On Spectral Graph Drawing

Yehuda Koren

Dept. of Computer Science and Applied Mathematics
The Weizmann Institute of Science, Rehovot, Israel
yehuda@wisdom.weizmann.ac.il

Abstract. The spectral approach for graph visualization computes the layout of a graph using certain eigenvectors of related matrices. Some important advantages of this approach are an ability to compute optimal layouts (according to specific requirements) and a very rapid computation time. In this paper we explore spectral visualization techniques and study their properties. We present a novel view of the spectral approach, which provides a direct link between eigenvectors and the aesthetic properties of the layout. In addition, we present a new formulation of the spectral drawing method with some aesthetic advantages. This formulation is accompanied by an aesthetically-motivated algorithm, which is much easier to understand and to implement than the standard numerical algorithms for computing eigenvectors.

1 Introduction

A graph $G(V, E)$ is an abstract structure that is used to model a relation E over a set V of entities. Graph drawing is a standard means for visualizing relational information, and its ultimate usefulness depends on the readability of the resulting layout, that is, the drawing algorithm's ability to convey the meaning of the diagram quickly and clearly. To date, many approaches to graph drawing have been developed [4, 8]. There are many kinds of graph-drawing problems, such as drawing di-graphs, drawing planar graphs and others. Here we investigate the problem of drawing undirected graphs with straight-line edges. In fact, the methods that we utilize are not limited to traditional graph drawing and are also intended for general low dimensional visualization of a set of objects according to their pair-wise similarities (see, e.g., Fig. 2).

We have focused on spectral graph drawing methods, which construct the layout using eigenvectors of certain matrices associated with the graph. To get some feeling, we provide results for three graphs in Fig. 1. This spectral approach is quite old, originating with the work of Hall [6] in 1970. However, since then it has not been used much. In fact, spectral graph drawing algorithms are almost absent in the graph-drawing literature (e.g., they are not mentioned in the two books [4, 8] that deal with graph drawing). It seems that in most visualization research the spectral approach is difficult to grasp in terms of aesthetics. Moreover, the numerical algorithms for computing the eigenvectors do not possess an intuitive aesthetic interpretation.

We believe that the spectral approach has two distinct advantages that make it very attractive. First, it provides us with an exact solution to the layout problem, whereas almost all other formulations result in an NP-hard problem, which can only be approximated. The second advantage is computation speed. Spectral drawings can be computed

extremely fast as we have shown in [9]. This is very important because the amount of information to be visualized is constantly growing exponentially.

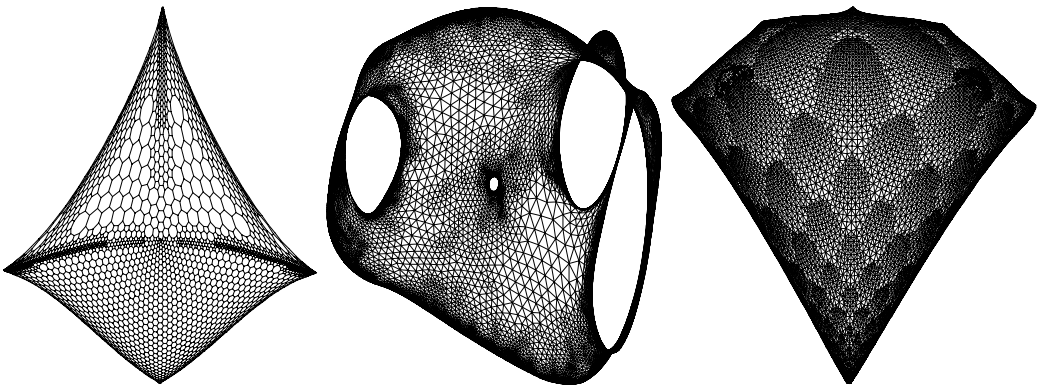

Fig. 1. Drawings obtained from the Laplacian eigenvectors. **(a)** The 4970 graph. $|V| = 4970$, $|E| = 7400$. **(b)** The 4elt graph. $|V| = 15606$, $|E| = 45878$. **(c)** The Crack graph. $|V| = 10240$, $|E| = 30380$.

