Husn-Hsien Chang José M. F. Moura "Biomedical Signal Processing,"
ed. Myer Kutz, in Biomedical Engineerng and Design Handbook,
2nd Edition, Volume 1, McGraw Hill. 2010, Chapter 22, pp. 559-579. 
Invited Chapter

# Biomedical Signal Processing

Hsun-Hsien Chang and José M. F. Moura

## I. Introduction

Biomedical signals are observations of physiological activities of organisms, ranging from gene and protein sequences, to neural and cardiac rhythms, to tissue and organ images. Biomedical signal processing aims at extracting significant information from biomedical signals. With the aid of biomedical signal processing, biologists can discover new biology and physicians can monitor distinct illnesses.

Decades ago, the primary focus of biomedical signal processing was on filtering signals to remove noise [1]–[6]. Sources of noise arise from imprecision of instruments to interference of power lines. Other sources are due to the biological systems themselves under study. Organisms are complex systems whose subsystems interact, so the measured signals of a biological subsystem usually contain the signals of other subsystems. Removing unwanted signal components can then underlie subsequent biomedicine discoveries. A fundamental method for noise cancelation analyzes the signal spectra and suppresses undesired frequency components. Another analysis framework derives from statistical signal processing. This framework treats the data as random signals; the processing, e.g. Wiener filtering [6] or Kalman filtering [7], [8], utilizes statistical characterizations of the signals to extract desired signal components.

While these denoising techniques are well established, the field of biomedical signal processing continues to expand thanks to the development of various novel biomedical instruments. The advancement of medical imaging modalities such as ultrasound, magnetic resonance imaging (MRI), and positron emission tomography (PET), enables radiologists to visualize the structure and function of human organs; for example, segmentation of organ structures quantifies organ dimensions [9]. Cellular imaging such as fluorescence tagging and cellular MRI assists biologists

H. H. Chang is with the Harvard Medical School and J. M. F. Moura is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, (email: hsun-hsien.chang@childrens.harvard.edu; moura@ece.cmu.edu).

monitoring the distribution and evolution of live cells [10]; tracking of cellular motion supports modeling cytodynamics [11]. The automation of DNA sequencing aids geneticists to map DNA sequences in chromosomes [12]; analysis of DNA sequences extracts genomic information of organisms [13]. The invention of gene chips enables physicians to measure the expressions of thousands of genes from few blood drops [14]; correlation studies between expression levels and phenotypes unravels the functions of genes [15]. The above examples show that signal processing techniques (segmentation, motion tracking, sequence analysis, and statistical processing) contribute significantly to the advancement of biomedicine.

Another emerging, important need in a number of biomedical experiments is *classification*. In clinics, the goal of classification is to distinguish pathology from normal. For instance, monitoring physiological recordings, clinicians judge if patients suffer from illness [16]; watching cardiac MRI scans, cardiologists identify which region the myocardium experiences failure [17]; analyzing gene sequences of a family, geneticists infer the likelihood that the children inherit disease from their parents [18]. These examples illustrate that automatic decision making can be an important step in medical practice. In particular, incorrect disease identification will not only waste diagnosis resources but also delay treatment or cause loss of patients' lives. Another compelling impact of classification in biosignal processing is illustrated by laboratories where researchers utilize classification methods to identify the category of unknown biomolecules. Due to the high throughput of modern biomolecular experiments such as crystallography, nuclear magnetic resonance (NMR), and electron microscopy, biochemists have instrumental means to determine the molecular structure of vast numbers of proteins. Since proteins with similar structures usually have similar functions, to recognize protein function biochemists resort to exhaustive exploration of their structural similarities [19].

Classification in biomedicine faces several difficulties: (i) Experts need to take a long time to accumulate enough knowledge to reliably distinguish between different related cases, say, normal and abnormal; (ii) Manual classification among these cases is labor intensive and time consuming; (iii) The most formidable challenge occurs when the signal characteristics are not prominent and hence not easily discernible by experts. Automated methods for classification in signal processing hold the promise for overcoming some of these difficulties and to assist biomedical decision making. An automatic classifier can learn from a database the categorial information, replace human operators, and classify indiscernible features without bias. This chapter focuses

on automatic classification algorithms derived from signal processing methods.

### *A. Background on Classification*

The function of a classifier is to automatically partition a given set of biomedical signals into several subsets. For simplicity, we consider binary classification in this chapter, e.g., one subset representing diseased patients and the other representing healthy patients. Classifiers in general fall into two types: *supervised* and *unsupervised* [20], [21]. Supervised classifiers request experts to label a small portion of the data. The classifier then propagates this prior knowledge to the remaining unlabeled data. The labels provided by the experts are a good source of information for the classifier to learn decision making. Often, the experts lack confidence in labeling, which gives rise to uncertainty in the classification results. The worst case is, of course, when the experts mislabel the data, leading the classifier to produce incorrect results. Unlike supervised classification, which is highly sensitive to the prior labels, unsupervised classifiers need no prior class labels. An unsupervised classifier learns by itself the optimal decision rule from the available data and automatically partitions the data set into two. The operator then matches the two subsets to normal and abnormal states. In biomedical practice, it is common that different experts disagree in their labeling priors; in this chapter we focuses on unsupervised classification.

Basically, unsupervised classification collects *similar* signals into the same group such that different groups are as *dissimilar* as possible. Traditional approaches assume that the signals are samples of a mixture of probability models. The classifier estimates from the data the model parameters and then finds which model the signals fit best. Common approaches estimate the model parameters by maximum-likelihood methods when the parameters are deterministic and by Bayes estimation when the parameters are random variables. If the data does not follow the probability models assumed, the performance of the classifier deteriorates. Another disadvantage of these approaches is that they assume the parameters are unconstrained so they do not capture the intrinsic structure of the data.

Graphical models are an alternative to design unsupervised classifiers. Graphical models describe the data by a graph, where features extracted from the signals are mapped to vertices and edges linking vertices account for the statistical correlations or probabilistic dependencies between the features. The entire graph represents the global structure of the whole data set, while the graph edges show the local statistical relations among signals. Although graphical models

visualize the structural information, we need computable measures of the structural information to derive the classifier. Spectral graph theory [22] provides us a tool to derive these measures [23].

### *B. Spectral Graph Based Classifier*

Spectral graph theory studies the properties of graphs by their spectral analysis. The graph structure is uniquely represented by a matrix referred to as *graph Laplacian*. The spectrum of the Laplacian is equivalent to the spectrum of the graph. In the sequel, we exploit the eigenvalues and the eigenfunctions of the graph to design the classifier.

In the spectral graph framework, we formulate in [17] the task of data classification as a problem of graph partitioning [22]. Given a set of signals, the first step is to extract features from the signals and then to describe the feature set as a graph. We treat the feature values as the vertices of a graph, and prescribe a way to assign edges connecting the vertices with high correlated features. Figure 1(a) illustrates the graph representation of a set of five signals; the five circles are the graph vertices denoting the five signals, and the lines are graph edges linking correlated vertices. Graph partitioning is a method that separates the graph into disconnected subgraphs, for example, one representing the subset of abnormal patients and the other representing the subset of healthy patients. Figure 1(b) conceptualizes graph partitioning; we partition the original graph shown in Figure 1(a) by removing two edges, yielding two disjoint subgraphs. The goal in graph partitioning is to find a small as possible subset of edges whose removal will separate out a large as possible subset of vertices. In graph theory terminology, the subset of edges that disjoins the graph is called a *cut*, and the measure to compare partitioned subsets of vertices is the *volume*. Graph partitioning can be quantified by cut-to-volume ratio, termed Cheeger ratio [22]. The optimal graph partitioning seeks the *minimal* Cheeger ratio, which is called the *isoperimetric number* and is also known as the *Cheeger constant* [23] of the graph. Evaluating the Cheeger constant assists the determination of the optimal edge cut.

The determination of the Cheeger constant is a combinatorial problem. We can enumerate all the possible combinations of two subgraphs partitioning the original graph, and then choose the combination with the smallest Cheeger ratio. However, when the number of vertices is very large, the enumeration approach is infeasible. We circumvent this obstacle by adopting an optimization framework. We derive from the Cheeger constant an objective functional to be minimized with respect to the classifier. The minimization of the functional leads to optimal classification.

If there is a complete set of basis functions on the graph, we can represent the classifier by a linear combination of the basis functions. There are various ways to obtain the basis functions, e.g., using the Laplacian operator [24], the diffusion kernel [25], or the Hessian eigenmap [26]. Among these, we choose the Laplacian. The spectrum of the Laplacian operator has been used to obtain upper and lower bounds on the Cheeger constant [22]; these bounds will be helpful in our classifier design. The eigenfunctions of the Laplacian form a basis of the Hilbert space of square integrable functions defined on the graph. Thus, we express the classifier as a linear combination of the Laplacian eigenfunctions. Since the basis is known, the optimal classifier is determined by the coefficients in the linear combination. The classifier can be further approximated as a linear combination of only the *most relevant* basis functions. The approximation reduces significantly the problem from looking for a large number of coefficients to estimating only a few of them. Once we determine the optimal coefficients, the optimal classifier automatically partitions the data set into two parts.

