Corrected Laplacians: Closer Cuts and Segmentation with Shape Priors

David Tolliver Gary L. Miller Robert T. Collins
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We optimize over the set of corrected laplacians (CL) associated with a weighted graph to improve the average case normalized cut (NCut) of a graph. Unlike edge-relaxation SDPs, optimizing over the set CL naturally exploits the matrix sparsity by operating solely on the diagonal. This structure is critical to image segmentation applications because the number of vertices is generally proportional to the number of pixels in the image. CL optimization provides a guiding principle for improving the combinatorial solution over the spectral relaxation, which is important because small improvements in the cut cost often result in significant improvements in the perceptual relevance of the segmentation. We develop an optimization procedure to accommodate prior information in the form of statistical shape models, resulting in a segmentation method that produces foreground regions which are consistent with a parameterized family of shapes. We validate our technique with ground truth on MRI medical images, providing a quantitative comparison against results produced by current spectral relaxation approaches to graph partitioning.

1 Introduction

The normalized cut (NCut), proposed by Shi and Malik [9], provides a rigorous computational foundation for image segmentation problems. A graph is constructed from the image such that the pixels constitute the vertex set, connected by weighted edges representing similarity between nearby pixels. This formulation allows the image segmentation problem to be computed via graph partitioning. The partition measure associated with the NCut seeks to simultaneously minimize the cross segment connectivity and maximize the within segment association. Yet directly computing the optimal partition under the NCut is an NP-hard combinatorial problem.

The authors of [9] develop a spectral relaxation of the normalized cut criterion to approximate their NP-hard objective function. This approximate solution, obtained from a standard eigencalculation, is optimal on the continuous definition. Relaxations of this form were further extended by Yu and Shi in [12, 13] to allow constraints to be included in the cut function. They also derive multiway cuts. Ideally, the spectral relaxation affords efficient computation of nearly optimal combinatorial solutions. However, the bound associated with this relaxation is loose. Moreover, the approximation suffers from spurious structure introduced by the constraints on the target eigenvectors (known as the Fiedler space[1]). As a result the spectral algorithm may generate poor solutions for apparently simple segmentation problems (see Figure 1).

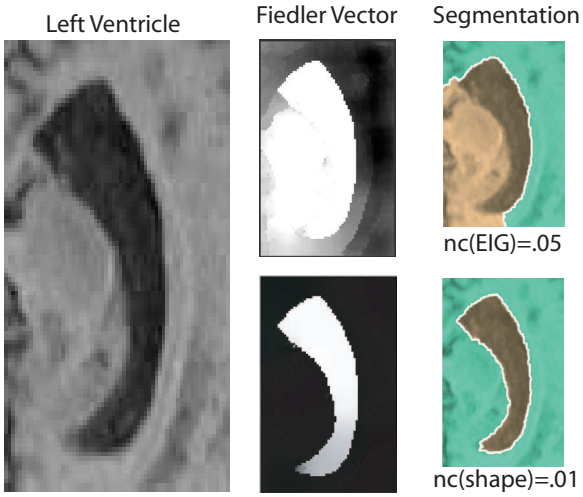


Figure 1. The eigenvector derived from the standard *spectral relaxation* to the normalized cut is shown on the top row. The eigenvector derived from the shape based *Corrected Laplacian* is shown on the bottom row. We see that the result is a superior segmentation, both visually and *w.r.t.* the cut criterion.

[1]The Fiedler space of a graph is the minimal $p-$eigenspace of a generalized graph laplacian. These vectors possess powerful properties, preserved by the NCut. For example, Fiedler vectors with minimal support partition the graph into connected subgraphs [5]. In image processing this result assures us that for most topologies, graph partitioning will result in connected components in the image plane.

We propose an additional optimization over the particular matrix encoding of the graph. This optimization is performed over a restricted class of generalized laplacians encoding the weighted graph known as corrected laplacians (CL). The result is an efficient means of improving the cut value aimed at image segmentation. By maximizing the Feidler value *(λ_2)* a tighter bound on the true optimal value of the cut is achieved, resulting in NCut segmentations that are generally better than those achieved by the standard spectral relaxation alone (Figure 2). The integration of model information allows the computationally intensive eigenvalue optimization to be exchanged for an efficient alignment procedure. More importantly, model information provides a mechanism to impose domain-specific shape constraints on the segmentation process (e.g. segmentation of ventricle-shaped regions from MRI brain images).

