectral and Ncuts algorithms to a changing edge weight.

3 Applications

3.1 Clustering applied to examples used by Zahn

When humans view a point cluster, certain groupings immediately emerge. The properties that define this grouping have been described by the Gestalt school of psychology . Unfortunately, these descriptions are not precisely defined and therefore finding an algorithm that can group clusters in the same way has proven very difficult. Zahn used his minimal spanning tree idea to try to capture these Gestalt clusters [1]. To this end, he established a collection of point sets with clear cluster structure (to a human), but which are difficult for a single algorithm to group.

We stochastically generated point clusters to mimic the challenges Zahn issues to automatic clustering algorithms. For a set of points, it is not immediately clear how to choose which nodes are connected by edges. In order to guarantee a connected graph, but still make use of local connections, we generated an edge set from the Delaunay triangulation of the points. Edge weights were generated as a function of Euclidean geometric distance, as in equation (2.21).

The clusters and partitions are shown below in Figure 3.5. Each parti-

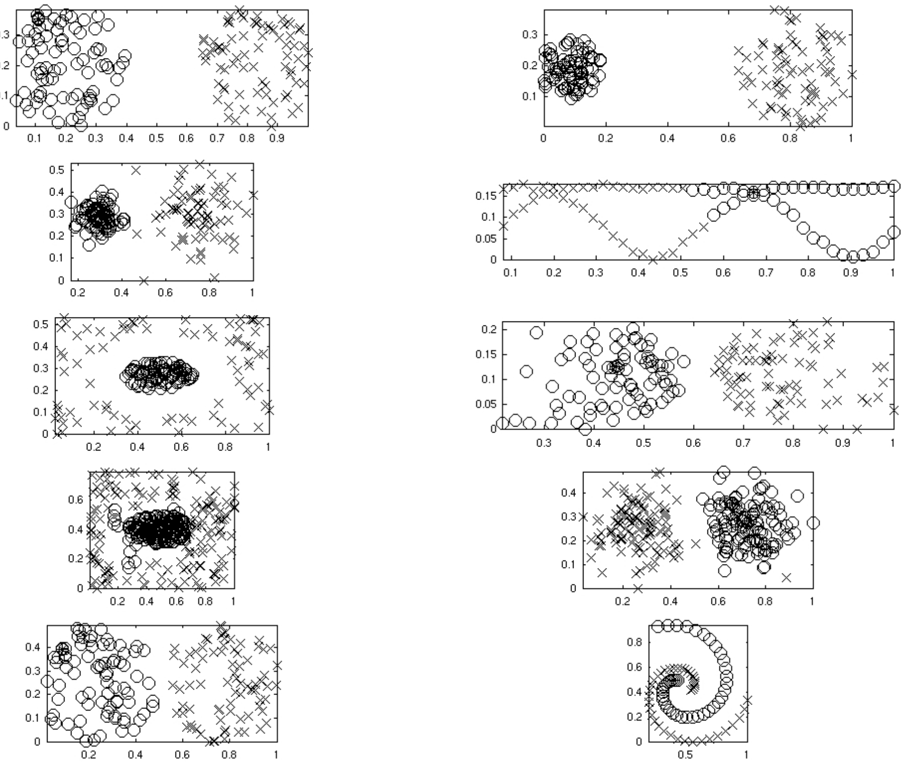


Figure 3.5: An example of partitioning the Gestalt-inspired point set challenges of Zahn using the isoperimetric algorithm. The x's and o's represent points in different partitions. $\beta = 50$.

tion is represented by a symbol, with the 'x's and 'o's indicating the points belonging to the same partition. Partitions were generated using the median cut on a single solution to (2.15). Ground nodes were chosen using the maximum degree rule discussed above. Of these clusters, it is shown in Figure 3.5 that the algorithm performs as desired on all groups except the problem in the second row of the second column that requires grouping into lines.

3.2 Methods of image segmentation

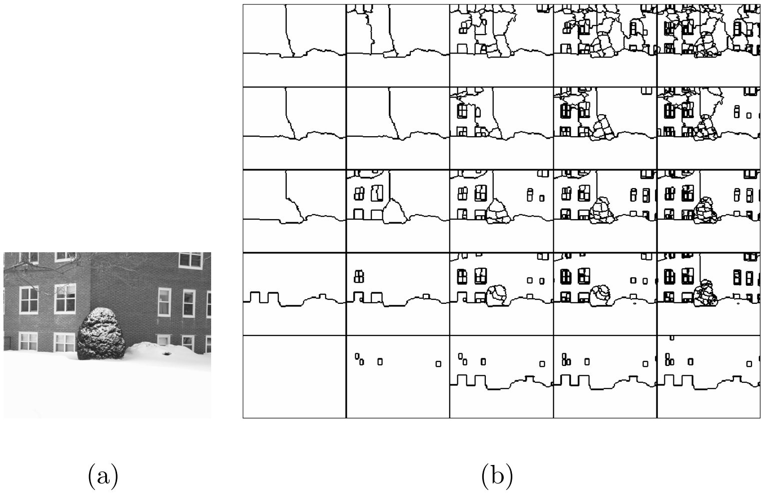
As in the case of point clustering, it is not clear, *a priori*, how to impose a graph structure on an image. Since pixels define the discrete input, a simple choice for nodes is the pixels and their values. Traditional neighborhood connectivity employs a 4-connected or 8-connected topology [52]. Another approach, taken by Shi and Malik [3] is to use a fully connected neighborhood within a parameterized radius of each node. We chose to use a minimal 4-connected topology since the matrix L becomes less sparse as more edges are added to the graph, and a graph with more edges requires more time to solve equation (2.15). Edge weights were generated from intensity values in the case of a grayscale image or from RGB color values in the case of a color image using equation (2.21).