Spectral methods have become standard techniques in algebraic graph theory; see, e.g., [3]. The most widely used techniques utilize eigenvalues and eigenvectors of the adjacency matrix of the graph. More recently, the interest has shifted somewhat to the spectrum of the closely related Laplacian. In fact, Mohar [11] claims that the Laplacian spectrum is more fundamental than this of the adjacency matrix.

Related areas where the spectral approach has been popularized include clustering [14], partitioning [12], and ordering [7]. However, these areas use discrete quantizations of the eigenvectors, unlike graph drawing, which employs the eigenvectors without any modification. Regarding this aspect, it is more fundamental to explore properties of graph-related eigenvectors in the framework of graph drawing.

In this paper we explore the properties of spectral visualization techniques, and provide different explanations for their ability to draw graphs nicely. Moreover, we have modified the usual spectral approach. The new approach uses what we will call *degree-normalized eigenvectors*, which have aesthetic advantages in certain cases. We provide an aesthetically-motivated algorithm for computing the degree-normalized eigenvectors. Our hope is that this will eliminate the vagueness of spectral methods and will contribute to their recognition as an important tool in the field of graph-drawing and information-visualization.

2 Basic Notions

A graph is usually written $G(V, E)$, where $V = \{1 \dots n\}$ is the set of n nodes, and E is the set of edges. Each edge $\langle i, j \rangle$ is associated with a non-negative weight w_{ij}

that reflects the similarity of nodes i and j. Thus, more similar nodes are connected with "heavier" edges. Henceforth, we will assume $w_{ij} = 0$ for any non-adjacent pair of nodes. Let us denote the neighborhood of i by $N(i) = \{j \mid \langle i, j\rangle \in E\}$. The degree of node i is $\mathrm{deg(i)} \stackrel{\mathrm{def}}{=} \sum_{\mathrm{j\in N(i)}} \mathrm{w_{ij}}$. Throughout the paper we have assumed, without loss of generality, that G is connected, otherwise the problem we deal with can be solved independently for each connected component.

The *adjacency-matrix* of the graph G is the symmetric $n \times n$ matrix A^G, where

$$A^G_{ij} = \begin{cases} 0 & i = j \\ w_{ij} & i \neq j \end{cases} \qquad i, j = 1, \ldots, n.$$

We will often omit the G in A^G.

The *Laplacian* is another symmetric $n \times n$ matrix associated with the graph, denoted by L^G, where

$$L^G_{ij} = \begin{cases} \mathrm{deg(i)} & i = j \\ -w_{ij} & i \neq j \end{cases} \qquad i, j = 1, \ldots, n.$$

Again, we will often omit the G in L^G.

The Laplacian has many interesting properties (see, e.g., [11]). Here we state some useful features:

- L is a real symmetric and hence its n eigenvalues are real and its eigenvectors are orthogonal.
- L is positive semi-definite and hence all eigenvalues of L are non-negative.
- $1_n \stackrel{\mathrm{def}}{=} (1, 1, \ldots, 1)^T \in \mathbb{R}^n$ is an eigenvector of L, with the associated eigenvalue 0.
- The multiplicity of the zero eigenvalue is equal to the number of connected components of G. In particular, if G is connected, then 1_n is the only eigenvector associated with eigenvalue 0.

The usefulness of the Laplacian stems from the fact that the quadratic form associated with it is just a weighted sum of all pairwise squared distances:

Lemma 1. *Let L be an $n \times n$ Laplacian, and let $x \in \mathbb{R}^n$. Then*

$$x^T L x = \sum_{i<j} w_{ij}(x_i - x_j)^2.$$

The proof of this lemma is direct.

Throughout the paper we will use the convention $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$ for the eigenvalues of L, and denote the corresponding real orthonormal eigenvectors by $v_1 = (1/\sqrt{n}) \cdot 1_n, v_2, \ldots, v_n$.