### C. Chapter Organization

The organization of this chapter is as follows. Section II describes how we represent a data set by a graph and introduces the Cheeger constant for graph partitioning. Section III details the optimal classification algorithm in the framework of spectral graph theory. The algorithm design follows our work presented in [17]. In Section IV, we adopt a toy model to illustrate the important concepts in developing the classifier. We also demonstrate the application of the classifier on a contrast-enhanced MRI data of the heart. Finally, Section V concludes this chapter.

## II. Graph Representation of Signals

Given a set of feature vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ extracted from the signals to be classified, we begin with its graph representation. There are several ways to achieve graph representation, for example, based on statistical correlation [17] or nearest neighbor [24]. We follow the former approach in this chapter.

### A. Weighted Graph Representation

A graph $G(V, E)$ that describes the data $X$ has a set $V$ of vertices and a set $E$ of edges linking the vertices. In this study, we assume that the graph $G$ is connected; that is, there are

no disjoint subgraphs before running the graph partitioning algorithm. In $G$, each vertex $v_i \in V$ corresponds to a feature vector $\mathbf{x}_i$. We next assign edges connecting the vertices. In the graph representation of $X$, the vertices with high possibility of being drawn from the same class are linked together. The main strategy is to connect vertices with similar feature vectors, because feature vectors in the same class have the same values up to noise. To account for the similarity among feature vectors, we need a metric $\rho_{ij} = \rho(\mathbf{x}_i, \mathbf{x}_j)$ between the features $\mathbf{x}_i$, $\mathbf{x}_j$ of vertices $v_i$, $v_j$. The choice of metric $\rho$ depends on the practical application. A simple choice is the Euclidean distance [24], i.e.,

$$\rho_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|. \tag{1}$$

Another metric that considers randomness in signals is the Mahalanobis distance, [17], [20]

$$\rho_{ij} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \Sigma_{ij}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}\,, \tag{2}$$

where $\Sigma_{ij}$ is the covariance matrix between $\mathbf{x}_i$ and $\mathbf{x}_j$. Other metrics include mutual information or $t$-statistics. When the distance $\rho_{ij}$ is below a predetermined threshold $\tau_\rho$, the vertices $v_i$, $v_j$ are connected by an edge; otherwise, they remain disconnected.

Since connected pairs of vertices do not have the same distance, we consider *weighted* edges by using a weight function on the edges. Belkin and Niyogi [24] and Coifman *et al.* [25] suggest Gaussian kernel for computing the weights $W_{ij}$ on edges $e_{ij}$:

$$W_{ij} = \begin{cases} \exp\left(-\frac{\rho_{ij}^2}{\sigma^2}\right), & \text{if there is an edge } e_{ij} \\ 0\,, & \text{if there is no edge } e_{ij} \end{cases}, \tag{3}$$

where $\sigma$ is the Gaussian kernel parameter. The larger the value of $\sigma$ is, the more weight far-away vertices will exert on the weighted graph. The weight $W_{ij}$ is large when the signals of two linked vertices $v_i$, $v_j$ are similar.

The weighted graph is equivalently represented by its $N \times N$ *weighted adjacency matrix* $\mathbf{W}$ whose elements $W_{ij}$ are the edge weights in equation (3). Note that the matrix $\mathbf{W}$ has a zero diagonal because we do not allow the vertices to be self-connected; it is symmetric since $W_{ij} = W_{ji}$.

### *B. Graph Partitioning and the Cheeger Constant*

In graph terms, classification means the division of the graph $G(V, E)$ into two disjoint subgraphs. The task is to find a subset $E_0$ of edges, called an *edge cut*, such that removing this cut

separates the graph $G(V, E)$ into two disconnected subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, and $E = E_0 \cup E_1 \cup E_2$.

In the framework of spectral graph theory, we define an *optimal* edge cut by looking for the Cheeger constant, [22] , $\Gamma(V_1)$ of the graph,

$$\Gamma(V_1) = \min_{V_1 \subset V} \frac{|E_0(V_1, V_2)|}{\min\{\text{vol}(V_1), \text{vol}(V_2)\}} \,. \tag{4}$$

Without loss of generality, we can always assume that $\text{vol}(V_1) \leq \text{vol}(V_2)$, so the Cheeger constant becomes

$$\Gamma(V_1) = \min_{V_1 \subset V} \frac{|E_0(V_1, V_2)|}{\text{vol}(V_1)} \,. \tag{5}$$

In equations (4) and (5), $|E_0(V_1, V_2)|$ is the sum of the edge weights in the cut $E_0$:

$$|E_0(V_1, V_2)| = \sum_{v_i \in V_1, v_j \in V_2} W_{ij} \,. \tag{6}$$

The volume $\text{vol}(V_1)$ of $V_1$ is defined as the sum of the vertex degrees in $V_1$:

$$\text{vol}(V_1) = \sum_{v_i \in V_1} d_i \,, \tag{7}$$

where the degree $d_i$ of the vertex $v_i$ is defined as

$$d_i = \sum_{v_j \in V} W_{ij} \,. \tag{8}$$

The volume $\text{vol}(V_2)$ of $V_2$ is defined in a similar way to volume $\text{vol}(V_1)$, as in (7).

We can rewrite the vertex degree $d_i$ by considering the vertex $v_j$ in either $V_1$ or $V_2$; i.e.,

$$d_i = \sum_{v_j \in V_1} W_{ij} + \sum_{v_j \in V_2} W_{ij} \,. \tag{9}$$

Assuming that the vertex $v_i$ is in $V_1$, the second term in equation (9) is the contribution of $v_i$ made to the edge cut $|E_0(V_1, V_2)|$. Taking into account all the vertices in $V_1$, we reexpress the edge cut:

$$|E_0(V_1, V_2)| = \sum_{v_i \in V_1} \sum_{v_j \in V_2} W_{ij} \tag{10}$$

$$= \sum_{v_i \in V_1} \left( d_i - \sum_{v_j \in V_1} W_{ij} \right) \,. \tag{11}$$

Equation (11) can be written in matrix form. We introduce an indicator vector $\chi$ for $V_1$ whose elements are defined as

$$\chi_i = \begin{cases} 1, & \text{if } v_i \in V_1 \\ 0, & \text{if } v_i \in V_2 \end{cases} . \tag{12}$$

It follows that the edge cut (11) is

$$|E_0(V_1, V_2)| = \chi^T \mathbf{D} \chi - \chi^T \mathbf{W} \chi \tag{13}$$

$$= \chi^T \mathbf{L} \chi \,, \tag{14}$$

where $\mathbf{D} = \text{diag}\,(d_1, d_2, \cdots, d_N)$ is a diagonal matrix of vertex degrees, and

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{15}$$

is the Laplacian of the graph, see [22]. The Laplacian $\mathbf{L}$ is symmetric, because $\mathbf{D}$ is diagonal and $\mathbf{W}$ is symmetric. Further, it can be shown that $\mathbf{L}$ is positive semidefinite since the row sums of $\mathbf{L}$ are zeros.

The volume vol$(V_1)$ can be expressed in terms of the indicator vector $\chi$ for $V_1$, see (12),

$$\text{vol}(V_1) = \sum_{v_i \in V_1} d_i = \chi^T \mathbf{d} \,, \tag{16}$$

where $\mathbf{d}$ is the column vector collecting all the vertex degrees $d_i$. Replacing (14) and (16) into the Cheeger constant (5), we write the Cheeger constant in terms of the indicator vector $\chi$:

$$\Gamma(\chi) = \min_{\chi} \frac{\chi^T \mathbf{L} \chi}{\chi^T \mathbf{d}} \,. \tag{17}$$

The optimal graph partitioning corresponds to the optimal indicator vector

$$\hat{\chi} = \arg\min_{\chi} \frac{\chi^T \mathbf{L} \chi}{\chi^T \mathbf{d}} \,. \tag{18}$$

### *C. Objective Functional for Cheeger Constant*

In equation (17), intuitively, minimizing the numerator $\chi^T \mathbf{L} \chi$ and maximizing the denominator $\chi^T \mathbf{d}$ would minimize the Cheeger ratio. Putting a minus sign in front of $\chi^T \mathbf{d}$, we determine the optimal indicator vector $\hat{\chi}$ by minimizing the objective functional

$$Q(\chi) = \chi^T \mathbf{L} \chi - \beta \chi^T \mathbf{d} \,, \tag{19}$$

where $\beta$ is the weight. The objective $Q(\chi)$ is convex, because the graph Laplacian $\mathbf{L}$ is positive semidefinite. In addition, the second term $0 \leq \chi^T \mathbf{d} \leq \text{vol}(V)$ is finite, so the minimizer $\hat{\chi}$ exists.

Since the indicator $\chi_i$ is either 1 or 0 at vertex $v_i$, see equation (12), there are $2^N$ candidate indicator vectors. When the number $N$ of signals is large, it is not computationally feasible to determine the Cheeger constant, or to minimize the objective $Q(\chi)$, by enumerating all the candidate indicator vectors. The next section describes a method to avoid this combinatorial obstacle.