A similar measure of partition quality has been employed by other authors [53, 54] to develop image segmentation algorithms, but a different notion of volume (e.g., the algorithm in [53] is defined only for planar graphs) and different methods for achieving good partitions under this metric of quality separate their work from ours.

The isoperimetric algorithm is controlled by only two parameters: the `scale` parameter β of equation (2.21) and the `stop` parameter used to end the recursion. The `scale` affects how sensitive the algorithm is to changes in feature space (e.g., RGB, intensity), while the `stop` parameter determines the maximum acceptable isoperimetric ratio a partition must generate in order to accept it and continue the recursion. In order to illustrate the dependence of the results on parameterization, a sweep of the two-dimensional parameter space was performed on individual natural images. An example of this parameter-sweep is shown using a natural image, with the `scale` parameter on the vertical and the `stop` parameter on the horizontal (Figure 3.6). It can be seen that the solution is similar over a broad range with respect to changes in `scale` and that the effect of raising the `stop` parameter (i.e., making more partitions admissible) is to generate a greater number of small partitions.

3.3 Completion

Study of the classic Kaniza illusion [55] suggests that humans segment objects based on something beyond perfectly connected edge elements. The isoperimetric algorithm was used to segment the image in Figure 3.7, using only one level of recursion with all nodes corresponding to the black “inducers” removed. In this case, choice of the ground node is important for determining the single bipartition. If the ground node is chosen inside the illusory triangle, the resulting partition is the illusory triangle. However, if the ground is chosen outside, the triangle partition is not produced, but instead a partition that hugs the corner in which the ground is located. In this way, the ground node may be considered as representing something like



(a)

(b)

Figure 3.6: (a) Image used to benchmark the effects of a changing `scale` and `stop` parameter. (b) This tiled figure demonstrates the results of varying the `scale` (vertical) and `stop` (horizontal) parameters when processing the image in (a), showing a large range of stable solutions. `scale` range: 300–30, `stop` range: $1 \times 10^{-5.5}$–$1 \times 10^{-4.5}$.

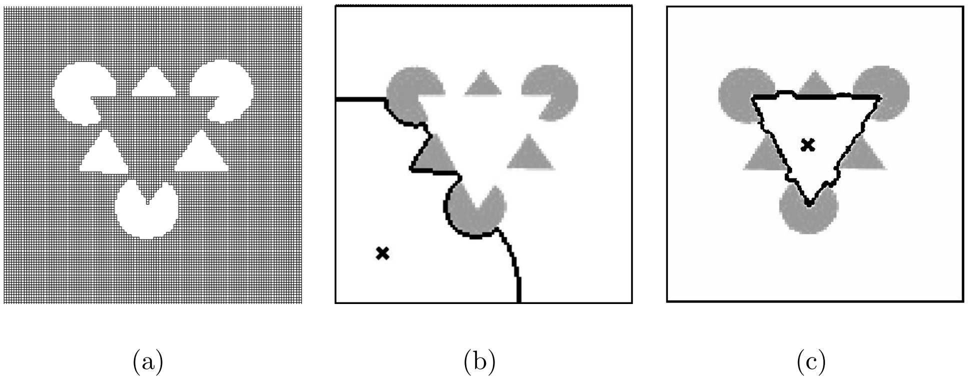


Figure 3.7: The Kaniza triangle illusion with the single bipartition outlined in black and the ground node marked with an ‘x’. (a) The graph being segmented. (b) Isoperimetric partition using a ground point in the corner. (c) Isoperimetric partition using a ground point inside the triangle. Uniform weights ($\beta = 0$) were employed in both cases.

an “attentional” point, since it induces a partition that favors the region of the ground node. However, note that these partitions are compatible with each other, suggesting that the choice of ground may affect only the order in which partitions are found.

