Numerical Aspects of Spectral Segmentation on Polygonal Grids

Anna Matsekh, Alexei Skurikhin, Lakshman Prasad, and Edward Rosten

Space and Remote Sensing Sciences Group, Los Alamos National Laboratory
Engineering Department, University of Cambridge
{matsekh,alexei,prasad}@lanl.gov,
er258@cam.ac.uk

Abstract. We present an implementation of the Normalized Cuts method for the solution of the image segmentation problem on polygonal grids. We show that in the presence of rounding errors the eigenvector corresponding to the k-th smallest eigenvalue of the generalized graph Laplacian is likely to contain more than k nodal domains. It follows that the Fiedler vector alone is not always suitable for graph partitioning, while the eigenvector subspace, corresponding to just a few of the lowest eigenvalues, contains sufficient information needed for obtaining meaningful segmentation. At the same time, the eigenvector corresponding to the trivial solution often carries nontrivial information about the nodal domains in the image and can be used as an initial guess for the Krylov subspace eigensolver. We show that proposed algorithm performs favorably when compared to the Multiscale Normalized Cuts and Segmentation by Weighted Aggregation.

Keywords: image segmentation, spectral graph partitioning, symmetric eigenvalue problem, generalized graph Laplacian.

1 Introduction

Image segmentation methods often rely on the use of the spectral graph partitioning techniques [1,2], when an image is represented by a simple weighted undirected graph. In this setting, graph vertexes represent image primitives, such as pixels, while its edges describe relationships between the neighboring image primitives. On the subsequent steps, graph clustering objective, seeking to partition image graph into a set of disjoint coherent segments, is set up.

Spectral graph partitioning presents a number of numerical challenges. The most notable problem is the complexity reduction of the underlying eigenvector computations – the bottle neck of all spectral segmentation methods [3]. This problem is typically addressed either within an algebraic multigrid or multilevel graph partitioning framework. This approach has been successfully adapted for the solution of image segmentation problems in multiscale [4,5,6] and algebraic multigrid-like frameworks [7]. In the current study, we replace an image with its

K. Jónasson (Ed.): PARA 2010, Part I, LNCS 7133, pp. 193–203, 2012.

compressed version on an unstructured polygonal grid, assuming that compression preserved main properties of the eigenvalue spectrum of the graph Laplacian.

Sometimes, the authors construct spectral segmentation in multidimensional spaces spanned by a few eigenvectors, corresponding to the smallest eigenvalues of the graph Laplacians [8], rather than relying on recursive bisection of the Fiedler vector alone – the theoretical optimal solution to the graph partitioning problems [1,2]. However, as far as we can tell, there has never been given a clear explanation why recursive bisection of the Fiedler vector alone may not always produce satisfactory segmentation. We try to give this explanation by analyzing the effects of the rounding errors on the finite precision solution to the problem in the Krylov subspace generated by a variant of the Cullum-Willoughby-Lanczos method [9,10].

2 Problem Setup

Let J be a pixel image and let P be its approximation on a polygonal grid obtained using VISTA software [11]. Polygonized image P consists of n primitives p_i obtained by the application of the Canny edge detector on the image J, followed by the application of an algorithm that superimposes Constrained Delaunay Triangulation (CDT) over the detected edges, assigning average colors to the triangles by Monte Carlo sampling of the pixels in each triangle [11]. The triangulated image is then further coarsened by combining neighboring triangles representing coherent patches, into polygons $p_i,\ i = 0, 1, \ldots, n-1$ [11]. Each polygon p_i is attributed with color u_i which is an aggregate of the colors of the triangles polygon p_i is comprised of. Typically, the number of polygons will be significantly smaller than the number of the pixels (*e.g.* see Figure 1). We will treat approximation P as a compressed image J, replacing J with P on the consecutive steps.

Our goal is to segment the polygonized image P into coherent visual scenes using the Normalized Cuts graph partitioning objective function [13] that seeks to minimizes graph cuts relative to the weighted degrees of its clusters. This problem formulation provides valuable information about the weighted degrees of graph nodes that we exploit when solving the underlying sparse symmetric eigenvalue problem. We represent P as a simple weighted undirected graph $G(V, E)$ with vertex set $V = (v_0, v_1, ..., v_{n-1})$ and edge set $E = \{e_{ij} \mid e_{ij} = v_i\, v_j\}$. Graph vertexes v_i describe polygons p_i, while its edges e_{ij} connect vertexes that represent neighboring polygons p_i and p_j. Graph G is then described by the weighted adjacency, or affinity, matrix $W \in \mathbb{R}^{n\times n},\ W = W^T$ with elements w_{ij} representing weights associated with the edges e_{ij}.