## III. Optimal Classification Algorithm

### A. Spectral Analysis of the Graph Laplacian $\mathbf{L}$

The spectral decomposition of the graph Laplacian $\mathbf{L}$, which is defined in equation (15), gives the eigenvalues $\{\lambda_n\}_{n=0}^{N-1}$ and eigenfunctions $\{\phi^{(n)}\}_{n=0}^{N-1}$. By convention, we index the eigenvalues starting with 0 and in ascending order. Because the Laplacian $\mathbf{L}$ is symmetric and positive semidefinite, its spectrum $\{\lambda_n\}$ is real and nonnegative and its rank is $N-1$. In the framework of spectral graph theory [22], the eigenfunctions $\{\phi^{(n)}\}$ assemble a complete set and span the Hilbert space of square integrable functions on the graph. Hence, we can express any square integrable function on the graph as a linear combination of the basis functions $\{\phi^{(n)}\}$. The domain of the eigenfunctions are vertices, so the eigenfunctions $\{\phi^{(n)}\}$ are discrete and are represented by vectors

$$\forall n = 0, \cdots, N-1, \quad \phi^{(n)} = [\phi_1^{(n)}, \phi_2^{(n)}, \cdots, \phi_N^{(n)}]^T. \tag{20}$$

Note that the subscripts $i \in I = \{1, 2, \cdots, N\}$ correspond to the indices denoting the vertices $v_i$.

We list here the properties of the spectrum of the Laplacian (see [22] for additional details) that will be exploited to develop the classification algorithm:

1) For a *connected* graph, there is only one zero eigenvalue $\lambda_0$, and the spectrum can be ordered as

$$0 = \lambda_0 < \lambda_1 \leq \cdots \leq \lambda_{N-1}\,. \tag{21}$$

The zeroth eigenvector $\phi^{(0)}$ is constant, i.e.,

$$\phi^{(0)} = \alpha[1, 1, \cdots, 1]^T\,, \tag{22}$$

where $\alpha = \frac{1}{\sqrt{N}}$ is the normalization factor for $\phi^{(0)}$.

2) The eigenvectors $\phi^{(n)}$ with nonzero eigenvalues have zero averages, namely

$$\sum_{i=1}^{N} \phi_i^{(n)} = 0 \,. \tag{23}$$

The low order eigenvectors correspond to low frequency harmonics.

3) For a *connected* graph, the Cheeger constant $\Gamma$ defined by (17) is upper and lower bounded by the following inequality:

$$\frac{2}{\text{vol}(V)} \leq \Gamma < \sqrt{2\lambda_1} \,, \tag{24}$$

where $\text{vol}(V) = \mathbf{1}^T \mathbf{d}$.

Without loss of generality, we can assume that these spectral properties hold in our study.

### *B. Classifier*

The classifier $\mathbf{c}$ partitioning the graph vertex set $V$ into two classes $V_1$ and $V_2$ is defined as

$$c_i = \begin{cases} 1, & \text{if } v_i \in V_1 \\ -1, & \text{if } v_i \in V_2 \end{cases} . \tag{25}$$

Utilizing spectral graph analysis, we express the classifier in terms of the eigenbasis $\{\phi^{(n)}\}$

$$\mathbf{c} = \sum_{n=0}^{N-1} a_n \phi^{(n)} \,, \tag{26}$$

where $a_n$ are the coordinates of the eigen representation. Note that the zeroth eigenvector $\phi^{(0)}$ is a constant that can be ignored in designing the classifier $\mathbf{c}$. Thus, the classifier expression becomes

$$\mathbf{c} = \sum_{n=1}^{N-1} a_n \phi^{(n)} = \Phi \mathbf{a} \,, \tag{27}$$

where

$$\mathbf{a} = [a_1, a_2, \cdots, a_{N-1}]^T \tag{28}$$

stacks the classifier coefficients, and $\Phi$ is a matrix collecting the vector of the eigen-basis

$$\Phi = \left[\phi^{(1)}, \phi^{(2)}, \cdots, \phi^{(N-1)}\right] \,. \tag{29}$$

The design of the optimal classifier $\mathbf{c}$ becomes now the problem of estimating the linear combination coefficients $a_n$.

### C. Objective Functional for Classification

In equation (17), the Cheeger constant is expressed in terms of the set indicator vector $\chi$ that takes 0 or 1 values. On the other hand, the classifier $\mathbf{c}$ defined in (25) takes $\pm 1$ values. We relate $\chi$ and $\mathbf{c}$ by the standard Heaviside function $\mathcal{H}(x)$ defined by

$$\mathcal{H}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} . \tag{30}$$

Hence, the indicator vector $\chi = [\chi_1, \chi_2, \cdots, \chi_N]^T$ for the set $V_1$ is given by

$$\chi_i = \mathcal{H}(c_i) . \tag{31}$$

In equation (31), the indicator $\chi$ is a function of the classifier $\mathbf{c}$ using the Heaviside function $\mathcal{H}$. Furthermore, by (27), the classifier $\mathbf{c}$ is parameterized by the coefficient vector $\mathbf{a}$, so the objective functional $Q$ is parameterized by this vector $\mathbf{a}$, i.e.,

$$Q(\mathbf{a}) = \chi(\mathbf{c}(\mathbf{a}))^T \mathbf{L} \chi(\mathbf{c}(\mathbf{a})) - \beta \chi(\mathbf{c}(\mathbf{a}))^T \mathbf{d} . \tag{32}$$

Minimizing $Q$ with respect to the vector $\mathbf{a}$ gives the optimal coefficient vector $\hat{\mathbf{a}}$, which leads to the optimal classifier $\hat{\mathbf{c}} = \Phi \hat{\mathbf{a}}$. Using the eigenbasis to represent the classifier transforms the problem of combinatorial optimization in (19) to estimating the real-valued coefficient vector $\mathbf{a}$ in (32).

When the classifier is represented by the full eigenbasis, there are $N-1$ parameters $a_n$ to be estimated. To avoid estimating too many parameters, we relax the classification function to be a smooth function, which means that only the first $p$ harmonics of the eigenbasis are kept in (27). The classifier $\mathbf{c}$ is now

$$\mathbf{c} = \sum_{n=1}^{p} a_n \phi^{(n)} = \Phi \mathbf{a} , \tag{33}$$

where we redefine

$$\mathbf{a} = [a_1, a_2, \cdots, a_p]^T \tag{34}$$

and

$$\Phi = \left[\phi^{(1)}, \phi^{(2)}, \cdots, \phi^{(p)}\right] . \tag{35}$$

The estimation of the $N-1$ parameters in (27) is reduced to the estimation of the $p \ll (N-1)$ parameters in (33). As long as $p$ is chosen small enough, the latter is more numerically tractable than the former.

The weighting parameter $\beta$ is unknown in the objective functional (19). If we knew the Cheeger constant $\Gamma$, we could set $\beta = \Gamma$ and the objective function would be

$$Q(\chi) = \chi^T \mathbf{L}\chi - \Gamma \chi^T \mathbf{d}\,, \tag{36}$$

whose solution is $Q(\chi) = 0$, see (17). However, we cannot set $\beta = \Gamma$ beforehand, since the Cheeger constant $\Gamma(\hat{\chi})$ is dependent on the unknown optimal indicator vector $\hat{\chi}$.

We can reasonably predetermine $\beta$ by using spectral properties of the graph Laplacian: The upper and lower bounds of the Cheeger constant are related to the first nonzero eigenvalue $\lambda_1$ and the graph volume $\text{vol}(V)$, see equation (24). The bounds restrain the range of values for the weight $\beta$. For simplicity, we set $\beta$ to the average of the Cheeger constant's upper and lower bounds,

$$\beta = \frac{1}{2}\left(\frac{2}{\text{vol}(V)} + \sqrt{2\lambda_1}\right)\,. \tag{37}$$

### *D. Minimization Algorithm*

Taking the gradient of $Q(\mathbf{a})$ with respect to the vector $\mathbf{a}$ yields

$$\frac{\partial Q}{\partial \mathbf{a}} = 2\left(\frac{\partial \chi^T}{\partial \mathbf{a}}\right)\mathbf{L}\chi - \beta\left(\frac{\partial \chi^T}{\partial \mathbf{a}}\right)\mathbf{d}\,, \tag{38}$$

where the matrix $\frac{\partial \chi^T}{\partial \mathbf{a}}$ is

$$\frac{\partial \chi^T}{\partial \mathbf{a}} = \left[\frac{\partial \chi_1}{\partial \mathbf{a}}, \frac{\partial \chi_2}{\partial \mathbf{a}}, \cdots, \frac{\partial \chi_N}{\partial \mathbf{a}}\right] \tag{39}$$

$$= \begin{bmatrix} \frac{\partial \chi_1}{\partial a_1} & \frac{\partial \chi_2}{\partial a_1} & \cdots & \frac{\partial \chi_N}{\partial a_1} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \chi_1}{\partial a_p} & \frac{\partial \chi_2}{\partial a_p} & \cdots & \frac{\partial \chi_N}{\partial a_p} \end{bmatrix}. \tag{40}$$

Using the chain rule, the entries $\left(\frac{\partial \chi^T}{\partial \mathbf{a}}\right)_{mn}$ are

$$\left(\frac{\partial \chi^T}{\partial \mathbf{a}}\right)_{mn} = \frac{\partial \chi_n}{\partial a_m} \tag{41}$$

$$= \frac{\partial \chi_n}{\partial c_n}\frac{\partial c_n}{\partial a_m} \tag{42}$$

$$= \delta(c_n)\frac{\partial \sum_{j=1}^{p} a_j \phi_n^{(j)}}{\partial a_m} \tag{43}$$

$$= \delta(c_n)\phi_n^{(m)}\,. \tag{44}$$

In (43), $\delta(x)$ is the delta (generalized) function defined as the derivative of the Heaviside function $\mathcal{H}(x)$.