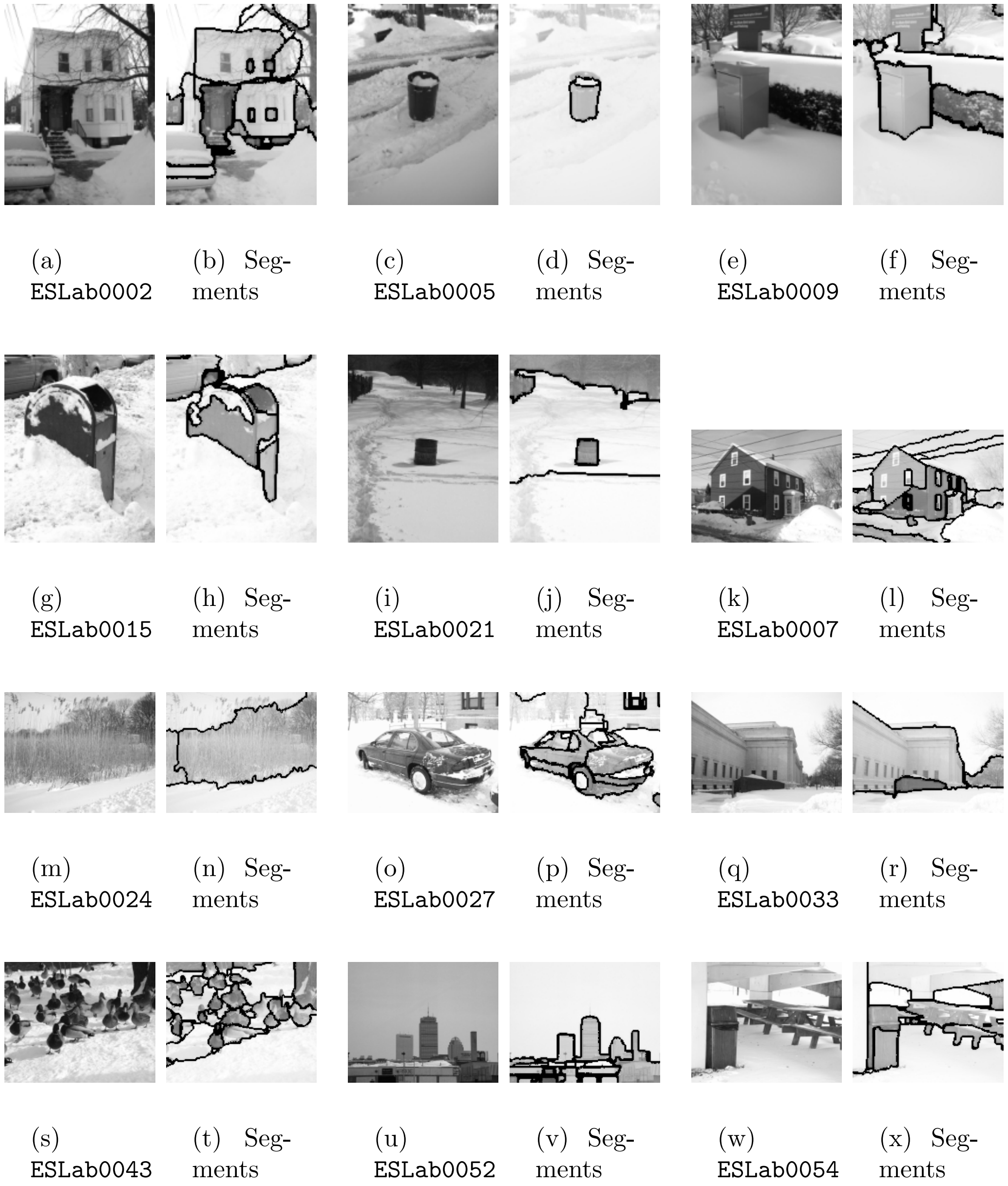
3.4 Segmentation of natural images

Having addressed issues regarding stability and completion, we proceed to examples of the segmentation found by the isoperimetric algorithm when applied to natural images. Examples of the segmentation found by the isoperimetric algorithm for some natural images are displayed in Figure 3.8. All results in the example segmentations were obtained using the same two parameters. It should be emphasized in comparisons of segmentations produced by the Ncuts algorithm that the authors of Ncuts make use of a more fully connected neighborhood as well as fairly sophisticated spatial filtering (e.g., oriented Gabor filters at multiple scales) in order to aid in textural segmentation. The demonstrations with the isoperimetric algorithm used a basic 4-connected topology and no spatial filtering at all. Consequently, the segmentations produced by the isoperimetric algorithm should be expected to perform less well on textural cues. However, for general grayscale images, it appears to perform at least as well as Ncuts, but with increased numerical stability and a speed advantage of more than one order of magnitude (based on our $^{\text{TM}}$MATLAB implementation of both algorithms). Furthermore, because of the implementation (e.g., 4-connected lattice, no spatial filtering), the isoperimetric algorithm makes use of only two parameters, compared to the four basic parameters (i.e., radius, two weighting parameters and the recursion stop criterion) required in the Ncuts paper [3].

The asymptotic (formal) time complexity of Ncuts is roughly the same as the isoperimetric algorithm. Both algorithms have an initial stage in which nodal values are computed that requires approximately $\mathcal{O}(n)$ operations (i.e., via Lanczos or conjugate gradient). Generation of the nodal values is followed in both algorithms by an identical cutting operation. Using the $^{\text{TM}}$MATLAB sparse matrix solver for the linear system required by the isoperimetric algorithm and the Lanczos method ($^{\text{TM}}$MATLAB employs ARPACK [56] for this calculation) to solve the eigenvalue problem required by Ncuts, the time was compared for a 10000×10000 L matrix (i.e., a 100×100 pixel image). Since other aspects of the algorithms are the same (e.g., making weights from the image, cutting the indicator vector, etc.), and because solving for the indicator vector is the main computational hurdle, we only compare the time required to solve for the indicator vector. On a 1.4GHz AMD Athlon with 512K RAM, the time required to approximate the Fiedler vector in equation (2.24) was 7.1922 seconds while application of the direct solver to the isoperimetric partitioning equation (2.15) required 0.5863 seconds. In terms of actual computation time (using $^{\text{TM}}$MATLAB), this result means that solving the crucial equation for the isoperimetric algorithm is more than an order of magnitude faster than solving the crucial equation required by the Ncuts algorithm.

3.5 Stability

Stability of the solution for both the isoperimetric algorithm and the spectral algorithms differs considerably, as does the perturbation analysis for the solution to a system of equations versus the solution to the eigenvector problem [37]. Differentiating equations (2.15) and (2.24) with respect to an edge weight reveals that the derivative of the solution to the spectral (2.22) and Ncuts (2.24) equations is highly dependent on the current Fiedler value, even taking degenerate solutions for some values (see Appendix 2). By contrast, the derivative of the isoperimetric solution has no poles. Instability in spec-



(a) ESLab0002 (b) Segments (c) ESLab0005 (d) Segments (e) ESLab0009 (f) Segments

(g) ESLab0015 (h) Segments (i) ESLab0021 (j) Segments (k) ESLab0007 (l) Segments

(m) ESLab0024 (n) Segments (o) ESLab0027 (p) Segments (q) ESLab0033 (r) Segments

(s) ESLab0043 (t) Segments (u) ESLab0052 (v) Segments (w) ESLab0054 (x) Segments

Figure 3.8: Examples of segmentations produced by the isoperimetric algorithm using the same parameters ($\beta = 95$, `stop` $= 10^{-5}$). Our ^{TM}MATLAB implementation required approximately 10–15 seconds to segment each image. More segmentation results from the same database may by found at `http://eslab.bu.edu/publications/2003/grady2003isoperimetric/`. Images may be obtained from `http://eslab.bu.edu/resources/imageDB/imageDB.php`.

tral methods due to algebraic multiplicity of the Fiedler value is a common problem in implementation of these algorithms (see [53]). This analysis suggests that the Ncuts algorithm may be more unstable to minor changes in an image than the isoperimetric algorithm.