Let $u_i = (u_i^0, u_i^1, u_i^2)$ be the aggregate color assigned to the polygon p_i in the CIELAB color space. Let (x_i^k, y_i^k), $k = 0, 1, \ldots, n_{i-1}$ be the coordinates of its n_i vertexes. We call two polygons p_i and p_j neighbors if they share at least one

(a) original image (b) polygonized image

Fig. 1. Geometric image coarsening with VISTA: (a) original 321 × 481 pixel image from the Berkeley Segmentation Data Set [12]; (b) polygonized image consisting of 4521 primitives

Canny or CDT-detected edge. Let $\hat{u}_{ij} = (u_i - u_j)$. We can compute the elements of the affinity matrix W as follows:

$$w_{ij} \stackrel{\text{def}}{=} \begin{cases} \dfrac{e^{-c\,\hat{u}_{ij}^T\,\Sigma^{-1}\,\hat{u}_{ij}}}{|\Sigma|^{1/2}}\; e^{-f(g_{ij})}, & i \neq j \wedge j \in P_{\text{nbr}}^i \\ \\ 0, & i = j \vee j \notin P_{\text{nbr}}^i \end{cases} \tag{1}$$

where $i = 0, 1, \ldots, n-1$, P_{nbr}^i is the list of indexes of the neighbors of v_i, Σ is the color sample variance-covariance matrix, $c > 0$ is an image-dependent constant and $f(g_{ij})$ is a function describing geometric affinity of p_i and p_j.

The Normalized Cuts graph clustering objective [13] is typically considered in its weak formulation

$$L\,z_i = \lambda_i\,D\,z_i \tag{2}$$

with the set of continuous solutions (λ_i, z_i), $i = 0, 1, \ldots$, where $L = D - W$, $L = L^T$ is the generalized graph Laplacian [2] and $D = \text{diag}(d_0, d_1, \ldots d_{n-1})$ is the diagonal matrix of weighted degrees $d_k = \sum_{i \neq k} w_{ki}$ of the graph nodes v_k. The generalized symmetric eigenvalue problem (2) is easily reduced to the spectrally equivalent symmetric eigenvalue problem

$$A\,x_i = \lambda_i\,x_i, \tag{3}$$

where $A = I - D^{-1/2}\,W\,D^{-1/2}$ is the normalized affinity matrix, I is the identity matrix and $x_i = D^{1/2}\,z_i$. Matrix A is symmetric positive semidefinite and consequently has nonnegative real spectrum. The smallest nonzero eigenvalue λ_1 of A corresponds to the optimal solution to the relaxed Normalized Cuts problem (2). Fiedler [1] calls λ_1 the algebraic connectivity of graph, as the corresponding eigenvector z_1, often called Fiedler vector [2], is the new representation of the original image, consisting of positive and negative components that correspond to the two nodal domains in graph G. Fiedler vector is typically used in spectral segmentation methods for recursive graph bipartitioning.

3 Numerical Findings

Although recursive application of the Fiedler vector gives analytically optimal solution to the segmentation problem, some authors often use a few additional vectors to perform spectral graph partitioning. For instance, Malik et al. [8] use a two-stage segmentation procedure that requires to partition the subspace consisting of the 11 eigenvectors corresponding to the 11 smallest nonzero eigenvalues in order to construct an oversegmented image followed by the Fiedler-based segmentation of the graph of the oversegmented image. We would like to understand whether, in finite precision, it is sufficient to compute high quality segmentation with the Fiedler vector alone, or whether we do need to generate multidimentional segmentation of the eigenvector subspace corresponding to the few of the smallest eigenvalues.

In order to understand how the analytical properties of the graph Laplacians change when their spectral characteristics are computed in the presence of rounding errors, we will analyze the first two smallest distinct eigenvalues and the corresponding eigenvectors of the affinity matrix. Our choice of the eigenvalue solver is the Lanczos Method with Guaranteed Accuracy (LMGA) [10] – an implementation of the Cullum-Willoughby-Lanczos method [9] that uses eigenvalue intervals computed using Interval Bisection method and the two-sided Sturm sequences [14,10,15] to accurately identify and discard spurious and numerically multiple eigenvalues. LMGA relies on the latest version of the two-sided Sturm sequence-based implementation of the Inverse Iteration method [10,15] – Inverse Iteration with Guaranteed Accuracy (IIGA) – to compute the eigenvectors of the underlining tridiagonal symmetric eigenvalue problem.