When optimizing shape based segmentation eigenvector methods possess desirable properties compared to flow based optimization. This is because the intermediate solutions, in the form of the eigenvectors of the corrected laplacians, contain a great deal of global information about the geometry of the cut. Accordingly, they can be used to update estimates of shape parameters. This differs from the spatially coherent clustering method proposed by Zabih [14], as the intermediate solutions in the flow optimization are not necessarily geometrically meaningful.

2. Graph Based Image Segmentation

We begin by revisiting the statement of image segmentation as a graph partitioning problem, that is, we construct a graph from the image data. Typically this is done by assigning a vertex in the graph to each pixel in the image. The connectivity of the graph is generally defined by local spatial neighborhoods in the image plane. For any two neighboring vertices an edge weight is assigned in proportion to the level of feature agreement between their pixels. Typical pairwise pixel features include color similarity, texture descriptions, and the magnitude of intervening edges along the path between two pixels. Given such a graph $G = (V, E)$ with a vertex set V and edge set E, a cut separates the graph into p disjoint subsets such that $V \supseteq \bigcup_{i=1}^{p} V_i$ and $\forall (i,j)\ V_i \cap V_j = \emptyset$.

We focus on the normalized cut [9], as it is a principled measure that partitions the graph into strongly connected subgraphs that are roughly proportionate in size while being only weakly connected to each other. The normalized cut criterion is:

$$\underset{V_1,..,V_p}{argmin} \ : \ \frac{1}{p}\sum_{i=1}^{p}\frac{|E(V_i, V \setminus V_i)|}{vol(V_i)} \qquad (1)$$

where $vol(V_i)$ is the sum of edge weights associated with the vertices in V_i, and $|E(V_i, V \setminus V_i)|$ is the sum of the edge weights connecting V_i to remainder of the graph.

The normalized cut graph partitioning criterion can be expressed as a quadratic form on the matrix representation of the graph. The combinatorial problem formulation leads to an NP-hard quadratically constrained quadratic program for the optimal p-way cut [12], expressed as:

$$nc(G) = \min_{Z} \ : \ \frac{1}{p}\, tr\,(Z^T D Z)^{-1} Z^T L Z \qquad (2)$$

$$s.t. \ : \ Z_{ij} \in \{0, 1\} \qquad (3)$$

$$: \ Z^T Z = diag(|V_1|, ..., |V_p|]) \qquad (4)$$

where $L \doteq D - W$, W is the weight matrix associated with G, D is defined as $D(i,i) = \sum_{j=1}^{n} W(i,j) = vol(v_i)$, and Z is the binary partition matrix. The constraints in Equation 3 and 4 insure that the partition matrix Z is binary and that the partitions are disjoint.

The above program can be approximated in many ways. In the vision community, the most common approximation is the eigenvector relaxation *(or spectral relaxation)*. This entails relaxing the constraints on Z to a $D-$orthonormality constraint on the generalized eigenvectors $Y \mid Y^T D Y = I$. Under this relaxation Equation 2 is a Generalized Eigenvalue Problem *(GEP)*. For computational stability, the symmetrized form *(SEP)* is used, where the minimization occurs over X in $min \ : \ tr\ (X^T X)^{-1} X^T \mathcal{L}(W) X$ and $\mathcal{L}(W) \doteq D^{-1/2}(D - W)D^{-1/2}$. The solutions to *SEP* can be mapped to solutions for *GEP* as $Y = D^{-1/2}X$.

In the multi-cut spectral algorithms proposed by [8, 12] the $p-$dimensional eigenspace minimizing Equation 2 associates a coordinate with each vertex in the graph. These $p-$dimensional coordinates are then projected on the sphere $\mathcal{S}^p$. Heuristic geometric clustering on the sphere is then applied to partition the graph. We will use $\mathcal{C}_Z(Y)$ to denote the final result of the clustering step in which Y is projected onto a feasible solution Z. Unless stated otherwise, we employ Yu's method from [12] for $\mathcal{C}_Z$.

2.1 Spectral Approximations for Partitioning

Applications of spectral relaxations for graph partitioning were first proposed in the field of domain decomposition *(surveyed in [10])*. In domain decomposition the goal is the construction of a divide-and-conquer algorithm that partitions a large problem into several proportionate small problems with minimal communication. These techniques have been applied to derive parallel algorithms for solving large sparse linear systems, VLSI layout, and discrete PDEs.

The quality of the spectral relaxation has been analyzed for the eqi-partitioning problem by Gauttery and Miller [6], and Spielman and Teng [10]. The former present a class of graphs, with analysis, that result in arbitrarily poor partitions under the spectral relaxation. These pathologies hold

for the spectral algorithms used to optimize the normalized cut. The authors of [10] give a class of meshes that are well partitioned under the spectral relaxation. In terms of computer vision problems, it is unclear when the spectral relaxation is effective or if near pathological cases exist in common visual phenomena. Figure 2 demonstrates that the straight eigenrelaxation is problematic even for simple segmentation problems.