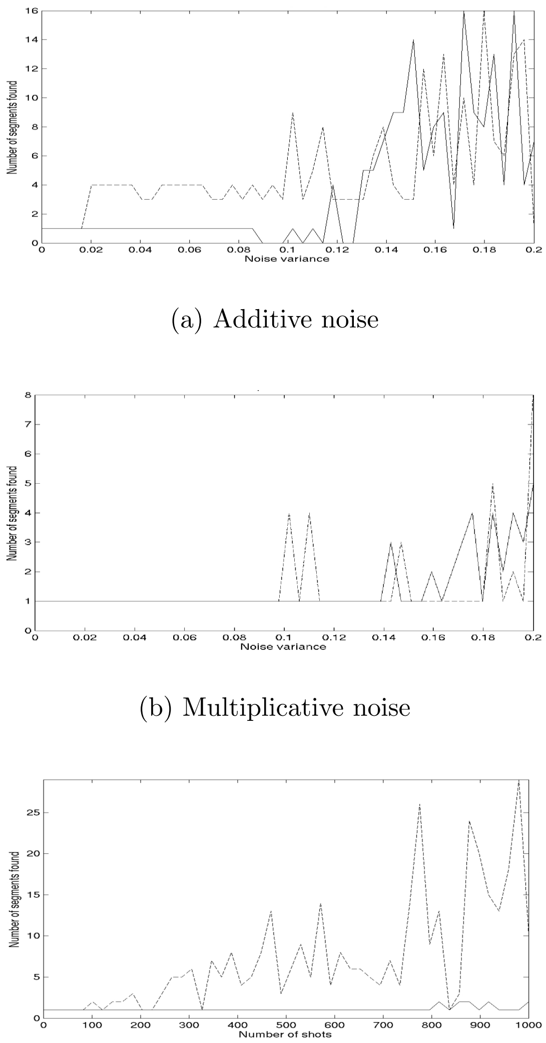
The sensitivity of Ncuts (our implementation) and the isoperimetric algorithm to noise is compared using a quantitative and qualitative measure. First, each algorithm was applied to an artificial image of a white circle on a black background, using a 4-connected lattice topology. Increasing amounts of additive, multiplicative and shot noise were applied, and the number of segments output by each algorithm was recorded. Results of this comparison are recorded in Figure 3.9.

In order to visually compare the result of the segmentation algorithms applied to progressively noisier images, the isoperimetric and Ncuts algorithms were applied to a relatively simple natural image of red blood cells. The isoperimetric algorithm operated on a 4-connected lattice, while Ncuts was applied to an 8-connected lattice, since we had difficulty finding parameters that would cause Ncuts to give a good segmentation of the original image if a 4-connected lattice was used.

In both comparisons, additive, multiplicative, and shot noise were used to test the sensitivity of the two algorithms to noise. The additive noise was zero mean Gaussian noise with variance ranging from 1–20% of the brightest luminance. Multiplicative noise was introduced by multiplying each pixel by a unit mean Gaussian variable with the same variance range as above. Shot noise was added to the image by randomly selecting pixels that were fixed to white. The number of "shots" ranged from 10 to 1,000. The above discussion of stability is illustrated by the comparison in Figure 3.10. Although additive and multiplicative noise heavily degrades the solution found the Ncuts algorithms, the isoperimetric algorithm degrades more gracefully. The presence of even a significant amount of shot noise appears to not seriously disrupt the isoperimetric algorithm, but it significantly affects the convergence of Ncuts to any solution.

4 Conclusion

We have presented a new algorithm for graph partitioning that attempts to find sets with a low isoperimetric ratio. Our algorithm was then applied to the problems of data point clustering and image segmentation. The algorithm