Let $(\tilde{\lambda}_i,\, \tilde{z}_i),\ i = 0, 1, \ldots$ be the finite precision solution to the generalized eigenvalue problem (2) computed using the solution $(\tilde{\lambda}_i,\, \tilde{x}_i)$ to the equivalent standard eigenvalue problem (3). In exact arithmetic, zero eigenvalue $\lambda_0 = 0$ is the trivial solution to the problem (2), as the corresponding eigenvector $z_0 = c\,(1, 1, \ldots, 1)^T$, where $c \neq 0$ is a scalar, contains equal components that would assign all graph vertexes to one nodal domain, or one segment. This is the property of all generalized graph Laplacians, whose eigenvectors behave very similarly to the modes of a vibrating string, with the first mode, corresponding to $\lambda_0 = 0$, assigning all graph vertexes to one nodal domain, the second mode splitting graph nodes into two domains, the third – into three domains, and so forth [16]. Due to the presence of rounding errors, approximate eiegnvector $\tilde{z}_0$ has distinct components $\tilde{z}_0^k$ dependent on the weighted degrees d_k of v_k:

$$\tilde{z}_0^k = \tilde{x}_0^k / \sqrt{d_k},\ k = 0, 1, \ldots, n-1. \tag{4}$$

We can model $\tilde{z}_0$ as the vector $\bar{z}_0$ with the components

$$\bar{z}_0^k = 1.0 + \zeta\, \|A\|_E\, \varepsilon_{\text{mach}} / \sqrt{d_k}, \tag{5}$$

where $\varepsilon_{\text{mach}}$ is the unit round-off error, $\|A\|_E$ is the Euclidean norm of A and $\zeta > 1$ is a scalar that accounts for the error of the algorithm used to compute the eigenvectors. We compute the eigenvectors $\tilde{x}_k,\ k = 0, 1, \ldots, n-1$ of the

underlining tridiagonal problem using Interval Bisection followed by the IIGA method. The error in the eigenvectors $\tilde{x}_i$ will be at least as high as the error of the computed eigenvalues, which is guaranteed not to exceed $\sqrt{3}\,\|A\|_E\,\varepsilon_{\text{mach}}$ [10]. In our case, ζ is at least $\sqrt{3}$, which means that $\bar{z}_0$ is likely to contain errors that slightly exceed the unit round-off. This means that, in general, components $\bar{z}_0^k$ are not identical, each inheriting information about the weighted degree

$$d_k = \sum_{i \neq k} w_{ki} \tag{6}$$

of the corresponding graph node v_k through the error term. Weighted degree d_k of the node v_k is the sum of the weights of the graph edges that node v_k is incident to. This means that d_k accumulates a wealth of information about adjacent graph nodes and incident edges of the node v_k. If we are looking for a nontrivial initial guess for our Lanczos eigensolver, vector $\bar{z}_0$ with an appropriately chosen parameter ζ certainly makes a good choice.

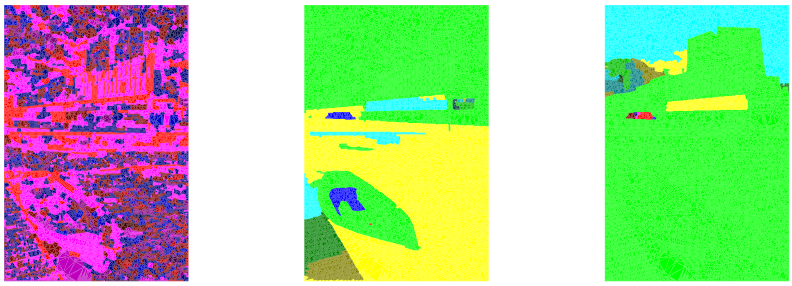
It is reasonable to assume that the error term $\zeta\,\|A\|_E\,\varepsilon_{\text{mach}}/\sqrt{d_k},\ \zeta > 1$ will be present in the components of all $\tilde{z}_i$. We can model all eigenvectors of the graph Laplacian as