Let us define the *degrees matrix* as the $n \times n$ diagonal matrix D that satisfies $D_{ii} = \mathrm{deg(i)}$. Given a degrees matrix, D, and a Laplacian, L, then a vector u and a scalar μ are termed *generalized eigen-pairs* of (L, D) if $Lu = \mu Du$. Our convention is to denote the generalized eigenvectors of (L, D) by $\alpha \cdot 1_n = u_1, u_2, \ldots, u_n$, with corresponding generalized eigenvalues $0 = \mu_1 < \mu_2 \leqslant \cdots \leqslant \mu_n$. (Thus, $Lu_i = \mu_i Du_i,\ i = 1, \ldots, n$.) To uniquely define $u_1, u_2, \ldots, u_n$, we require them

to be D-normalized: so $u_i^T D u_i = 1,\ i = 1, \ldots n$. We term these generalized eigenvectors *the degree normalized eigenvectors*. It can be shown (see Appendix A) that all the generalized eigenvalues are real non-negative, and that all the degree normalized eigenvectors are D-orthogonal, i.e. $u_i^T D u_j = 0, \forall i \neq j$.

3 Spectral Graph Drawing

The earliest spectral graph-drawing algorithm was that of Hall [6]; it uses the low eigenvectors of the Laplacian. Henceforth, we will refer to this method as *the eigen-projection method*. Later, a similar idea was suggested in [13], where the results are shown to satisfy several desired aesthetic properties. A few other researchers utilize the top eigenvectors of the adjacency matrix instead of those of the Laplacian. For example, consider the work of [10], which uses the adjacency matrix eigenvectors to draw molecular graphs. Recently, eigenvectors of a modified Laplacian were used in [1] for the visualization of bibliographic networks.

In fact, for regular graphs of uniform degree deg, the eigenvectors of the Laplacian equal those of the adjacency matrix, but in a reversed order, because $L = deg \cdot I - A$, and adding the identity matrix does not change eigenvectors. However, for non-regular graphs, use of the Laplacian is based on a more solid theoretical basis, and in practice also gives nicer results than those obtained by the adjacency matrix. Hence, we will focus on visualization using eigenvectors of the Laplacian.

3.1 Derivation of the Eigen-Projection Method

We will introduce the eigenprojection method as a solution to a minimization problem. We begin by deriving a 1-D drawing, and then we show how to draw in more dimensions.

Given a weighted graph $G(V, E)$, we denote its 1-D layout by $x \in \mathbb{R}^n$, where $x(i)$ is the location of node i. We take x as the solution of the following constrained minimization problem

$$\begin{aligned} &\min_x \ E(x) \stackrel{\text{def}}{=} \sum_{\langle i,j \rangle \in E} w_{ij}(x(i) - x(j))^2 \\ &\text{given: } \mathrm{Var}(x) = 1, \end{aligned} \tag{1}$$

where $\mathrm{Var}(x)$ is the *variance* of x, defined as usual by $\mathrm{Var}(x) = \frac{1}{n}\sum_{i=1}^{n}(x(i) - \bar{x})^2$, and where $\bar{x}$ is the mean of x.

The energy to be minimized, $E(x)$, strives to make edge lengths short. Since the sum is weighted by edge-weights, "heavy" edges have a stronger impact and hence will be typically shorter. The constraint $\mathrm{Var}(x) = 1$ requires that the nodes be scattered in the drawing area, and prevents an overcrowding of the nodes at the same point. Note that the choice of variance 1 is arbitrary, and simply states the scale of the drawing. We could equally have chosen a constraint of the form $\mathrm{Var}(x) = c$. In this way, if x_0 is the optimal solution of variance 1, then $\sqrt{c} \cdot x_0$ is the optimal solution of variance c. Such a representation of the problem reminds the *force-directed graph drawing approach*

(see [4, 8]), where the energy to be minimized replaces the "attractive forces", and the variance constraint takes the role of the "repulsive forces".

The energy and the constraint are invariant under translation (ensure that for every α: $E(x) = E(x + \alpha \cdot 1_n)$, $\mathrm{Var}(x) = \mathrm{Var}(x + \alpha \cdot 1_n)$). We eliminate this degree of freedom by requiring that the mean of x is 0, i.e. $\sum_{i=1}^{n} x(i) = x^T \cdot 1_n = 0$. This is very convenient since now the variance can be written in a simple form: $\mathrm{Var}(x) = \frac{1}{n} x^T x$. To simplify the notation we will change the scale, and require the variance to be $\frac{1}{n}$, which is equivalent to $x^T x = \sum_{i=1}^{n} x(i)^2 = 1$.