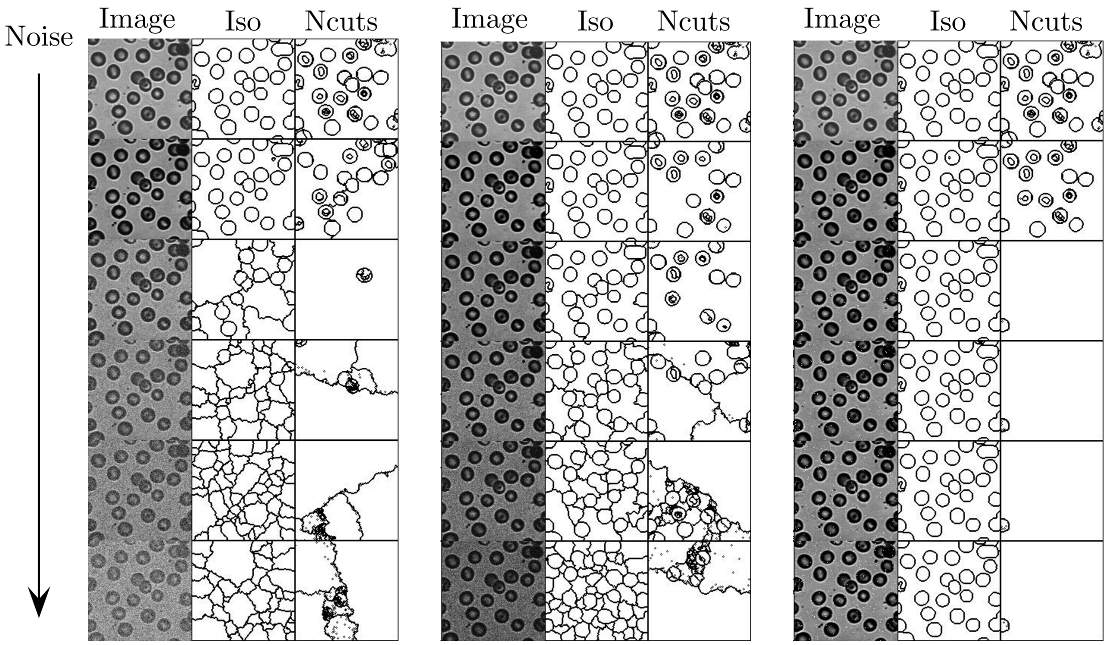


(a) Additive noise

(b) Multiplicative noise

(c) Shot noise

Figure 3.9: Stability analysis relative to additive, multiplicative and shot noise for an artificial image of a white circle on a black background, for which the correct number of segments should be one. The x-axis represents an increasing noise variance for the additive and multiplicative noise, and an increasing number of "shots" for the shot noise. The y-axis indicated the number of segments found by each algorithm. The solid line represents the results of the isoperimetric algorithm and the dashed line represents the results of the Ncuts algorithm. The underlying graph topology was the 4-connected lattice with $\beta = 95$ for the isoperimetric algorithm and $\beta = 35$ for the Ncuts algorithm. Ncuts `stop` criterion $= 10^{-2}$ (relative to the Ncuts criterion) and isoperimetric `stop` criterion $= 10^{-5}$. In all cases, the isoperimetric algorithm outperforms Ncuts, most dramatically in response to shot noise.



(a) Additive noise

(b) Multiplicative noise

(c) Shot noise

Figure 3.10: Stability analysis relative to additive, multiplicative and shot noise. Each row represents an increasing amount of noise of the appropriate type. The top row in each subfigure is the segmentation found for the `blood1.tif` image packaged with $^{\mathrm{TM}}$MATLAB (i.e., zero noise). Each figure is divided into three columns representing the image with noise, isoperimetric segmentation and Ncuts segmentation from left to right respectively. The underlying graph topology was the 4-connected lattice for isoperimetric segmentation and an 8-connected lattice for Ncuts segmentation (due to failure to obtain quality results with a 4-connected lattice) with $\beta = 95$ for the isoperimetric algorithm and $\beta = 35$ for the Ncuts algorithm. Ncuts `stop` criterion $= 5 \times 10^{-2}$ (relative to the Ncuts criterion) and isoperimetric `stop` criterion $= 10^{-5}$. Results were slightly better for additive noise, and markedly better for multiplicative and shot noise. (a) Additive noise. (b) Multiplicative noise. (c) Shot noise.

was compared with Ncuts to demonstrate that it is faster and more stable, while providing visually comparable results with less pre-processing.

Developing algorithms to process a distribution of data on graphs is an exciting area. Many biological sensory units are nonuniformly distributed in space (e.g., vision, somatic sense) with spatial distribution often differing radically between species. The ability to develop algorithms that allow the designer a free hand in choosing the distribution of sensors (or data of any sort) represents a large step over existing algorithms that require a regular, shift-invariant lattice.

These initial findings are encouraging. Since the graph representation is not tied to any notion of dimension, the algorithm applies equally to graph-based problems in N-dimensions as it does to problems in two dimensions. Suggestions for future work are applications to segmentation in space-variant architectures, supervised or unsupervised learning, 3-dimensional segmentation of mesh-based objects, and the segmentation/clustering of other areas that can be naturally modeled with graphs.

Appendix

1 Connectivity

The purpose of this section is to prove that regardless of how a ground is chosen, the partition containing the grounded node (i.e., the set S) must be connected, regardless of how a threshold (i.e., cut) is chosen. The strategy for showing this will be to show that every node has a path to ground such that each node in that path has a monotonically decreasing potential.

Proposition 1 *If the set of vertices, V, is connected then, for any α, the subgraph with vertex set $N \subseteq V$ defined by $N = \{v_i \in V | x_i < \alpha\}$ is connected when x_0 satisfies $L_0 x_0 = f_0$ for any $f_0 \geq 0$.*

This proposition follows directly from proof of the following

Lemma 1 *For every node, v_i, there exists a path to the ground node, v_g, defined by $P_i = \{v_i, v^1, v^2, \dots, v_g\}$ such that $x_i \geq x^1 \geq x^2 \geq \dots \geq 0$, when $L_0 x_0 = f_0$ for any $f_0 \geq 0$.*

Proof: By equation (2.15) each non-grounded node assumes a potential

$$x_i = \frac{1}{d_i} \sum_{e_{ij} \in E} x_j + \frac{f_i}{d_i}, \tag{1.1}$$

i.e., the potential of each non-grounded node is equal to a nonnegative constant added to the (weighted) average potential of its neighbors. Note that (1.1) is a combinatorial formulation of the Mean Value Theorem [57] in the presence of sources.

For any connected subset, $S \subseteq V, v_g \notin S$, denote the set of nodes on the boundary of S as $S_b \subset V$, such that $S_b = \{v_i |\, e_{ij} \in E,\ \exists\, v_j \in S,\ v_i \notin S\}$.

Now, either

1. $v_g \in S_b$, or
2. $\exists\, v_i \in S_b$, such that $x_i \leq \min x_j$, $\forall\, v_j \in S$ by (1.1), since the graph is connected.