The common spectral algorithms for the NCut are in fact equivalent to spectral algorithms for partitioning a linearly normalized graph. Accordingly, it is not surprising that the solutions reflect the oscillatory structure vital to mesh partitioning, yet damaging to image processing applications. We state the following simple comment on the representations derived from the spectral algorithms.

Proposition 1 *The geometric representation of the graph derived from the generalized eigenproblem (GEP) relaxation to the NCut used in [8, 12], is identical to that derived from the normalized laplacian (SEP).*

Proof: This reduces to showing that the GEP and the SEP yield the same coordinates for each vertex in the graph. First, the eigenvectors minimizing the GEP and the SEP are related as follows: the solution Y to $(D-W)Y = DY\Lambda_p$ is related to X in $\mathcal{L}(W)X = X\Lambda_p$ as $Y = D^{-1/2}X$. Second, the embedding procedure employed in [8, 12] normalizes the rows of Y, projecting the points onto the unit sphere. This removes the effect of the normalizer $D^{-1/2}$ since D is positive and diagonal it only scales the rows of Y along a ray. Therefore the multiclass spectral approximation algorithms in [8, 12] for the NCut are equivalent to an approximation for the minimum multiway cut of $D^{-1/2}WD^{-1/2}$. And so, we have an approximation upon an approximation. The corrected laplacians aid here, as we further optimize over $D-$orthonormalized eigenspaces of nearby graph laplacians.

3 Method

Geometric optimization approaches to graph partitioning optimize over coordinate maps for the vertices of the graph. These geometric representations construct a point-set embedding of the graph such that the target cut metric may be optimized by geometric clustering. Spectral algorithms fall into this family, since the end result of the eigenspace computation is a $p-$dimensional coordinate for each vertex in the graph. By searching over the set of corrected laplacians *(CL)* we maintain this feature; as the optimization is over the eigenspaces of a family of positive semidefinite (PSD) matrices encoding the graph.

Good geometric representations allow quality cuts to be computed with high probability. For example vector programs approximating the maximum cut seek a geometry such that strongly connected vertices possess coordinates as near to antipodal as possible. Thus a random hyperplane will, with high probability, yield a large cut. For the normalized cut, the tightly connected vertices should be close under the geometry, with roughly orthogonal mean coordinate vectors between clusters.

3.1 Bound Optimization

We begin by recalling that the conductance of a graph can be related to the normalized cut (NCut) of the graph by a constant factor (as noted Shi and Malik in [9]). Given this we can use the eigenvalue bound associated on conductance, $\Phi(G)$, of the graph G : $\Phi(G) \geq c\,\lambda_2(G)$ *(see [5] for further details on the bound and conductance)*. This suggests a simple bound optimization for improving the quality of the NCut approximation. We do this by maximizing $\lambda_2(\mathcal{L} + C)$ as a function of trace zero diagonal matrices C, subject to a semi-definite constraint on the sum, along the lines of Bopanna's MAX-CUT bound in [1].

We illustrate the effect of the bound optimization on the image of an ellipse corrupted with structured noise shown in Figure 2. This image demonstrates how spectral solutions contain ancillary structure due to the graph topology. The result obtained from the standard spectral algorithm is shown on the top row. This segmentation is often explained by the large-partition bias in the normalized cut. However, the segmentation obtained from the corrected laplacian extracts the ellipse and has an improved cut value.

Upon detailed examination, we see that the Fiedler vector, used in the standard spectral algorithm, contains a planar sinusoidal structure. This structure leads to an ambiguous embedding, as shown as the blue points in Figure 3. The embedding obtained from the corrected laplacian, $\mathcal{L}+C^*$, is shown in green and suggests a less ambiguous partitioning of the vertices.

In essence the bound optimization minimizes the effect of the eigenmodes of the graph topology. That is, the eigenvectors of the image even without the ellipse are highly structured, due to the topology of the graph. This structure appears visually as sinusoidal functions in the image plane, evocative of the physical modes of a square mesh. We see that a correction vector, applied to the diagonal of $\mathcal{L}$, flattens this oscillation out of the Fiedler vector, bottom row of Figure 2.