To facilitate numerical implementation, we use the *regularized* Heaviside function $\mathcal{H}_\epsilon$ and the regularized delta function $\delta_\epsilon$; they are defined, respectively, as

$$\mathcal{H}_\epsilon(x) = \frac{1}{2}\left[1 + \frac{2}{\pi}\arctan\left(\frac{x}{\epsilon}\right)\right] , \tag{45}$$

$$\delta_\epsilon(x) = \frac{d\mathcal{H}_\epsilon(x)}{dx} = \frac{1}{\pi}\left(\frac{\epsilon}{\epsilon^2 + x^2}\right) . \tag{46}$$

Using the regularized delta function, the explicit expression of $\frac{\partial \chi^T}{\partial \mathbf{a}}$ is

$$\frac{\partial \chi^T}{\partial \mathbf{a}} = \begin{bmatrix} \delta_\epsilon(c_1)\phi_1^{(1)} & \delta_\epsilon(c_2)\phi_2^{(1)} & \cdots & \delta_\epsilon(c_N)\phi_N^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ \delta_\epsilon(c_1)\phi_1^{(p)} & \delta_\epsilon(c_2)\phi_2^{(p)} & \cdots & \delta_\epsilon(c_N)\phi_N^{(p)} \end{bmatrix} \tag{47}$$

$$= \Phi^T \Delta , \tag{48}$$

where we define the diagonal matrix

$$\Delta = \text{diag}\left(\delta_\epsilon(c_1), \delta_\epsilon(c_2), \cdots, \delta_\epsilon(c_N)\right) . \tag{49}$$

Substituting (48) into (38), the gradient of the objective $Q$ has the compact form

$$\frac{\partial Q}{\partial \mathbf{a}} = 2\Phi^T \Delta \mathbf{L} \chi - \beta \Phi^T \Delta \mathbf{d} . \tag{50}$$

The optimal coefficient vector $\hat{\mathbf{a}}$ is obtained by looking for

$$\frac{\partial Q}{\partial \mathbf{a}} = 0. \tag{51}$$

We have to solve this minimization numerically, because both the matrix $\Delta$ and the vector $\chi$ depend on the unknown coefficient vector $\mathbf{a}$. We adopt the gradient descent algorithm to iteratively find the solution $\hat{\mathbf{a}}$. The classifier $\hat{\mathbf{c}}$ is then determined by

$$\hat{\mathbf{c}} = \Phi \hat{\mathbf{a}} . \tag{52}$$

The vertices with indicators $\hat{\chi}_i = \mathcal{H}(\hat{c}_i) = 1$ correspond to class $V_1$ and $\hat{\chi}_i = \mathcal{H}(\hat{c}_i) = 0$ correspond to class $V_2$.

### *E. Algorithm Summary*

The development of the classifier involves two major steps: graph representation and classification. We summarize them in Algorithms 1 and 2, respectively.

**Algorithm 1** Graph representation algorithm

**procedure** GRAPHREP($X$)
  Index all the signal features by a set of integers $I = \{1, \cdots, N\}$
  Initialize $\mathbf{W}$ as an $N \times N$ zero matrix
  **for all** $i \neq j \in I$ **do**
    Compute metric $\rho_{ij}$ by (1) or (2)
    **if** $\rho_{ij} < \tau_\rho$ **then**
      $W_{ij} \leftarrow$ Compute edge weight $W_{ij}$ by (3)
    **end if**
  **end for**
  **return** $\mathbf{W}$
**end procedure**

**Algorithm 2** Classification algorithm

**procedure** CLASSIFIER($\mathbf{W}$)
  Compute graph Laplacian $\mathbf{L}$ by (15)
  Eigendecompose $\mathbf{L}$ to obtain $\{\lambda_n\}$ and $\{\phi^{(n)}\}$
  Keep the smallest $p$ nonzero eigenvalues and corresponding eigenvectors
  Compute the weighting parameter $\beta$ by (37)
  Initialize the classifier coefficient vector $\mathbf{a} = \mathbf{1}$ and the objective $Q = \infty$
  **repeat**
    $\mathbf{c} \leftarrow$ Compute classifier $\mathbf{c}$ by (33)
    $\chi \leftarrow$ Compute indicator vector $\chi$ by (31)
    $Q \leftarrow$ Compute objective $Q$ by (32)
    $\mathbf{a} \leftarrow$ Compute $\mathbf{a} - \frac{\partial Q}{\partial \mathbf{a}}$ by (50)
  **until** $\frac{\partial Q}{\partial \mathbf{a}} = 0$
  **return** $\chi$
**end procedure**

## IV. EXAMPLES

### A. Toy Model

We use a toy model to illustrate all the concepts introduced in the preceding sections. Suppose that a given set of 3D features have values $\mathbf{x}_1 = [0, 0, 1]^T$, $\mathbf{x}_2 = [0, 0.5, 1]^T$, $\mathbf{x}_3 = [0.7, 0.1, 0.9]^T$, and $\mathbf{x}_4 = [0.7, 0, 1]^T$.

**Graph Representation.** The feature vectors $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ correspond to four nodes $\{v_1, v_2, v_3, v_4\}$, respectively, in their graph representation, shown in Figure 2(a). The next task is to assign edges to link the vertices. For simplicity, we choose the Euclidean distance as the metric, see equation (1). Table I summarizes the distance $\rho_{ij}$ between every pair of vertices. When we set the threshold $\tau_\rho$ to $0.85$, we connect all the vertices except between $v_2$ and $v_4$, resulting in the graph structure shown in Figure 2(b). Next, we compute the edge weights $W_{ij}$ by using the Gaussian kernel (3) with $\sigma^2 = 1$. Table I records the weights $W_{ij}$. Note that $W_{24} = 0$ because $\rho_{24} > \tau_\rho$. Figure 2(c) presents the weighted graph, where the thicker the edges, the larger the weights. A large weight means that the signals of two connected vertices are highly similar and the two corresponding vertices are strongly connected. The graph now is equivalently represented by the weighted adjacency matrix

$$\mathbf{W} = \begin{bmatrix} 0 & 0.78 & 0.60 & 0.61 \\ 0.78 & 0 & 0.52 & 0 \\ 0.60 & 0.52 & 0 & 0.98 \\ 0.61 & 0 & 0.98 & 0 \end{bmatrix}. \tag{53}$$

It follows that the degree matrix and the Laplacian are

$$\mathbf{D} = \begin{bmatrix} 1.99 & 0 & 0 & 0 \\ 0 & 1.30 & 0 & 0 \\ 0 & 0 & 2.10 & 0 \\ 0 & 0 & 0 & 1.59 \end{bmatrix} \tag{54}$$

and

$$\mathbf{L} = \begin{bmatrix} 1.99 & -0.78 & -0.60 & -0.61 \\ -0.78 & 1.30 & -0.52 & 0 \\ -0.60 & -0.52 & 2.10 & -0.98 \\ -0.61 & 0 & -0.98 & 1.59 \end{bmatrix}, \tag{55}$$

respectively.

**Graph Cut.** Recall that a cut is a set of edges whose removal will disjoin the graph. For example, with reference to Figure 3(a), the edges $e_{13}$, $e_{14}$, and $e_{23}$ assemble a cut; these edges are drawn by dashed lines. Removing these edges partitions the graph into two subgraphs, see Figure 3(b). There are many graph cuts in a given graph. Figure 4 enumerates all the graph cuts of this toy model. We next discuss which cut is the optimal one.

**Cheeger Constant.** The Cheeger constant quantifies two *strongly intraconnected* subgraphs that are *weakly interconnected* in the original graph. It is mathematically equivalent to searching for a cut with the minimum cut-to-volume ratio. We take Cut5 in Figure 4(e) as an example to illustrate how to compute its cut-to-volume ratio. Cut5 disjoins the vertex set $V$ into $V_1 = \{v_1, v_2\}$ and $V_2 = \{v_3, v_4\}$, corresponding to the indicator vector $\chi = [1, 1, 0, 0]^T$. Calculating the quantity $\chi^T \mathbf{L} \chi$ on the cut Cut5 and the volume $\chi^T \mathbf{d}$ leads to the ratio $0.53$. We could let $V_1 = \{v_3, v_4\}$ and $V_2 = \{v_1, v_2\}$ with indicator $\chi = [0, 0, 1, 1]^T$, but the volumes $\text{vol}(V_1) = \chi^T \mathbf{d} = 3.69$ and $\text{vol}(V_2) = (\mathbf{1} - \chi)^T \mathbf{d} = 3.29$ violate the definition $\text{vol}(V_1) \leq \text{vol}(V_2)$, see equation (5). Table II details the intermediate steps to evaluate the cut-to-volume ratios for all the cuts. According to the definition of the Cheeger constant, Cut5 is the optimal one because its ratio is the smallest among all. This result is not surprising when we look at the pictorial descriptions of all graph cuts in Figure 4. The thinnest (weakest) three edges $e_{13}, e_{14}, e_{23}$ form the optimal cut, while the thickest (strongest) two edges $e_{12}, e_{34}$ remain in the two disjoint, tightly linked subgraphs.