Therefore, every node has a path to ground with a monotonically decreasing potential, by induction (i.e., start with $S = \{v_i\}$ and add nodes with a nondecreasing potential until ground is reached). ■

2 Sensitivity Analysis

Previous work in network theory allows for a straightforward analysis of the sensitivity of the isoperimetric, spectral, and normalized cuts algorithms. Here we specifically examine the sensitivity to the edge weights for these three algorithms.

Sensitivity to a single, general parameter, s, is developed in this section. Sensitivity computation for many parameters (e.g., all the weights in a graph) may be obtained efficiently using the adjoint method [58].

2.1 Isoperimetric

Given the vector of degrees, d, the Laplacian matrix, L, and the reduced Laplacian matrix L_0, the isoperimetric algorithm requires the solution to

$$L_0 x_0 = d_0. \tag{2.1}$$

The sensitivity of the solution to equation (2.1) with respect to a parameter s may be determined from

$$L_0 \frac{\partial x_0}{\partial s} = -\frac{\partial L_0}{\partial s} x_0 + \frac{\partial d_0}{\partial s}. \tag{2.2}$$

Since L_0, x_0 are known (for a given solution to equation (2.1) and $\frac{\partial L_0}{\partial s}$ may be determined analytically, $\frac{\partial x_0}{\partial s}$ may be solved for as a system of linear equations (since L_0 is nonsingular) in order to yield the derivative at a point x_0.

2.2 Spectral

The spectral method solves the equation

$$Lx = \lambda_2 x, \tag{2.3}$$

where λ_2 is the Fiedler value. The sensitivity of the solution to equation (2.3) to a parameter s is more complicated, but proceeds in a similar fashion from the equation

$$\frac{\partial L}{\partial s} x + L \frac{\partial x}{\partial s} = \frac{\partial \lambda_2}{\partial s} x + \lambda_2 \frac{\partial x}{\partial s}. \tag{2.4}$$

The term $\frac{\partial \lambda_2}{\partial s}$ may be calculated from the Rayleigh quotient for λ_2 and the chain rule. The Rayleigh quotient is

$$\lambda = \frac{x^T L x}{x^T x}. \tag{2.5}$$

The chain rule determines $\frac{\partial \lambda_2}{\partial s}$ by $\frac{\partial \lambda_2}{\partial s} = \frac{\partial \lambda_2}{\partial x} \frac{\partial x}{\partial s}$. This may be solved by finding $\frac{\partial \lambda_2}{\partial x}$ from the Rayleigh quotient via

$$\frac{\partial \lambda_2}{\partial x} = 2Lx(x^T x)^{-1} - 2x^T Lx(x^T x)^{-2} x. \tag{2.6}$$

Equation (2.6) allows us to solve for $\frac{\partial \lambda_2}{\partial s}$ via equations (2.4) and (2.6)

$$\left(L - \left(\frac{\partial \lambda_2}{\partial x}^T x + \lambda_2 \right) I \right) \frac{\partial x}{\partial s} = \frac{\partial L}{\partial s} x. \tag{2.7}$$

Equation (2.7) also gives a system of linear equations which may be solved for $\frac{\partial x}{\partial s}$ since all the other terms are known or may be determined analytically.

2.3 Normalized Cuts

The normalized cuts algorithm [3] requires the solution to

$$D^{-\frac{1}{2}}LD^{-\frac{1}{2}}x = \lambda_2 x, \tag{2.8}$$

where D is a diagonal vector with $D_{ii} = d_i$. In a similar fashion to the above treatment on the spectral algorithm, the sensitivity of x with respect to a parameter s may be determined using the Rayleigh quotient and the chain rule.

Employing the chain rule, taking the derivative of equation (2.8) with respect to s and rearranging yields

$$\left(D^{-\frac{1}{2}}LD^{-\frac{1}{2}} - \left(\frac{\partial \lambda_2}{\partial x}^T x + \lambda_2\right) I\right)\frac{\partial x}{\partial s} = \\ \left(2\frac{\partial D^{-\frac{1}{2}}}{\partial s}LD^{-\frac{1}{2}} + D^{-\frac{1}{2}}\frac{\partial L}{\partial s}D^{-\frac{1}{2}}\right)x. \tag{2.9}$$

Again, this is a system of linear equations for $\frac{\partial x}{\partial s}$. For Ncuts, the eigenvalue corresponds to $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$ instead of L, so $\frac{\partial \lambda_2}{\partial x}$ must be recomputed from the Rayleigh quotient. The result of this calculation is

$$\frac{\partial \lambda_2}{\partial x} = 2D^{-\frac{1}{2}}LD^{-\frac{1}{2}}x(x^Tx)^{-1} - 2x^TD^{-\frac{1}{2}}LD^{-\frac{1}{2}}x(x^Tx)^{-2}x. \tag{2.10}$$

2.4 Sensitivity to a weight

Using the results above, it is possible to analyze the effect of a specific parameter by finding $\frac{\partial L}{\partial s}$, $\frac{\partial d}{\partial s}$ and $\frac{\partial D^{-\frac{1}{2}}}{\partial s}$ for the specific parameter in question. The value for $\frac{\partial L_0}{\partial s}$ is determined from $\frac{\partial L}{\partial s}$ simply by deleting the row and column corresponding to the grounded node. For a specific weight, w_{ij}, these quantities become

$$\left(\frac{\partial d}{\partial w_{ij}}\right)_{v_i} = \begin{cases} 1 & \text{if } e_{ij} \text{ is incident on } v_i, \\ 0 & \text{otherwise,} \end{cases} \tag{2.11}$$

and

$$\left(\frac{\partial D^{-\frac{1}{2}}}{\partial w_{ij}}\right)_{v_p v_q} = \begin{cases} -\frac{1}{2}d_p^{-\frac{3}{2}} & \text{if } p = q,\ p = i \text{ or } p = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

The matrix $\frac{\partial L}{\partial w_{ij}}$ equals the L matrix of a graph with an edge set reduced to just $E = \{e_{ij}\}$. The degree of node v_i is specified by d_i.