$$\bar{z}_i^k = (\tilde{x}_i^k + \zeta\,\|A\|_E\,\varepsilon_{\text{mach}})/\sqrt{d_k}, \tag{7}$$

where $k = 0, 1, \ldots, n-1,\ i = 0, 1, \ldots$. It is clear that the error term will contribute additional information about the weighted degrees of the graph nodes into the floating point solution $\tilde{z}_i$. Consequently, we can expect that all $\tilde{z}_i$, including the Fiedler vector, may contain more than i nodal domains. Although the presence of the error term is desirable for computing an initial guess for the Krylov subspace eigensolver, it is likely to contaminate the Fiedler vector with the information about more than two nodal domains. This is probably the reason why in floating point arithmetic Fiedler vector alone often fails to deliver clear-cut results. As far as the use of the multidimensional eigenvector spaces for graph partitioning, the presence of the error term in the eigenvectors should not be damaging, as we are no longer looking for just two nodal domains. As a matter of fact, the information about more than i nodal domains in the i-th eigenvector in such a subspace may help to identify all nodal domains of interest without having to include a large number of eigenvectors.

We conducted a number of numerical experiments in order to verify our theory. We discovered that straightforward use of the formula (5) with $\zeta = 1.0$ produces extremely noisy results that can be seen in Figure 2(a), where we recursively bipartitioned vector $\bar{z}_0$ of the polygonized image presented in Figure 1(b). At the same time, it is clear that $\bar{z}_0$ has nontrivial components in the direction of the solution and we can take $\bar{z}_0$ as an initial guess for an iterative eigenvalue solver that can be computed in linear time. This means that we can solve the problem running LMGA to solve (3) with the initial iterate $\bar{x}_0$ with the following components

$$\bar{x}_0^k = \sqrt{d_k} + \zeta\,\|A\|_E\,\varepsilon_{\text{mach}}, \quad k = 0, 1, , \ldots, n-1 \tag{8}$$



(a) initial iterate $\bar{x}_0$ (b) first eigenvector $\hat{z}_0$ (c) Fiedler vector $\hat{z}_1$

Fig. 2. (a) Initial iterate $\bar{x}_0$ for LMGA, (b) rescaled eigenvector $\hat{z}_0$ ($\tilde{\lambda}_0 = 1.88469e-16$), (c) rescaled Fiedler vector $\hat{z}_1$ ($\tilde{\lambda}_1 = 1.53525e-06$)

that, due to the nature of the Normalized Cuts objective, incorporate information about the weighted degrees d_k of the corresponding graph nodes v_k.

Let $(\tilde{\lambda}_i, \tilde{z}_i)$, $i = 0, 1, \ldots, m-1$ be a set of m eigenpairs that we managed to compute with the LMGA such that $\tilde{\lambda}_0 < \tilde{\lambda}_1 < \cdots < \tilde{\lambda}_{m-1}$. We would like to examine the Fiedler vector $\tilde{z}_1$ and to verify our hypothesis that $\tilde{z}_0 \neq c\,\mathbf{1}$, $c \in \mathbb{R}$ may contain nontrivial information about the polygonized image P. In order to visualize our results we rescale components of $\tilde{z}_i$ as follows: $\hat{z}_i^k = (\tilde{z}_i^k - \min_j \tilde{z}_i^j)/(\max_j \tilde{z}_i^j - \min_j \tilde{z}_i^j)$ where $i = 0, \ldots, m-1$, $k = 0, \ldots, n-1$, so that $\hat{z}_i^k \in [0, 1]$. Next, we will apply a recursive bisection procedure to the rescaled vectors $\hat{z}_0$ and $\hat{z}_1$, splitting the vectors in two subvectors relative to the sample mean of its components. In Figure 2 we present results that we obtained in double IEEE-754 precision after 12 steps of the bisection procedure used to split and visualize vectors $\bar{z}_0$, $\hat{z}_0$ and $\hat{z}_1$ generated for the polygonized version of the image from the Berkeley Segmentation Data Set [12] shown in Figure 1. As expected, due to the presence of rounding errors, both the first (trivial) eigenvector $\hat{z}_0$, and the Fiedler vector $\hat{z}_1$, have more than two nodal domains, however, neither $\hat{z}_0$ (Figure 2(b)) nor $\hat{z}_1$ (Figure 2(c)) produce satisfactory segmentation. The corresponding eigenvalues are well separated: $\tilde{\lambda}_0 = 1.88469e-16$ represents numerical zero, while $\tilde{\lambda}_1 = 1.53525e-06$ is the nontrivial eigenvalue corresponding to the Fiedler vector.