**Classifier.** The classifier is represented by the eigenbasis of the graph Laplacian. We begin with the spectral analysis of the graph Laplacian $\mathbf{L}$, whose eigenvalues and eigenvectors are given in Table III. The reader can verify that the spectral properties, (21) to (24), hold. For illustration purposes, take the full eigenbasis to represent the classifier. Later on in the next paragraph, we discuss the approximate representation. After running the optimization algorithm 2, we determine the optimal classifier to be $\hat{\mathbf{c}} = [1, 1, -1, -1]^T$ with linear combination coefficients: $a_1 = 1.69$, $a_2 = -1.04$, $a_3 = 0.28$. Substituting the components of the classifier $\mathbf{c}$ into the Heaviside function results in the optimal indicator vector $\hat{\chi} = [1, 1, 0, 0]^T$, which is equivalent to the optimal cut, namely Cut5. This shows that our optimization approach circumvents the exhaustive enumeration of all the cuts shown in Table II.

We now consider the approximate representation of the classifier. Truncate by dropping in the

classifier representation the last eigenvector $\phi^{(3)}$. The linear combination of $\phi^{(1)}$ and $\phi^{(2)}$ with coefficients $a_1 = 1.69$ and $a_2 = -1.04$ leads to the classifier $\tilde{\mathbf{c}} = [1.04, 0.92, -0.80, -1.17]^T$. Due to running a truncated representation, the classifier $\tilde{\mathbf{c}}$ no longer exactly follows the definition (25) that all its components $\tilde{c}_i$ should be either $1$ or $-1$. We force this definition by applying the Heaviside function, which yields the same optimal indicator vector $\hat{\chi} = [1, 1, 0, 0]^T$. This example demonstrates that relaxing the number of eigenvectors used for classifier representation does not necessarily sacrifice the classification performance, but does reduce the number of coefficients $a_n$ to be estimated. Thus, the computational complexity and storage demands are reduced as well.

### *B. Contrast-Enhanced Cardiac MRI*

Contrast-enhanced MRI is a useful tool to monitor the role of immune cells in many pathophysiological processes. In 1990, Weissleder *et al.* [27] introduced a new contrast agent: ultrasmall superparamagnetic iron oxide (USPIO) particles. After being administered intravenously into the circulation system, USPIO particles can be endocytosed by immune cells, so that the infiltration of the USPIO-labeled immune cells in malfunction regions of an organ will display hypointensities under $T_2^*$ weighted MRI. For example, Figure 5(a) shows the left ventricular images of two transplanted hearts with rejection, imaged on post-operation days (PODs) 4 and 6; the darker intensities in the heart reveal the presence of USPIO particles, leading to the localization of rejecting myocardial tissue.

To identify the rejecting cardiac regions, the first task is to detect the USPIO-labeled areas. There are two challenges when detecting the USPIO particles: (i) Macrophages accumulate in multiple regions without known patterns, so cardiologists must scrutinize carefully the entire image to determine dark pixels; (ii) The heart motion blurs the image, causing it to be very difficult to visually classify pixels at the boundary between dark and bright regions. Manual detection becomes labor-intensive and time-consuming, and the results are operator dependent. To reduce the expert labor work and to achieve consistent detection, we apply the spectral graph algorithm discussed in Sections II and III to classify USPIO-labeled and -unlabeled pixels.

**Experimental Setting.** In this study, the signals are image intensities. We associate each vertex $v_i$ with a $3 \times 3$ block of pixels centered at pixel $i$. Vectorization of this block of pixel intensities is treated as the feature vector $\mathbf{x}_i$ assigned to vertex $v_i$. To build the graph

representation, we adopt the Mahalanobis distance (2) to compute the vertex similarities. There are several parameters needed for running the classifier; their values are described next.

- We set $\sigma = 0.1$ when computing the edge weights in (3). This choice of $\sigma$ is suggested by Shi and Malik [28], who indicate empirically that $\sigma$ should be set at $10\%$ of the range of the image intensities. In our MRI data, the pixel intensities are in the range from $0$ to $1$.
- The parameter $\epsilon$ for the regularized Heaviside and delta functions in (45) and (46), respectively, is set to $0.1$. The smaller the parameter $\epsilon$ is, the sharper these two regularized functions are. For $\epsilon = 0.1$, the regularized functions are a good approximation to the standard ones.
- To determine the number $p$ of eigenfunctions kept to represent the classifier $\mathbf{c}$, we tested values of $p$ from $5$ to $20$. The best classification results are obtained when $p = 16$.
- To reach the minimum of the objective functional, we solve $\frac{\partial Q}{\partial \mathbf{a}} = 0$ iteratively. We terminate the iterative process when the norm of the gradient is smaller than $10^{-4}$, or when we reach $200$ iterations. This upper limit on the number of iterations led to convergence in all our experiments; in most cases, we observed convergence within the first $100$ iterations.

**Automatic Classification Results.** We apply the classifier to the images displayed in Figure 5(a). Figure 5(b) shows the detected USPIO-labeled areas denoted by red (darker pixels). The algorithm takes less than three minutes per image to localize the regional macrophage accumulation.

**Validation with Manual Classification.** To validate the results, we compare our automatic classification results with the results obtained manually by a human expert. Manual classification was carried out before running the automatic classifier. Figure 5(c) shows the manually classified USPIO-labeled regions. Our automatically detected regions show good agreement with the manual results. To appreciate better how much the classifier deviates from manual classification, define the percentage error by

$$P(\varepsilon) = \frac{|\text{(automatic USPIO-labeled area)} - \text{(manual USPIO-labeled area)}|}{\text{myocardium area}} . \qquad (56)$$

The deviation of the classifier is below $2.53\%$ average, showing a good agreement between the automatic classifier and manual classification.

**Comparisons with Other Classification Approaches.** Beyond manual classification, simple thresholding [29] is a common automatic method used for classification of USPIO-labeled regions. Figure 6(a) shows the classification results obtained by thresholding the images in

Figure 5(a). Table IV summarizes the error analysis of the thresholding classifier by using the same definition for percentage error given in (56). Although the classification results by our classifier and by thresholding shown in Figures 5(b) and 6(a), respectively, are visually difficult to distinguish, the quantitative error analysis shown in Table IV demonstrates that the thresholding method has higher error rates than the automatic classifier. Thresholding is prone to inconsistency because of the subjectivity in choosing the thresholds and because it does not account for the noise and motion blurring of the images.

We provide another comparison by contrasting our graph based classifier presented in Sections II and III with an alternative classifier, namely, the *isoperimetric partitioning* algorithm proposed by Grady and Schwartz [30]. The isoperimetric algorithm uses also a graph representation, but does not take into account the noise on the edge weights in the approach presented before. The isoperimetric algorithm attempts to minimize the objective function $\mathbf{c}^T\mathbf{L}\mathbf{c}$, where $\mathbf{c}$ is the real-valued classification function and $\mathbf{L}$ is the graph Laplacian. The minimization is equivalent to solving the linear system $\mathbf{L}\mathbf{c} = \mathbf{0}$ with a constraint that $\mathbf{c}$ is not a constant. We applied this method to the images in Figure 5(a). The classification results are shown in Figure 6(b). Comparing these results with the manual classification results in Figure 5(c), we conclude that the isoperimetric partitioning algorithm fails completely on this data set. The problems with this method are twofold. First, the objective function captures the edge cut but ignores the volume enclosed by the edge cut. This contrasts with the functional in equation (32), the Cheeger constant, that captures faithfully the goal of minimizing the cut-to-volume ratio. Second, although the desired classifier obtained by the isoperimetric partitioning is a binary function, the actual classifier it derives is a relaxed real-valued function. Our approach addresses this issue via the Heaviside function.

The final comparison is between the method presented in this chapter and the classifier derived using a *level set* approach [31], [32], which has been applied successfully to segment heart structures [9]. The level set method finds automatically contours that are the zero level of a level set function defined on the image and that are the boundaries between USPIO-labeled and -unlabeled pixels. The optimal level set is obtained to meet the following desired requirements: (i) the regions inside and outside the contours have distinct statistical models; (ii) the contours capture sharp edges; and (iii) the contours are as smooth as possible. Finally, we can classify the pixels enclosed by the optimal contours as USPIO-labeled areas. The experimental results using

the level set approach are shown in Figures 6(c) and Table IV. In the heart images, macrophages are present not only in large regions but also in small blobs with irregular shapes whose edges do not provide strong forces to attract contours. The contour evolution tends to ignore small blobs, leading to a larger misclassification rate than the graph based method presented in this chapter.

## V. Conclusion and Research Directions

Biomedical signal processing is a rapidly developing field. Biomedical data classification in particular plays an important role in biological findings and medical practice. Due to high data throughput in modern biomedical experiments, manually classifying a large volume of data is no longer feasible. It is desirable to have automatic algorithms to efficiently and effectively classify the data on behalf of domain experts. A reliable classifier avoids bias induced by human intervention and yields consistent classification results.