This modification has a simple physical intuition. Imagine the graph as a spring and mass system. The mass associated with each vertex has been normalized to 1 by $\mathcal{L}(W)$, and is close to the total weighted degree of the vertex. The effect of $\vec{u}$ can be interpreted as either increasing or decreasing the relative mass of vertex v_i. If the mass is decreased at v_i, it is proportionally more strongly coupled to its neighbors. If the mass is increased, v_i can oscillate more freely

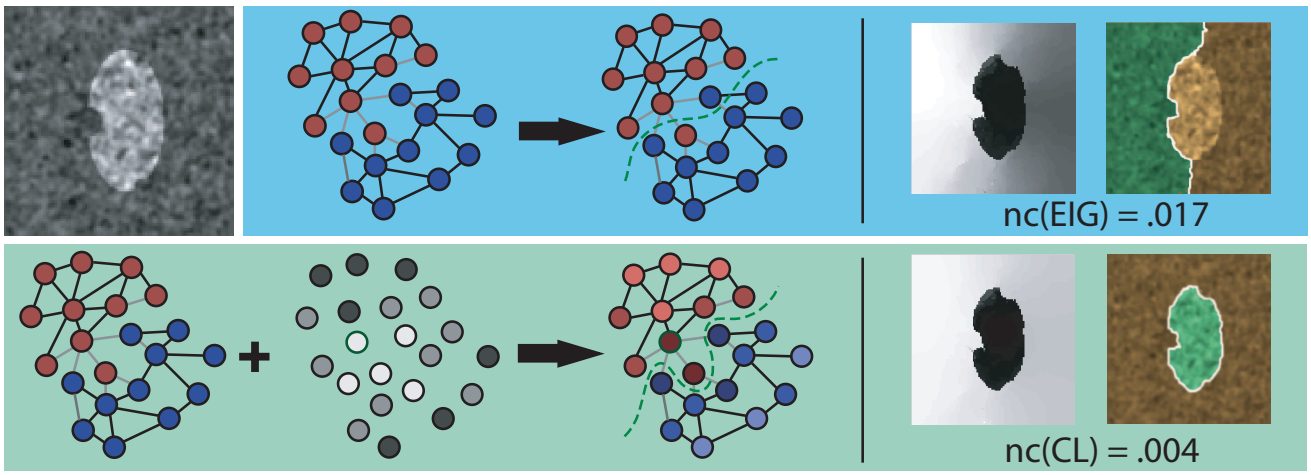


Figure 2. The image of a corrupted ellipse motivates the use of an corrected laplacians for NCut based image segmentation. The top row shows the solution obtained from the spectral algorithm for the NCut. Note that the Fiedler vector contains the ellipse, and a low frequency sinusoid. The values of the pixels at the far right are close to those taken by the ellipse. The second row shows the Fiedler vector obtained after adding the correction vector to the diagonal of the normalized laplacian. The discrete solutions, and NCut values are shown in the right column. We see, the corrected solution is not only more aesthetically pleasing, but also improves the NCut value.

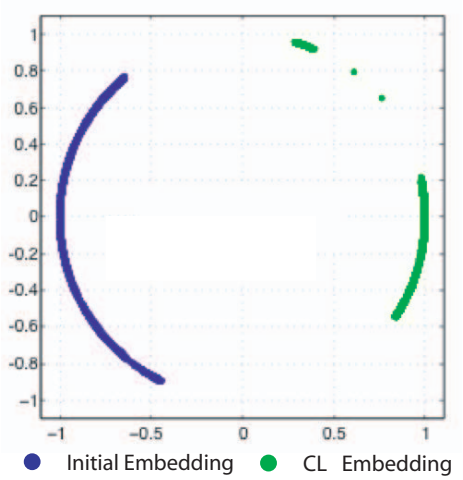


Figure 3. The geometric representation derived for the noisy ellipse example in Figure 2. The initial embedding derived from NCut eigenrelaxation is shown on the left. The Fiedler space derived from the corrected laplacian is shown on the right. *Note: the effect of the graph topology has been suppressed, resulting in the crisp embedding (green).*

from its neighborhood. In terms of the cut, increases in mass result in discounted cuts involving v_i.

The SDP paradigm was applied to the normalized cut by Xing and Jordan [11], however they worked with an *edge*-relaxation. Edge-relaxations involve $O(n^2)$ new parameters, whereas the vertex-relaxation only requires n new parameters. The application of image segmentation seems to benefit from SDPs in a way that generic clustering may not. This difference is explained as the graph linkage, in image processing applications, is defined by both local spatial neighborhoods and feature agreement. These spatial neighborhoods introduce sinusoidal structure in the eigenspace. Whereas in clustering problems, the graph linkage is determined entirely by the geometry of the feature space.