Alternatively, we can attempt to construct segmentation using a set of the few of the lowest eigenvectors of the graph Laplacian. A multidimensional partitioning method can be used to split the computed eigenvector subspace into segments representing coherent visual scenes in an image. The larger the eigenvalue is that the eigenvector corresponds to, the more nodal domains in the image it will reveal. Note that the use of eigenvectors corresponding to larger eigenvalues has its downside – it may result in an oversegmentation of an image [8] due to the presence of a large number of nodal domains. Additionally, we can include the first, trivial, eigenvector into this set.

Depending on the implementation, multidimensional partitioning methods, such as the K-means and Mean-Shift [17], have computational complexity

comparable and even higher than that of sparse eigensolvers, including LMGA. As a low complexity alternative, we developed an implementation of the Mean-Shift algorithm that we call 'Spatially Truncated Mean Shift' (STMS). We define the STMS iteration as follows. Let $Z = (\hat{z}_0, \hat{z}_1, \ldots, \hat{z}_{m-1})$ denote rescaled eigenvectors corresponding the first m smallest eigenvalues of the matrix W (2). Let $Y = Z^T = (y_0, y_1, \ldots, y_{n-1})$, where $y_i \in \mathbb{R}^m$. Then Spatially Truncated Mean Shift iteration takes the following form:

$$y_i^{(\tau+1)} = \sum_{l \in P_{\text{nbr}}^i} \frac{e^{-\|y_i^{(\tau)} - y_l\|_2^2/(\sqrt{2\,\pi}\,\sigma_{il})}}{\sum_{j \in P_{\text{nbr}}^i} e^{-\|y_i^{(\tau)} - y_j\|_2^2/(\sqrt{2\,\pi}\,\sigma_{ij})}}\, y_i, \tag{9}$$

where $y_i^{(0)} \stackrel{\text{def}}{=} y_i,\ i = 0, 1, \ldots, n-1;\ \tau = 0, 1, \ldots;\ \sigma_{ij}$ is sample variance of the shifted iterate $y_i^{(\tau)} - y_j$ and P_{nbr}^i is the set of indices of the neighbors of the data point y_i representing graph vertex v_i. Spatial truncation amounts to computing weights using only neighboring data points y_i instead of all $n-1$ data points, that is, we are computing locally a reweighted solution. Since the $\hat{z}_i$ are distinct representations of the same image and each can be separately treated as a suboptimal solution to the segmentation problem, we expect that this scheme will suffice for the identification of distinct nodal domains. The computational complexity of the Spatially Truncated Mean Shift is $O(n\, p_{\text{nbr}}\, k\, \tau)$, where p_{nbr} is the size of the largest set of neighbors P_i^{nbr}, while the computational complexity of the classical Mean Shift method is $O(n^2\, k\, \tau)$. Since $p_{\text{nbr}} \ll n$ we should see a significant speedup.

Our segmentation procedure can be summarized as an algorithm that applies Spatially Truncated Mean Shift (9) to the eigenvector subspace generated with the Lanczos Method with Guaranteed Accuracy on a VISTA-preprocessed image. We can formalize this algorithm that we call 'Spectral Segmentation on Polygonal Grids', or SSPG, as follows:

Algorithm 1 (Spectral Segmentation on Polygonal Grids (SSPG)).

1. *Construct graph affinity matrix W of a VISTA-polygonized image.*
2. *Apply LMGA to solve the eigenvalue problem* (3) *with the initial iterate set to* (8) *to compute 5 of the smallest distinct eigenvalues and the corresponding eigenvectors of W.*
3. *Apply STMS* (9) *to partition appropriately rescaled eigenvectors, including the eigenvector corresponding to the trivial solution.*
4. *If the number of identified segments is smaller than the number of requested segments, return to the step 2, increasing the number of eigenvectors to be computed by 5.*
5. *Visualize the results by assigning the same color to the pixels that belong to the same cluster identified by STMS.*

Note that for the BSDS images Algorithm 1 finds the requested number of segments in one or two steps, while for more challenging large images, such as images from remote sensing applications, it typically converges in about three to five iterations. Iteratively repeating step 8 is relatively inexpensive as the LMGA does not use reorthogonalization.