This chapter has discussed the usefulness of spectral graph theory to automatically classify biomedical signals. The edges of the local graph encode the statistical correlations among the data, and the entire graph presents the intrinsic global structure of the data. The Cheeger constant studied in spectral graph theory is a measure of goodness of graph partitioning. The classifier is the optimization of a functional derived from the Cheeger constant and is obtained by exploiting the graph spectrum. We detail step by step how to develop the classifier using a toy model. The application of the classifier to contrast-enhanced MRI data sets demonstrates that the graph based automatic classification agrees well with the ground truth; the evaluation shows that the spectral graph classifier outperforms other methods like the commonly used thresholding, the isoperimetric algorithm, and a level set based approach.

## Acknowledgement

The authors acknowledge Dr. Yijen L. Wu for providing the MRI data used in this chapter and Dr. Chien Ho for helpful discussions on the research. This work was supported by the National Institutes of Health under grants R01EB/AI-00318 and P41EB001977.

## References

[1] O. Majdalawieh, J. Gu, T. Bai, and G. Cheng, “Biomedical signal processing and rehabilitation engineering: A review,” in *Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, (Victoria, Canada), pp. 1004–1007, August 2003.

[2] C. Levkov, G. Mihov, R. Ivanov, I. Daskalov, I. Christov, and I. Dotsinsky, "Removal of power-line interference from the ECG: a review of the subtraction procedure," *BioMedical Engineering OnLine*, vol. 4, pp. 1–8, August 2005.

[3] N. V. Thakor and Y.-S. Zhu, "Applications of adaptive filtering to ECG analysis: Noise cancellation and arrhythmia detection," *IEEE Transactions on Biomedical Engineering*, vol. 38, pp. 785–794, August 1991.

[4] H. Gholam-Hosseini, H. Nazeran, and K. J. Reynolds, "ECG noise cancellation using digital filters," in *Proceedings of International Conference on Bioelectromagnetism*, (Melbourne, Australia), pp. 151–152, February 1998.

[5] W. Philips, "Adaptive noise removal from biomedical signals using warped polynomials," *IEEE Transactions on Biomedical Engineering*, vol. 43, pp. 480–492, May 1996.

[6] E. A. Clancy, E. L. Morin, and R. Merletti, "Sampling, noise-reduction and amplitude estimation issues in surface electromyography," *Journal of Electromyography and Kinesiology*, vol. 12, pp. 1–16, February 2002.

[7] G. Bonmassar, P. L. Purdon, I. P. Jaaskelainen, K. Chiappa, V. Solo, E. N. Brown, and J. W. Belliveau, "Motion and ballistocardiogram artifact removal for interleaved recording of EEG and EPs during MRI," *Neuroimage*, vol. 16, pp. 1127–1141, August 2002.

[8] S. Charleston and M. R. Azimi-Sadjadi, "Reduced order Kalman filtering for the enhancement of respiratory sounds," *IEEE Transactions on Biomedical Engineering*, vol. 43, pp. 421–24, April 1996.

[9] C. Pluempitiwiriyawej, J. M. F. Moura, Y. L. Wu, and C. Ho, "STACS: New active contour scheme for cardiac MR image segmentation," *IEEE Transactions on Medical Imaging*, vol. 24, pp. 593–603, May 2005.

[10] H.-H. Chang, J. M. F. Moura, Y. L. Wu, and C. Ho, "Immune cells detection of *in vivo* rejecting hearts in USPIO-enhanced magnetic resonance imaging," in *Proceedings of IEEE International Conference of Engineering in Medicine and Biology Society*, (New York, NY), pp. 1153–1156, August 2006.

[11] A.-K. Hadjantonakis and V. E. Papaioannou, "Dynamic *in vivo* imaging and cell tracking using a histone fluorescent protein fusion in mice," *BMC Biotechnology*, vol. 4, pp. 1–14, December 2004.

[12] G. J. Wiebe, R. Pershad, H. Escobar, J. W. Hawes, T. Hunter, E. Jackson-Machelski, K. L. Knudtson, M. Robertson, and T. W. Thannhauser, "DNA sequencing research group (DSRG) 2003—a general survey of core DNA sequencing facilities," *Journal of Biomolecular Techniques*, vol. 14, pp. 231–235, September 2003.

[13] D. W. Mount, *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press, 2001.

[14] J. Wang, "From DNA biosensors to gene chips," *Nucleic Acids Research*, vol. 28, pp. 3011–3016, August 2000.

[15] E. R. Dougherty, I. Shmulevich, J. Chen, and Z. J. Wang, *Genomic Signal Processing and Statistics*. Hindawi Publishing, 2005.

[16] N. Hazarika, A. C. Tsoi, and A. A. Sergejew, "Nonlinear considerations in EEG signal classification," *IEEE Transactions on Signal Processing*, vol. 45, pp. 829–836, April 1997.

[17] H.-H. Chang, J. M. F. Moura, Y. L. Wu, and C. Ho, "Automatic detection of regional heart rejection in USPIO-enhanced MRI," to appear *IEEE Transactions on Medical Imaging*.

[18] B. S. Carter, T. H. Beaty, G. D. Steinberg, B. Childs, and P. C. Walsh, "Mendelian inheritance of familial prostate cancer," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, pp. 3367–3371, April 1992.

[19] M. Huynen, B. Snel, W. Lathe, and P. Bork, "Predicting protein function by genomic context: Quantitative evaluation and qualitative inferences," *Genome Research*, vol. 10, pp. 1204–1210, August 2000.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY: John Wiley & Sons, second ed., 2001.

[21] T. Mitchell, *Machine Learning*. New York, NY: McGraw Hill, 1997.

[22] F. R. K. Chung, *Spectral Graph Theory*, vol. 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997.

[23] J. Cheeger, "A lower bound for the smallest eigenvalue of the Laplacian," in *Problems in Analysis* (R. C. Gunning, ed.), pp. 195–199, Princeton, NJ: Princeton University Press, 1970.

[24] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, pp. 1373–1396, June 2003.

[25] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 7426–7431, May 2005.

[26] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, pp. 5591–5596, May 2003.

[27] R. Weissleder, G. Elizondo, J. Wittenberg, C. A. Rabito, H. H. Bengele, and L. Josephson, "Ultrasmall superparamagnetic iron oxide: Characterization of a new class of contrast agents for MR imaging," *Radiology*, vol. 175, pp. 489–493, May 1990.

[28] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, August 2000.

[29] R. A. Trivedi, C. Mallawarachi, J.-M. U-King-Im, M. J. Graves, J. Horsley, M. J. Goddard, A. Brown, L. Wang, P. J. Kirkpatrick, J. Brown, and J. H. Gillard, "Identifying inflamed carotid plaques using in vivo USPIO-enhanced MR imaging to label plaque macrophages," *Arteriosclerosis, Thrombosis, and Vascular Biology*, vol. 26, pp. 1601–1606, July 2006.

[30] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 469–475, March 2006.

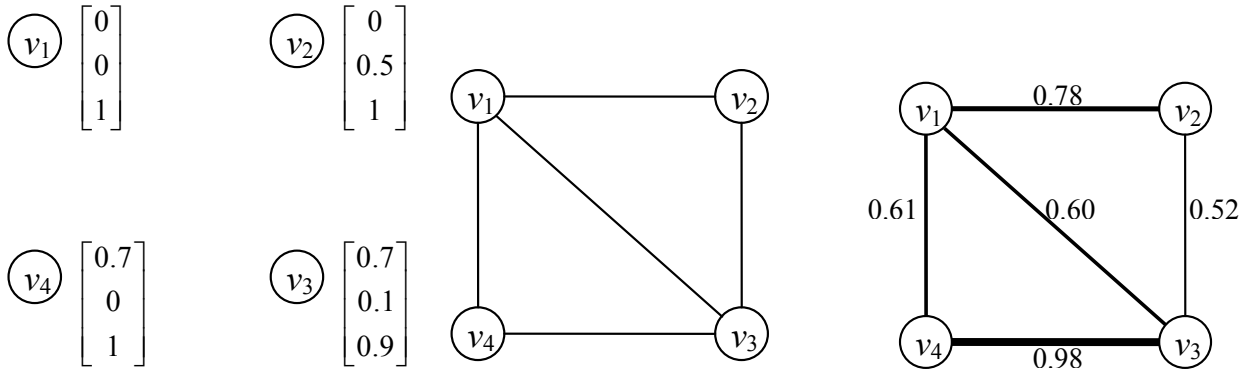
[31] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, November 1988.

[32] J. A. Sethian, *Level Set Methods and Fast Marching Methods*. New York, NY: Cambridge University Press, second ed., 1999.

(a) Graph representation.

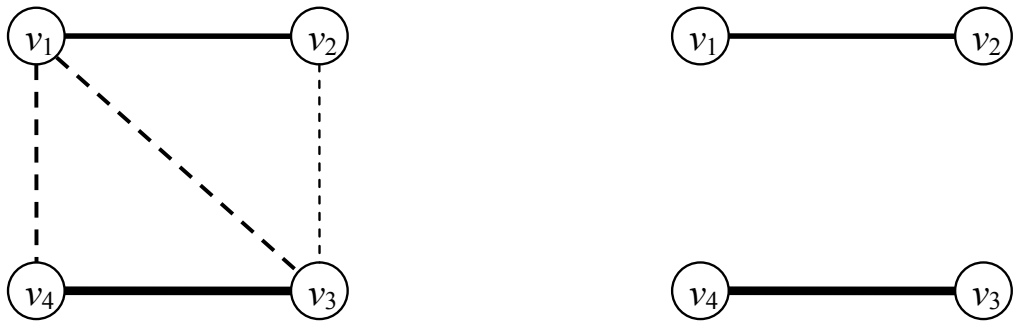
(b) Graph partitioning.