Equations (2.2), (2.4) and (2.9) demonstrate that the derivative of the isoperimetric solution is never degenerate (i.e., the left hand side is always nonsingular for a connected graph), whereas the derivative of the spectral and normalized cuts solutions may be degenerate depending on the current state of the Fiedler vector and value.

3 Fully connected graphs

The isoperimetric algorithm will produce an unbiased solution to equation (2.15) when applied to fully connected graphs with uniform weights. Any set with cardinality equal to half the cardinality of the vertex set and its complement is an isoperimetric set for a fully connected graph with uniform weights. For a uniform edge weight, $w(e_{ij}) = \kappa$ for all $e_{ij} \in E$, the solution, x_0, to equation (2.15) will be $x_i = 1/\kappa$ for all $v_i \in V$. The use of the median or ratio cut method will choose half of the nodes arbitrarily. Although it should be pointed out that using a median or ratio cut to partition a vector of randomly assigned potentials will also produce equal sized (in this case optimal) partitions, the solution to equation (2.15) is unique for a specified ground (in contrast to spectral partitioning or Ncuts, which has $n-1$ solutions) and explicitly gives no node a preference since all the potentials are equal.

Acknowledgments

The authors would like to thank Jonathan Polimeni and Mike Cohen for many fruitful discussions and suggestions.

This work was supported in part by the Office of Naval Research (ONR N00014-01-1-0624).

Bibliography

[1] Charles Zahn, "Graph theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computation*, vol. 20, pp. 68–86, 1971.

[2] Richard Wallace, Ping-Wen Ong, and Eric Schwartz, "Space variant image processing," *International Journal of Computer Vision*, vol. 13, no. 1, pp. 71–90, Sept. 1994.

[3] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, August 2000.

[4] Pietro Perona and William Freeman, "A factorization approach to grouping," in *Computer Vision - ECCV'98. 5th European Conference on Computer Vision. Proceedings*, B. Burkhardt, H.; Neumann, Ed., Freiburg, Germany, 2–6 June 1998, vol. 1, pp. 655–670, Springer-Verlag.

[5] Sudeep Sarkar and Padmanabhan Soundararajan, "Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 504–525, May 2000.

[6] Song Wang and Jeffrey Mark Siskund, "Image segmentation with ratio cut," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 675–690, June 2003.

[7] W.E. Donath and A.J. Hoffman, "Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices," *IBM Technical Disclosure Bulletin*, vol. 15, pp. 938–944, 1972.

[8] Hermann Weyl, “Reparticion de corriente en una red conductora,” *Revista Matemática Hispano-Americans*, vol. 5, no. 6, pp. 153–164, June 1923.

[9] J.P. Roth, “An application of algebraic topology to numerical analysis: On the existence of a solution to the network problem,” *Proceedings of the National Academy of Science of America*, pp. 518–521, 1955.

[10] Franklin H. Branin Jr., “The algebraic-topological basis for network analogies and the vector calculus,” in *Generalized Networks, Proceedings*, Brooklyn, N.Y., April 1966, pp. 453–491.

[11] Gilbert Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, 1986.

[12] G. Sandini, F. Bosero, F. Bottino, and A. Ceccherini, “The use of an anthropomorphic visual sensor for motion estimation and object tracking,” in *Image Understanding and Machine Vision 1989. Technical Digest Series, Vol.14. Conference Edition*, North Falmouth, MA, June 1989, Optical Society of America, AFOSR, vol. 14 of *Technical Digest Series*, pp. 68–72, Optical Society of America.

[13] Alan Rojer and Eric Schwartz, “Design considerations for a space variant visual sensort with complex logarithmic geometry,” in *Proceedings of the Tenth International Conference on Pattern Recognition*, 1990.

[14] Horst D. Simon, “Partitioning of unstructured problems for parallel processing,” *Computing Systems in Engineering*, vol. 2, pp. 135–148, 1991.

[15] Alex Pothen, Horst Simon, and L. Wang, “Spectral nested dissection,” Tech. Rep. CS-92-01, Pennsylvania State University, 1992.

[16] Charles J. Alpert and Andrew B. Kahng, “Recent directions in netlist partitioning: A survey,” *Integration: The VLSI Journal*, vol. 19, pp. 1–81, 1995.

[17] Z. Wu and R. Leahy, “An optimal graph theoretic approach to data clustering: theory and its application to image segmentation,” *IEEE PAMI*, vol. 11, pp. 1101–1113, 1993, 546.

[18] George Karypis and Vipin Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–393, 1998.

[19] Tony F. Chan, John. R. Gilbert, and Shang-Hua Teng, "Geometric spectral partitioning," Tech. Rep. CSL-94-15, Palo Alto Research Center, Xerox Corporation, 1994.

[20] Jefferey Cheeger, "A lower bound for the smallest eigenvalue of the laplacian," in *Problems in Analysis*, R.C. Gunning, Ed., pp. 195–199. Princeton University Press, Princeton, NJ, 1970.

[21] Bojan Mohar, "Isoperimetric numbers of graphs," *Journal of Combinatorial Theory, Series B*, vol. 47, pp. 274–291, 1989, 609.

[22] Bojan Mohar, "Isoperimetric inequalities, growth and the spectrum of graphs," *Linear Algebra and its Applications*, vol. 103, pp. 119–131, 1988.