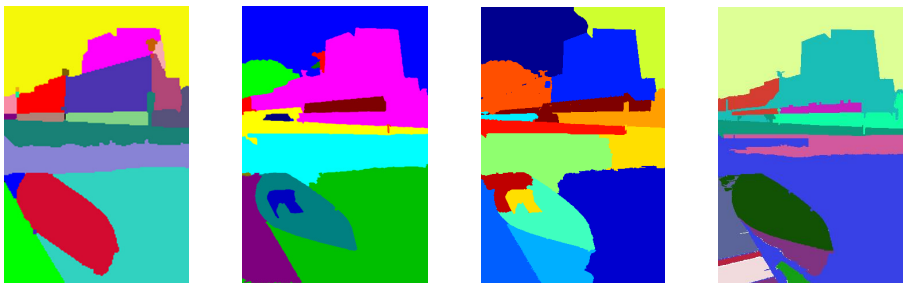
Table 1. Quantitative evaluation of segmentation results

Image	SSPG $(\rho_d, \rho_q, \rho_{cc}, \rho_{\mathbf{g}})$	MNC $(\rho_d, \rho_q, \rho_{cc}, \rho_{\mathbf{g}})$	SWA $(\rho_d, \rho_q, \rho_{cc}, \rho_{\mathbf{g}})$
1(a)	(0.47, 0.72, 0.90, **0.67)**	(0.53, 0.65, 0.97, **0.69)**	(0.65, 0.72, 0.62, **0.66)**
4(a)	(0.33, 0.64, 0.97, **0.59)**	(0.26, 0.59, 0.69, **0.47)**	(0.56, 0.68, 0.19, **0.41)**
4(b)	(0.70, 0.93, 0.19, **0.50)**	(0.70, 0.95, 0.28, **0.57)**	(0.50, 0.85, 0.89, **0.72)**
4(c)	(0.45, 0.51, 0.97, **0.58)**	(0.55, 0.85, 0.95, **0.76)**	(0.55. 0.76, 0.37, **0.53)**
4(d)	(0.41, 0.86, 0.83, **0.66)**	(0.32, 0.81, 0.92, **0.62)**	(0.55, 0.88, 0.12, **0.39)**

4 Experimental Results

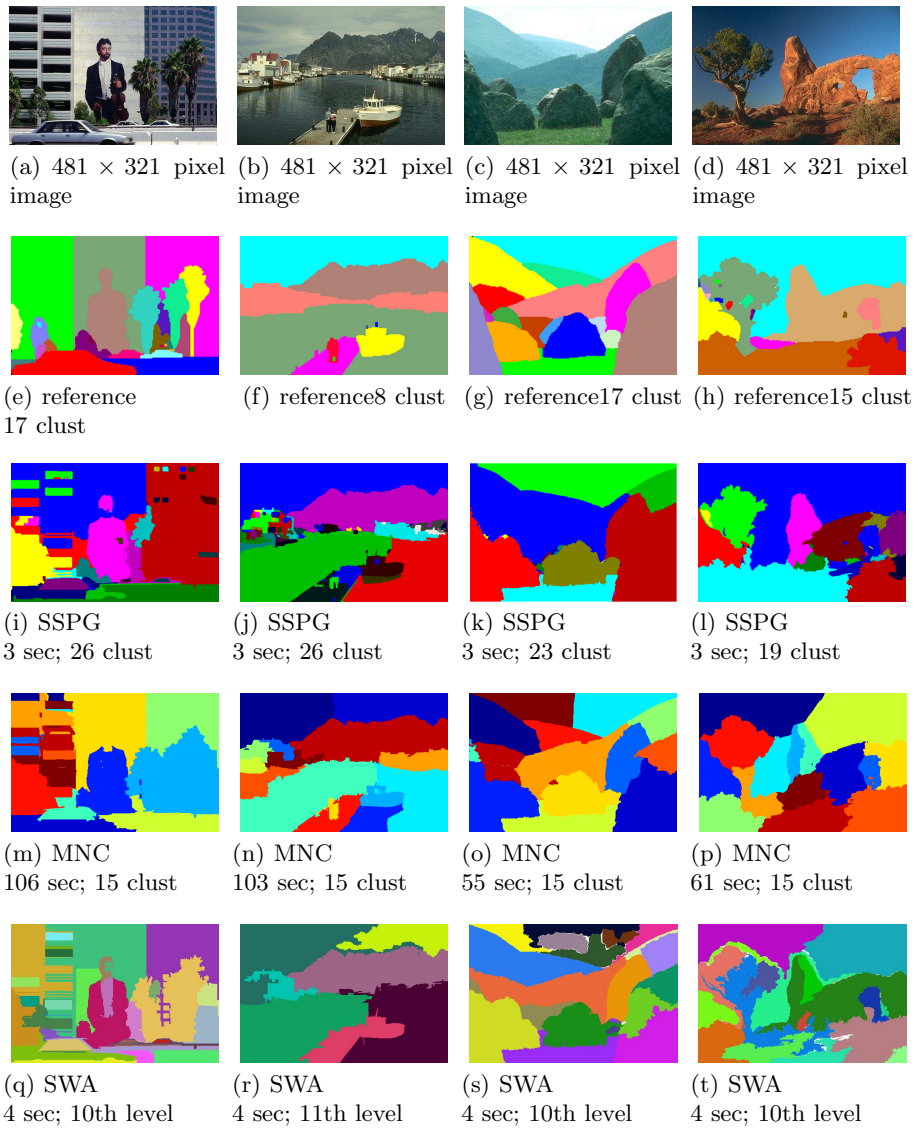
In order to evaluate the performance of the SSPG algorithm we carried out a set of tests that compares its C/C++ implementation to the Multiscale Normalized Cuts (MNC) [6] and to the Segmentation by Weighted Aggregation (SWA) [7]. We ran our experiments on a 2.4 GHz two dual-core AMD Opteron 64-bit workstation with 16 GB of RAM, and we used test images 1(a) and 4(a) – 4(d) from the Berkeley Segmentation Data Set (BSDS) [12]. We report SSPG and MNC execution times in the CPU seconds and in the wall-clock seconds for SWA, as it does not provide a built-in CPU timer. We report the number of clusters found by SSPG and MNC algorithms and the level at which SWA results had been produced. In Figure 3, we present segmentation of the 321×481 pixel BSDS image 1(a) produced by the SSPG, MNC and SWA algorithms along with the execution times of these algorithms in seconds. Likewise, in Figure 4 we present segmentation results and execution times for for the 481×321 pixel BSDS images 4(a) – 4(d).