Fig. 1. Illustration of graph representation and graph partitioning.




(a) Graph nodes. (b) Graph structure. (c) Weighted graph.

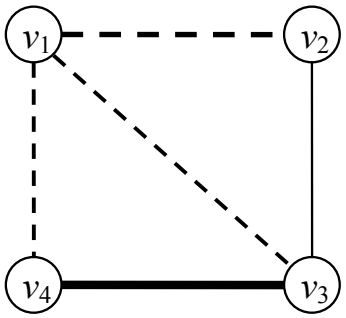
Fig. 2. Illustration of graph representation of the toy model.



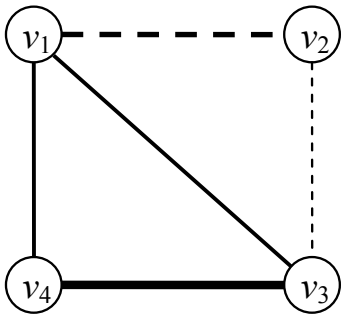

(a) The dashed edges assemble a cut.

(b) The removal of the cut partitions the graph.

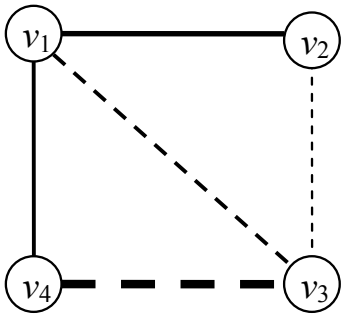
Fig. 3. Illustration of graph cut.



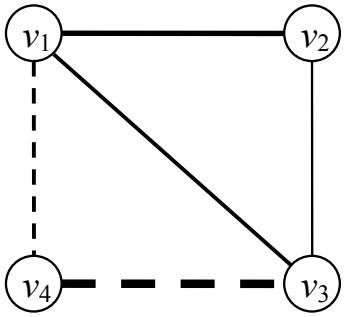

(a) Cut1.



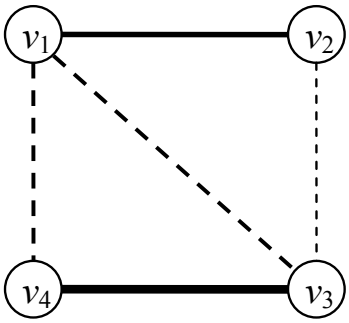

(b) Cut2.



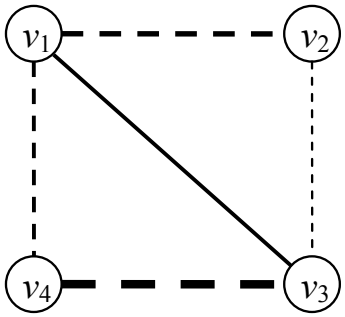

(c) Cut3.



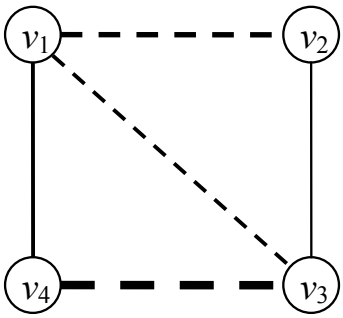

(d) Cut4.




(e) Cut5.

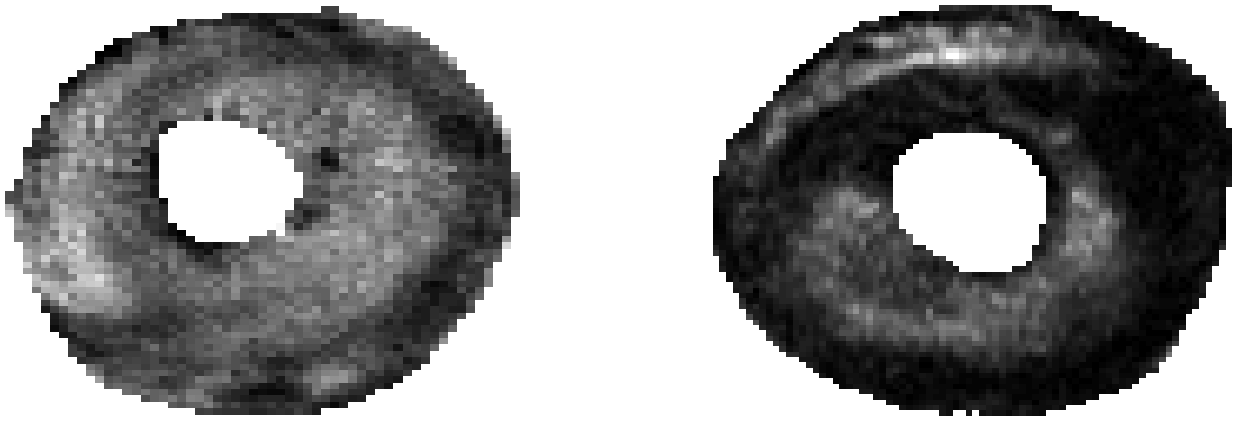



(f) Cut6.

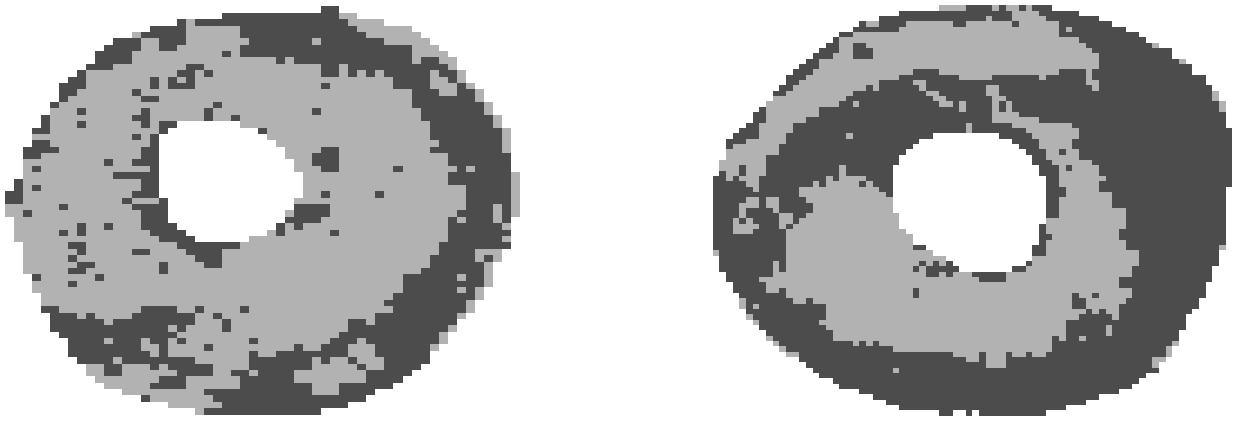



(g) Cut7.

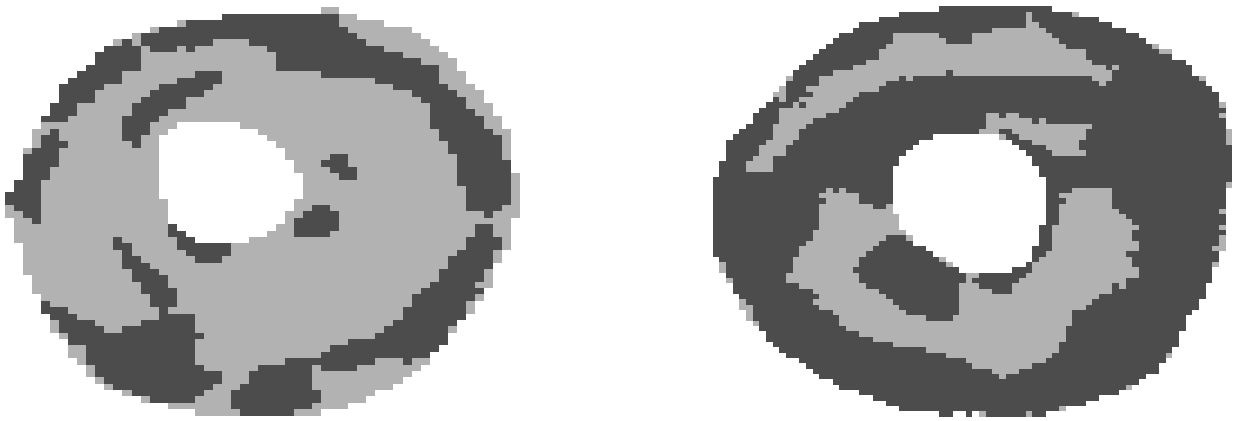
Fig. 4. Possible graph cuts of the toy model.



(a) USPIO-enhanced images.