The proposed *vertex*-relaxation is advantageous as it directly exploits the sparsity of the problem, and provides a simple physical intuition to motivate the introduction of prior information into the optimization. To this end we define a guided optimization using side information. The source of this information can be user input, a parametric shape model, or any other oracle capable of providing semi-reliable pixel level information regarding the image.

3.2 Alignment as Guided Optimization

In many domains, such as medical image processing, we know the type of structure we wish to segment in the image or volume. Accordingly, a mechanism for guiding the optimization of the graph cut with shape estimates improves the accuracy of the segmentation. To incorporate model data into the segmentation, we align the Fiedler space with a vector (or subspace) encoding the current estimate of the shape. The shape estimate is then updated by fitting regions in the Fiedler vector that are likely to contain the cut (see §3.1.1.) This is done by iteratively minimizing the following simple objective function:

$$\mathcal{O}(Y_{\mathcal{F}}, Y_S) \doteq 1 - (Y_{\mathcal{F}}^T Y_S)^2 \qquad (5)$$

where $Y_{\mathcal{F}}$ is the normalized Fielder vector from §2, and Y_S is a vector representation of the current shape model estimate. Clearly $\mathcal{O}$ is minimized when $Y_{\mathcal{F}}$ and Y_S are aligned,

up to a sign. As $Y_{\mathcal{F}}$ is a continuous function of the entries of the weight matrix, we determine values in $\vec{u}$ with a gradient update aligning $Y_{\mathcal{F}}$ to Y_S.

The following iterative scheme is used for simultaneous shape and normalized cut segmentation.

Algorithm 1: Minimize $\mathcal{O}(Y_{\mathcal{F}}, Y_S)$

1: *Fix $Y_{\mathcal{S}}$, and minimize $\mathcal{O}$ on $Y_{\mathcal{F}} \doteq D^{-1/2}X_{\mathcal{F}}$ with $min:\ tr\ X_{\mathcal{F}}^T(\mathcal{L}(W) + diag(t\,\vec{u}_n))X_{\mathcal{F}}$. Resulting in a 1-parameter search on the norm of the correction $\vec{u}$.*

2: *Update the shape model by fitting to high magnitude contours in the burn map, derived from $Y_{\mathcal{F}}$, and create a shape vector Y_S from the current shape estimate.*

3: *Compute the update vector $\vec{u}_n$ that aligns $Y_{\mathcal{F}}$ to $Y_{\mathcal{S}}$, and iterate until convergence.*

The method is initialized with $\vec{u} = 0$, *(i.e. the standard spectral relaxation)*. Finally we note that an estimate on the norm t can be determined from matrix perturbation relating $||Y_{\mathcal{F}} - Y_{\mathcal{S}}||_2$ to $|t|$.

3.2.1 The Differentials of a Generalized Eigenproblem

To update the eigenspace of a matrix as a function of its entries, we state the differentials of Y and λ, dY and $d\lambda$, with respect to the entries of L. Our derivation closely follows that of Magnus and Neudecker [7]. They employ the implicit function theorem (*IFT*) to show the existence of the differentials $dY,\ d\lambda$, and require that λ_0 is simple. Further we note that the authors of [3] previously used the same tools to derive a gradient rule learning the graph structure from training data in a NCut paradigm.

For the generalized eigenproblem we form the function f_{GEP}, such that $f_{GEP}(y_0, \lambda_0;\ L) = 0$ for the eigenpair (y_0, λ_0) as follows:

$$f_{GEP}(y_0, \lambda_0;\ L) \doteq \begin{pmatrix} A \\ b \end{pmatrix} \doteq \begin{pmatrix} (L - \lambda_0 D)y_0 \\ y_0^T D y_0 - 1 \end{pmatrix} \quad (6)$$

as above the eigenpair y_0, λ_0 is a zero of the function f_{GEP}. To validate f_{GEP} and insure that the smooth open ball $\mathcal{N}(y_0, \lambda_0)$ exists we must meet the requirements of the *IFT*. The first requirement that $f_{GEP}(y_0, \lambda_0;\ L) = 0$ follows from the definition; that a function $g_{GEP} = \dot{f}_{GEP}$ exists, and that $g_{GEP}(0) = 0$ is all that remains[2]. Given that the conditions of the *IFT* are met by f_{GEP} we arrive at explicit expressions for $d\lambda$ and dY. The explicit forms of the differentials of the implicit functions $\lambda(L)$ and $Y(L)$ are obtained by differentiating $Ly = \lambda Dy$ as follows:

$$(dL)y + L(dy) = (d\lambda)Dy + \lambda(dD)y + \lambda D(dy). \quad (7)$$

Solving for $d\lambda$ in 7 we obtain:

$$d\lambda = y_0^T(dL - \lambda_0(dD))y_0 \quad (8)$$

by premultiplying with y_0^T; noting that $y_0^T D_0 y_0 = 1$, and grouping the remaining terms $(dx)^T(L - \lambda_0 I)y_0 = 0$ as $(L - \lambda I)y_0 = 0$. We now derive the explicit form of dY:

$$(L_0 - \lambda D_0)dY = ((d\lambda)D_0 + \lambda_0(dD) - dL)Y_0 \quad (9)$$

where dY is clearly the solution a constrained linear system, subject to $dY \perp y$, $dD = I$ and $dL = Adjacency(G)$.

3.2.2 The Gradient Update: *a vertex formulation*

The alignment of the eigenspace to the shape estimate vector can be accomplished through the additive correction $\vec{u}$ in Algorithm 1, step 1. The update $\vec{u}_i$ for each vertex $v_i \in V$ is computed as the expected change in weighted degree.

$$\begin{aligned} \partial u_i &= \mathbb{E}_i\left\{\frac{\partial \mathcal{O}}{\partial X_i}\frac{\partial X_i}{\partial L_{ij}}\right\} \quad (10) \\ &= -2\left(A_{ZY} + A_{ZY}^T + \lambda I_{ZY} + A_{YY} - 2\lambda I_{YY}\right)_i \end{aligned}$$

where $Z \doteq (dY)$, $A_{ab} \doteq \mathcal{A}(G) \circ (ab^T)$, $I_{ab} = I \circ ab^T$, and $\circ$ denotes the point-wise product.

3.3 Shape Subspace Model

We construct a statistical family of foreground shapes using the point distribution model approach of Cootes *et.al.* [2]. First, a training set of N hand-labeled shape contours consisting of n corresponding 2D landmark points $\{(x_i, y_i)|i = 1, \ldots, n\}$ is aligned by scaling, rotating and translating each shape to minimize the sum of squared distances between corresponding points. Note that this is equivalent to performing a *Procrustes* shape fitting procedure to align the contours up to a similarity transformation [4]. Forming a $2n$x1 vector $\boldsymbol{x}_j$ containing the aligned points for the jth training example, the mean shape is computed as $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_1^N \boldsymbol{x}_j$, and the resulting "shape" model of the set of contours is defined by the residual variation of the $\boldsymbol{x}_j$ about the mean shape, captured by the $2n$x$2n$ covariance matrix $\boldsymbol{S} = \frac{1}{N}\sum_1^N (\boldsymbol{x}_j - \bar{\boldsymbol{x}})(\boldsymbol{x}_j - \bar{\boldsymbol{x}})^T$. The principle directions of shape variation are defined by the eigenvectors $\boldsymbol{p}_i$ of $\boldsymbol{S}$, with amount of variance along each direction measured by the corresponding eigenvalue λ_i. As is common in PCA-type models, we select an m-dimensional subspace composed of the eigenvectors corresponding to the m

[2] To show that g_{GEP} exits, the Jacobian on (y, λ) of f_{GEP} must be non-singular. This fact is verified from the formula for the determinant of a bordered matrix as follows: $det\ J_{GEP} = -y^T D^{1/2}(L - \lambda D)^\sharp D^{1/2} y$, where $B^\sharp$ denotes the transpose of cofactor matrix of B. Recall that for a non-singular matrix B, $B^{-1} = \frac{1}{detB}B^\sharp$. As the $(L - \lambda D)$ is $rank(n-1)$, the cofactor matrix consists of scaled versions of its nullspace space, therefore $det\ J_{GEP} > 0$. This follows because $(D^{1/2}y)^T y > 0$, for a positive diagonal operator D, and as y is in the nullspace of $(L - \lambda D)$. As $g = J^{-1}f$, $g_{GEP}(0) = 0$ follows from the formula for the determinant above. See Magnus and Neudecker [7] for details on the determinants of bordered matrices.

largest eigenvalues, with m chosen such that the resulting subspace accounts for 95% of the total shape variance. The mean shape and the principal axes of shape variation allow a shape $\boldsymbol{x}$ to be described by m shape parameters b_i, as

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \sum_{i=1}^{m} b_i \boldsymbol{p}_i.$$

Given parameters b_i of a proposed foreground shape $\hat{\boldsymbol{x}}$, we can compute the likelihood of that shape with respect to our training distribution by assuming a Gaussian process defined by the mean shape and variances along the principal axes

$$P(\hat{\boldsymbol{x}}) \propto \exp\{-\sum_{i=1}^{m} \frac{b_i^2}{\lambda_i^2}\} \quad . \tag{11}$$

The key, therefore, is to find an efficient mechanism for estimating the shape parameters b_i for the foreground region implied by a current segmentation, and to adjust the current segmentation in a way that favors shapes with higher likelihood.