Additionally, we carried out a quantitative evaluation of the segmentation results produced by the SSPG, MNC and SWA algorithms using the BSDS human segmentation (Figures 3(a), 4(e) – 4(h)) as the reference images. Our test results are summarized in Table 1, where for each test case we computed



(a) reference18 clust (b) SSPG 4 sec; 14 clust (c) MNC 92 sec; 15 clust (d) SWA 4 sec; 11th level

Fig. 3. (a) Test reference image (human segmentation from BSDS); (b) SSPG segmentation results; (c) MNC segmentation results; (d) SWA segmentation results



(a) 481 × 321 pixel image (b) 481 × 321 pixel image (c) 481 × 321 pixel image (d) 481 × 321 pixel image

(e) reference 17 clust (f) reference8 clust (g) reference17 clust (h) reference15 clust

(i) SSPG 3 sec; 26 clust (j) SSPG 3 sec; 26 clust (k) SSPG 3 sec; 23 clust (l) SSPG 3 sec; 19 clust

(m) MNC 106 sec; 15 clust (n) MNC 103 sec; 15 clust (o) MNC 55 sec; 15 clust (p) MNC 61 sec; 15 clust

(q) SWA 4 sec; 10th level (r) SWA 4 sec; 11th level (s) SWA 4 sec; 10th level (t) SWA 4 sec; 10th level

Fig. 4. (a) – (d) Original BSDS images; (e) – (h) test reference images (human segmentation from BSDS); (i) – (l) SSPG segmentation results; (m) – (p) MNC segmentation results; (q) – (t) SWA segmentation results

the following characteristics, commonly used for segmentation evaluation [18]: the detection rate $\rho_d = \sharp(S \cap R)/\sharp R$, the quality rate $\rho_q = |S \cap R|/|S \cup R|$, and the connectivity coefficient $\rho_{cc} = 2\,\min(\sharp S, \sharp R)/(\sharp S + \sharp R)$, where R is the reference image, $S = \cup S_i$ is a set of computed segments S_i such that $|S_i \cap R|/|S_i| > 0.5$, $\sharp(A)$ is the number of segments in A and $|A|$ is the number of pixels in A. Additionally, we report geometric mean $\rho_g = (\rho_d\,\rho_q\,\rho_{cc})^{1/3}$ as a characteristic of the overall quality of segmentation [18].