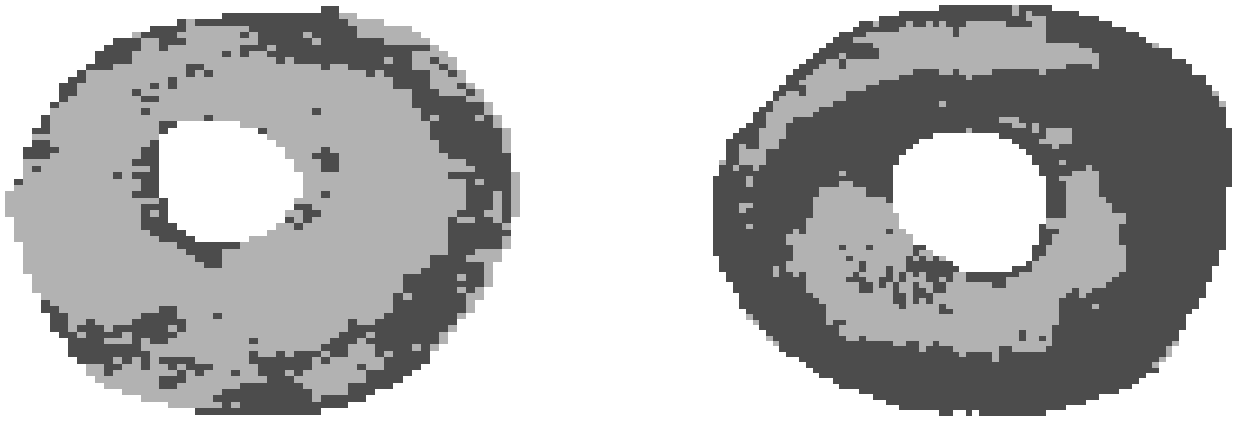


(b) Automatically classified results.

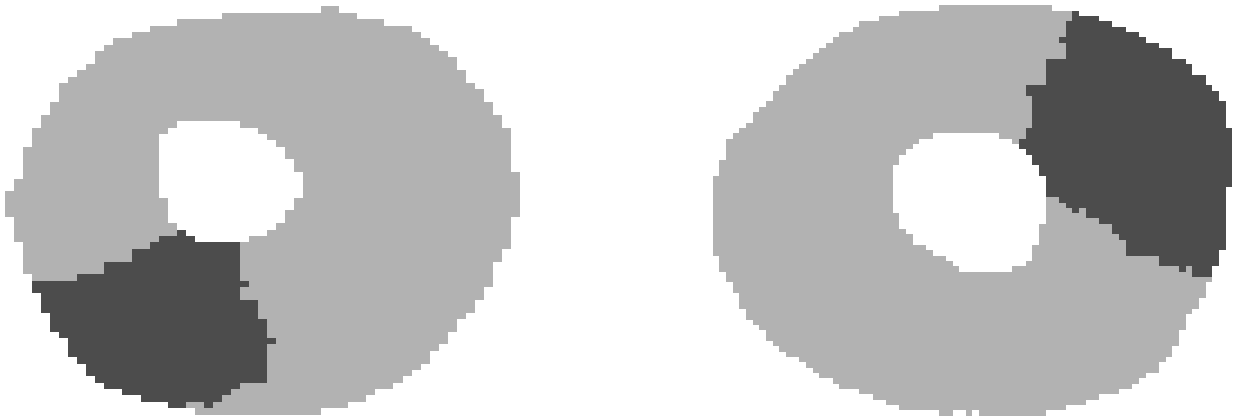


(c) Manually classified results.

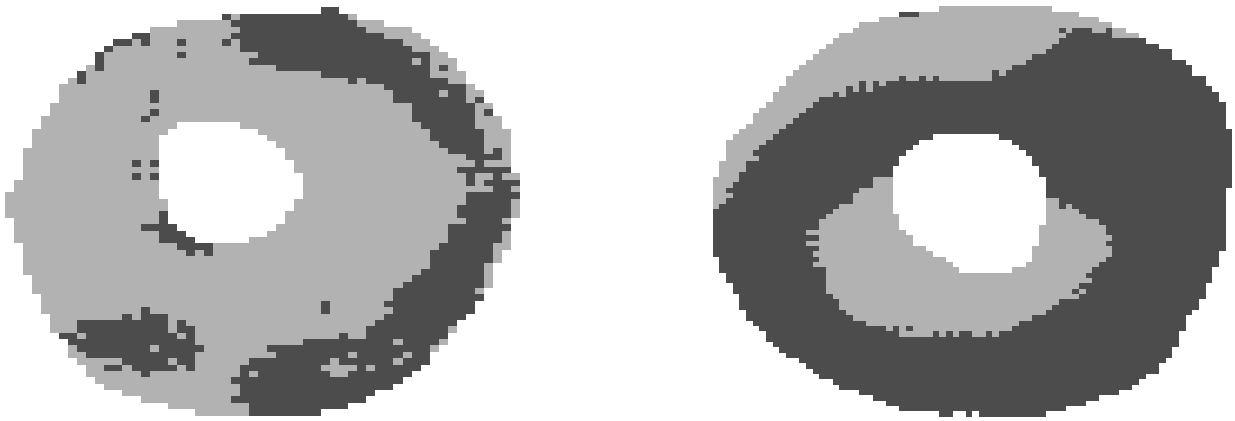
Fig. 5. Application of our algorithm to rejecting heart transplants. Darker regions denote the classified USPIO-labeled pixels. Left: POD4; right: POD6.



(a) Thresholding method.



(b) Isoperimetric algorithm.



(c) Level set approach.

Fig. 6. Application of other algorithms to rejecting heart transplants. Darker regions denote the classified USPIO-labeled pixels. Left: POD4; right: POD6.

TABLE I

SIGNAL SIMILARITIES AND EDGE WEIGHTS IN THE TOY MODEL.

| vertex pair $(v_i, v_j)$ | $(v_1, v_2)$ | $(v_1, v_3)$ | $(v_1, v_4)$ | $(v_2, v_3)$ | $(v_2, v_4)$ | $(v_3, v_4)$ |
|---|---|---|---|---|---|---|
| $\rho_{ij}$ | 0.50 | 0.71 | 0.70 | 0.81 | 0.86 | 0.14 |
| $W_{ij}$ | 0.78 | 0.60 | 0.61 | 0.52 | 0 | 0.98 |

TABLE II

INTERMEDIATE STEPS TO EVALUATE CHEEGER CONSTANTS FOR THE TOY MODEL.

| graph cut | Cut1 | Cut2 | Cut3 | Cut4 | Cut5 | Cut6 | Cut7 |
|---|---|---|---|---|---|---|---|
| $V_1$ | $\{v_1\}$ | $\{v_2\}$ | $\{v_3\}$ | $\{v_4\}$ | $\{v_1, v_2\}$ | $\{v_2, v_4\}$ | $\{v_2, v_3\}$ |
| $V_2$ | $\{v_2, v_3, v_3\}$ | $\{v_1, v_3, v_4\}$ | $\{v_1, v_2, v_4\}$ | $\{v_1, v_2, v_3\}$ | $\{v_3, v_4\}$ | $\{v_1, v_3\}$ | $\{v_1, v_4\}$ |
| $\chi$ | $[1, 0, 0, 0]^T$ | $[0, 1, 0, 0]^T$ | $[0, 0, 1, 0]^T$ | $[0, 0, 0, 1]^T$ | $[1, 1, 0, 0]^T$ | $[0, 1, 0, 1]^T$ | $[0, 1, 1, 0]^T$ |
| $\chi^T \mathbf{L} \chi$ | 1.99 | 1.30 | 2.10 | 1.59 | 1.73 | 2.89 | 2.36 |
| $\chi^T \mathbf{d}$ | 1.99 | 1.30 | 2.10 | 1.59 | 3.29 | 2.89 | 3.39 |
| $\frac{\chi^T \mathbf{L} \chi}{\chi^T \mathbf{d}}$ | 1 | 1 | 1 | 1 | 0.53 | 1 | 0.70 |

TABLE III

SPECTRUM OF THE GRAPH LAPLACIAN.

| eigenvalues | $\lambda_0 = 0$ | $\lambda_1 = 1.34$ | $\lambda_2 = 2.66$ | $\lambda_3 = 2.97$ |
|---|---|---|---|---|
| eigenvectors | $\phi^{(0)} = \begin{bmatrix} 0.50 \\ 0.50 \\ 0.50 \\ 0.50 \end{bmatrix}$ | $\phi^{(1)} = \begin{bmatrix} 0.10 \\ 0.74 \\ -0.22 \\ -0.62 \end{bmatrix}$ | $\phi^{(2)} = \begin{bmatrix} -0.85 \\ 0.33 \\ 0.41 \\ 0.11 \end{bmatrix}$ | $\phi^{(3)} = \begin{bmatrix} -0.16 \\ 0.30 \\ -0.73 \\ 0.59 \end{bmatrix}$ |

TABLE IV

PERCENTAGE DEVIATION OF VARIOUS ALGORITHMS VERSUS MANUAL CLASSIFICATION.

| method | spectral graph | thresholding | level set | isoperimetric |
|---|---|---|---|---|
| POD4 | 1.91% | 7.61% | 8.39% | fail |
| POD6 | 2.53% | 6.26% | 6.15% | fail |