3.3.1 Energy Term for the Model Optimization

The shape parameters are estimated by explaining vertices of the Fiedler vector that are likely to be adjacent to the cut. Drawing on the analogy of the graph to an electrical circuit, we can think of such vertices as those with high voltage burn. These *burn maps* are obtained by applying the laplacian of the prior graph topology to the vector. This results in a new vector that measures the level of discord in values taken in the neighborhood of each vertex. Accordingly, vertices that are likely adjacent to the cut will possess high value in the *burn map*. In this way, the Fiedler space of the graph laplacian can be interpreted as a collection of edge preserving smoothers of the image. We include in this space all minimizing vectors with minimal support [5]. The shape parameters are estimated by fitting the areas of high *burn* in the Fiedler space. We assign a discrete monotonic probability function such that $P(data|\hat{\boldsymbol{x}}) = \frac{1}{Z}1$ if all the energy in the *burn map* is explained by the contour shape $\hat{\boldsymbol{x}}$. The contour shape score is calculated as $P(data|\hat{\boldsymbol{x}})P(\hat{\boldsymbol{x}})$ where $P(\hat{\boldsymbol{x}})$ was defined in Equation 11. Multiple samples are fit by perturbed restarts under the transform and parameter distributions followed by gradient ascent maximization on the shape score.

3.3.2 Deriving Guiding Vectors from Shape Models

The sum of the scores, $P(data|\hat{\boldsymbol{x}}_k)P(\hat{\boldsymbol{x}}_k)$, of the sample fits of the model is normalized to 1. The interior of the shape region is set to 1, the exterior 0, for each sample fit. A combined estimate is then computed as the weighted average of the shape sample regions. Uniform negative values are added to the zero entries to create a vector $c|c^T D\mathbf{1} = 0$. The shape alignment target vector is then $Y_S = c/(c^T c)^{1/2}$. This vector is then used in step 3 of Algorithm 1 to calculate the update to the Fiedler space.

4 Empirical Evaluation

The parameters for weight matrix construction were fixed for all the presented segmentation results. The neighborhood radius was set to 10 pixels with 20% of the pixels being sampled in a regular radial pattern. The intervening contour, described in [9], was used as the pixel affinity cue. This cue decouples vertices that are on opposite sides of a strong edge in the image plane.

4.1 *Model Integration:* Ventrical Segmentation

We evaluate our method on a collection of 200 MRI images containing the left ventricle. The collection was built by taking 5 images from scans of 40 individuals drawn from two populations : $\{normal, Alzheimers\}$. The images were selected randomly from slices that contain the left ventricle. As shown in Figure 4 these images contain substantial variation in shape and size. The shape model used to guide the optimization was trained on 40 images taken from 2 subjects whose scan did not appear in the evaluation set.

The results of the comparison are shown in Table 1. The proposed method is compared to the NCut [9], and NCut with bias [13]. For the NCut with bias, a background constraint mask of 3 pixels was set along the image border. The top two rows contain the frequency of foreground pixel differences between the algorithm and ground truth with standard deviations. For the CL-Shape results, error generally occurs along the border of the foreground region. For the $nc(BIAS)$ results, the errors are generally small foreground regions, while the variance is due to images in which mask pixels were tightly coupled to the foreground pixels. The bottom two rows contain the average compute time for the solutions. The feasible space projection $Z = C_S(Y)$ finds the best total correlation foreground partition with the model estimate as a 1-parameter search along the value of the Fielder vector. This is presented for the bias and standard NCut to illustrate the improvement due to joint estimation of shape and segmentation.