[23] Fan R. K. Chung, *Spectral Graph Theory*, Number 92 in Regional conference series in mathematics. American Mathematical Society, Providence, R.I., 1997.

[24] Jozef Dodziuk, "Difference equations, isoperimetric inequality and the transience of certain random walks," *Transactions of the American Mathematical Soceity*, vol. 284, pp. 787–794, 1984.

[25] Jozef Dodziuk and W. S Kendall, "Combinatorial laplacians and the isoperimetric inequality," in *From local times to global geometry, control and physics*, K. D. Ellworthy, Ed., vol. 150 of *Pitman Research Notes in Mathematics Series*, pp. 68–74. Longman Scientific and Technical, 1986.

[26] Alex Pothen, Horst Simon, and Kang-Pu Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM Journal of Matrix Analysis Applications*, vol. 11, no. 3, pp. 430–452, 1990.

[27] Russell Merris, "Laplacian matrices of graphs: A survey," *Linear Algebra and its Applications*, vol. 197,198, pp. 143–176, 1994.

[28] George Arfken and Hans-Jurgen Weber, Eds., *Mathematical Methods for Physicists*, Academic Press, 3rd edition, 1985.

[29] Norman Biggs, *Algebraic Graph Theory*, Number 67 in Cambridge Tracts in Mathematics. Cambridge University Press, 1974.

[30] Miroslav Fiedler, *Special matrices and their applications in numerical mathematics*, Martinus Nijhoff Publishers, 1986.

[31] Franklin H. Branin Jr., "The inverse of the incidence matrix of a tree and the formulation of the algebraic-first-order differential equations of an rlc network," *IEEE Transactions on Circuit Theory*, vol. 10, no. 4, pp. 543–544, 1963.

[32] L.R. Foulds, *Graph Theory Applications*, Universitext. Springer-Verlag, New York, 1992.

[33] James Clerk Maxwell, *A Treatise on Electricity and Magnestism*, vol. 1, Dover, New York, 3rd edition, 1991.

[34] D.A. Van Baak, "Variational alternatives of kirchhoff's loop theorem in dc circuits," *American Journal of Physics*, 1998.

[35] Daniel A. Spielman and Shang-Hua Teng, "Spectral partitioning works: Planar graphs and finite element meshes," Tech. Rep. UCB CSD-96-898, University of California, Berkeley, 1996.

[36] William N. Jr. Anderson and Thomas D. Morley, "Eigenvalues of the laplacian of a graph," Tech. Rep. TR 71-45, University of Maryland, October 1971.

[37] Gene Golub and Charles Van Loan, *Matrix Computations*, The John Hopkins University Press, 3rd edition, 1996.

[38] Wolfgang Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Springer-Verlag, 1994.

[39] J. J. Dongarra, I. S. Duff, D. C. Sorenson, and H. A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*, Society for Industrial and Applied Mathematics, Philadelphia, 1991.

[40] Keith Gremban, *Combinatorial preconditioners for sparse, symmetric diagonally dominant linear systems*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, October 1996.

[41] John Gilbert, Cleve Moler, and Robert Schreiber, "Sparse matrices in matlab: Design and implementation," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 1, pp. 333–356, 1992.

[42] Miroslav Fiedler, "Eigenvalues of acyclic matrices," *Czechoslovak Mathematical Journal*, vol. 25, no. 100, pp. 607–618, 1975.

[43] Miroslav Fiedler, "A property of eigenvalues of nonnegative symmetric matrices and its applications to graph theory," *Czechoslovak Mathematical Journal*, vol. 25, no. 100, pp. 619–633, 1975.

[44] Miroslav Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 98, pp. 298–305, 1973.

[45] Norga Alon and V.D. Milman, "λ_1, isoperimetric inequalities for graphs and superconcentrators," *J. of Combinatorial Theory, Series B*, vol. 38, pp. 73–88, 1985.

[46] Norga Alon, "Eigenvalues and expanders," *Combinatorica*, vol. 6, pp. 83–96, 1986, 600.

[47] Y.F. Hu and R.J. Blake, "Numerical experiences with partitioning of unstructured meshes," *Parallel Computing*, vol. 20, pp. 815–829, 1994.

[48] Stephen Guattery and Gary Miller, "On the quality of spectral separators," *SIAM Journal on Matrix Analysis and Applications*, vol. 19, no. 3, pp. 701–719, 1998.

[49] Donald E. Knuth, "Big omicron and big omega and big theta," *SIGACT News*, vol. 8, no. 2, pp. 18–24, April–June 1976.

[50] John R. Gilbert, Gary L. Miller, and Shang-Hua Teng, "Geometric mesh partitioning: Implementation and experiments," *SIAM Journal on Scientific Computing*, vol. 19, no. 6, pp. 2091–2110, 1998.

[51] A.B.J. Kuijaars, "Which eigenvalues are found by the lanczos method?," *SIAM Journal of Matrix Analysis and Applications*, vol. 22, no. 1, pp. 306–321, 2000.

[52] Anil Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Inc., 1989.

[53] Bruce Hendrickson and Robert Leland, "The chaco user's guide," Tech. Rep. SAND95-2344, Sandia National Laboratory, Albuquerque, NM, July 1995.

[54] G. R. Schreiber and O. C. Martin, "Cut size statistics of graph bisection heuristics," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 231–251, 1999.

[55] Mark Fineman, *The Nature of Visual Illusion*, Dover Publications, 1996.

[56] R. B. Lehoucq, D. C. Sorenson, and C. Yang, *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, 1998.

[57] L. Ahlfors, *Complex Analysis*, McGraw-Hill, New York, 1966.

[58] Jiri Vlach and Kishore Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, 2nd edition, 1994.