Upon the examination of the Figures 3 and 4 it is clear that the SSPG and SWA algorithms produce segmentation roughly 20-30 times faster than MNC. SSPG computes meaningful segmentation in time comparable and even superior to that of SWA. Analyzing Table 1, we can see that the quality of SSPG segmentation is overall comparable to that of MNC and SWA. Low values of ρ_{cc} produced in a few instances by all three algorithms are the consequence of the fact that some of the identified clusters may consist of a few disjoint segments.

5 Conclusions

We show that on polygonal grids the eigenvector subspace, corresponding to just a few of the lowest eigenvalues of the graph Laplacian, contains sufficient information necessary for obtaining meaningful segmentation, while the Fiedler vector alone is not always suitable for spectral segmentation. We also show that the eigenvector, corresponding to the trivial solution, often carries nontrivial information about the nodal domains in the image and can be used as an initial guess for an iterative eigensolver. Favorable performance of the developed algorithm is achieved through the polygonal coarsening at the preprocessing steps, the choice of the initial iterate for the LMGA, and the use of the STMS method on the postprocessing stage.

References

1. Fiedler, M.: Algebraic Connectivity of Graphs. Czechoslovak Math. J. 23(98), 298–305 (1973)
2. Biyikoğlu, T., Leydold, J., Stadler, P.F.: Laplacian Eigenvectors of Graphs. Perron-Frobenius and Fraber-Krahn Type Theorems, vol. 1915. Springer, Heidelberg (2007)
3. Dhillon, I.S., Guan, Y., Kulis, B.: Weighted Graph Cuts without Eigenvectors: a Multilevel Approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(11), 1944–1957 (2007)
4. Yu, S.X., Shi, J.: Multiclass Spectral Clustering. In: International Conference on Computer Vision, Nice, France, pp. 11–17 (2003)
5. Yu, S.X.: Segmentation using multiscale cues. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 70–77 (2004)
6. Cour, T., Benezit, F., Shi, J.: Spectral Segmentation with Multiscale Graph Decomposition. In: IEEE International Conference on Computer Vision and Pattern Recognition, CVPR (2005),
`http://www.seas.upenn.edu/~timothee/software/ncut_multiscale/ncut_multiscale.html`

7. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and Adaptivity in Segmenting Visual Scenes. Nature 442, 810–813 (2006), http://www.wisdom.weizmann.ac.il/~swa/index.html
8. Malik, J., Belongie, S., Leung, T., Shi, J.B.: Contour and Texture Analysis for Image Segmentation. International Journal of Computer Vision 43(1), 7–27 (2001)
9. Cullum, J.K., Willoughby, R.A.: Lanczos algorithms for large symmetric eigenvalue computations, vol. 1. Society for Industrial and Applied Mathematics, Philadelphia (2002)
10. Matsekh, A.M., Shurina, E.P.: On Computing Spectral Decomposition of Symmetric Matrices and Singular Value Decomposition of Unsymmetric Matrices with Guaranteed Accuracy. Optoelectronics, Instrumentation and Data Processing 43(2), 159–169 (2007)
11. Prasad, L., Skourikhine, A.N.: Vectorized Image Segmentation via Trixel Agglomeration. Pattern Recognition 39, 501–514 (2006)
12. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp. 416–423 (2001)
13. Shi, J.B., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
14. Godunov, S.K., Antonov, A.G., Kiriljuk, O.P., Kostin, V.I.: Guaranteed accuracy in numerical linear algebra. Kluwer Academic Publishers Group, Dordrecht (1993); ISBN 0-7923-2352-1. Translated and revised from the 1988 Russian original
15. Matsekh, A.M.: The Godunov-Inverse Iteration: a Fast and Accurate Solution to the Symmetric Tridiagonal Eigenvalue Problem. Applied Numerical Mathematics 54(2), 208–221 (2005)
16. Demmel, J.: Lecture Notes S267. U.C. Berkeley (1996), http://www.cs.berkeley.edu/~demmel/cs267
17. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach toward Feature Space Analysis. IEEE Trans. Pattern Analysis Machine Intell., 603–619 (2002)
18. Sturm, U., Weidner, U.: Further investigations on segmentation quality assessment for remote sensing applications. In: ISPRS Hannover Workshop 2009 High-Resolution Earth Imaging for Geospatial Information, Hannover, Germany (2009)