5 Summary and Conclusions

We have shown that *corrected laplacians* improve the average case normalized cut, and can be used to encode shape information into the eigenvectors of the graph laplacian. Unlike edge-relaxation SDPs, the formulation directly exploits the sparse structure of graphs derived from image data. This provides an avenue for efficient techniques

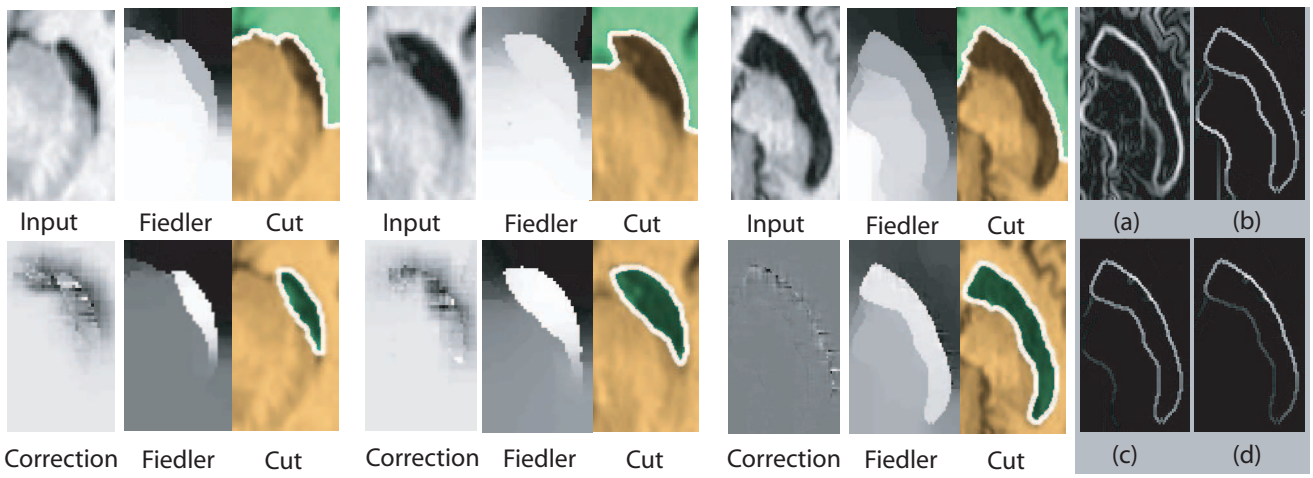


Figure 4. Examples of ventrical shape and size variation in the MRI image population. The upper row contains three columns of: input image, spectral relaxation vector, and partition. The bottom row contains: the correction vector for $\mathcal{L}$, the corrected Fiedler vector Y, and the resulting partition. The fourth column contains the original edge map (a), and *burn maps* at the first (b), third (c), and the fifth and final iteration (d) of **Algorithm 1**

	NCut(EIG)	NCut(BIAS)	NCut(CL-Shape)
$\mathcal{C}_Z$	$.72 \pm .10$	$.36 \pm .17$	$.10 \pm .04$
$\mathcal{C}_S$	$.61 \pm .15$	$.28 \pm .11$	$.07 \pm .01$
$\mathcal{C}_Z$	$1.0s \pm .01s$	$1.1s \pm .01s$	$8.1s \pm .11s$
$\mathcal{C}_S$	$1.1s \pm .01s$	$1.2s \pm .01s$	$8.2s \pm .13s$

Table 1. A comparison of segmentations derived from the spectral relaxation of [12], the biased NCut [13] and the shape corrected laplacians (CL-Shape).

for optimizing the bound and the cut. For segmentation with prior models corrected laplacians provide a scaffolding upon which coupled shape and data segmentation may be optimized.

The proposed technique was validated experimentally on real imagery using a collection of MRI images of the left ventricle in 40 human subjects. This concrete segmentation task afforded a quantitative comparison with current techniques.

In future work, we intend to analyze the computational properties of the correction vector approach. For example, conditions in which the optimal normalized cut value disagrees with our desired segmentation are bound to occur. In such cases, it would be useful to provide a quality normalized cut and yet bias more toward model consistency. At this time we have not yet analyzed the degree to which the model fidelity versus NCut partition score trade-off can be controlled. Finally, we are evaluating the vertex-SDP method on problems requiring robust clustering such as optical flow segmentation.

References

[1] R. Bopanna. Eigenvalues and graph bisection: an average-case analysis. *FOCS*, pages 280–285, 1987.

[2] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models – their training and application. *CVIU*, 61(1):38–59, January 1995.

[3] T. Cour, N. Gogin, and J. Shi. Learning spectral graph segmentation. *AISTATS*, 2005.

[4] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, New York, 1998.

[5] C. Godsil and R. Gordon. *Algebriac Graph Theory*. Springer, 2001.

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

[7] J. R. Magnus and H. Neudecker. Matrix differential calculus: with applications in statistics and econometrics. 1999.

[8] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[9] J. Shi and J. Malik. Normalized cuts and image segmentation. In *PAMI*, 2000.

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

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

[12] S. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, October 2003.

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

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