Using Lemma 1, we can write the energy in a matrix form: $E(x) = x^T L x = \sum_{\langle i,j \rangle \in E} w_{ij} \cdot (x(i) - x(j))^2$. Now the desired 1-D layout, x, can be described as the solution of the constrained minimization problem

$$\begin{aligned} &\min_x x^T L x \\ &\text{given: } x^T x = 1 \\ &\text{in the subspace: } x^T \cdot 1_n = 0. \end{aligned} \tag{2}$$

Let us substitute $B = I$ in Claim A (in Appendix A), to obtain the optimal solution $x = v_2$, the second smallest eigenvector of L. The resulting value of the energy is λ_2, the corresponding eigenvalue.

To achieve a 2-D drawing, we need to compute an additional vector of coordinates, y. Our requirements for y are the same as those that we required from x, but in addition there must be no correlation between y and x, so that the additional dimension will provide us with as much new information as possible[1]. Since x and y are centered, we simply have to require that $y^T \cdot x = y^T \cdot v_2 = 0$. Hence y is the solution of the constrained minimization problem

$$\begin{aligned} &\min_y y^T L y \\ &\text{given: } y^T y = 1 \\ &\text{in the subspace: } y^T \cdot 1_n = 0, y^T \cdot v_2 = 0. \end{aligned} \tag{3}$$

Again, use Claim A so that the optimal solution is $y = v_3$, the third smallest eigenvector of L. The resulting value of the energy is λ_3, the corresponding eigenvalue.

In order to obtain a k-D drawing of the graph, we take the first coordinate of the nodes to be v_2, the second coordinate to be v_3, and in general, we define the i-th coordinate of the nodes by v_{i+1}.

3.2 Tuning the aspect-ratio

The requirement of unit variance in each axis, forces the nodes to be equally scattered along each of the axes. In this sense, the drawing has a perfectly balanced aspect ratio. Yet, sometimes a graph structure would be better visualized with non-balanced aspect

[1] The strategy to require no correlation between the axes is used in many other visualization techniques like Principal Components Analysis [16] and Classical Multidimensional Scaling [16].

ratio. This can be achieved by replacing the minimization problem (1) by the following maximization problem

$$\begin{aligned} &\max_{x} \ \mathrm{Var}(x) \\ &\text{given: } x^T L x = 1. \end{aligned} \tag{4}$$

Here we maximize the scatter of the nodes, while fixing the energy. Like before, if additional dimensions are required, the same maximization problem should be solved, constraining the solutions to be uncorrelated to the previous ones.

By the same reasoning as above, one can prove that according to this new formulation, the optimal k-D drawing of the graph is obtained by taking the coordinates of the nodes to be the eigenvectors:

$$\frac{1}{\sqrt{\lambda_2}} v_2, \ \frac{1}{\sqrt{\lambda_3}} v_3, \ldots, \frac{1}{\sqrt{\lambda_{k+1}}} v_{k+1} \ .$$

The length of an axis serves as a measure to its importance. Thus, axes with a lower energy will have a stronger impact.

4 Drawing using Degree-Normalized Eigenvectors

In this section we introduce a new spectral graph drawing method that associates the coordinates with some generalized eigenvectors of the Laplacian.

Suppose that we weight nodes by their degrees, so the mass of node i is its degree — $\deg(\mathrm{i})$. Now if we take the original constrained minimization problem (2) and weight sums according to node masses, we get the following degree-weighted constrained minimization problem (where D is the degrees matrix)

$$\begin{aligned} &\min_{x} x^T L x \\ &\text{given: } x^T D x = 1 \\ &\text{in the subspace: } x^T D 1_n = 0. \end{aligned} \tag{5}$$

Substitute $B = D$ in Claim A to obtain the optimal solution $x = u_2$, the second smallest generalized eigenvector of (L, D). The resulting energy is μ_2, the corresponding generalized eigenvalue.

Using the same reasoning as in Subsection 3.1, we obtain a k-D drawing of the graph, by taking the first coordinate of the nodes to be u_2, the second coordinate to be u_3, and in general, we define the i-th coordinate of the nodes by u_{i+1}.

We will show by several means that using these degree-normalized eigenvectors is more natural than using the eigenvectors of the Laplacian. In fact Shi and Malik [14] have already shown that the degree-normalized eigenvectors are more suitable for the problem of image segmentation. For the visualization task, the motivation and explanation are very different.

In order to gain some intuition on (5), we shall rewrite it in the equivalent form:

$$\min_x \frac{x^T L x}{x^T D x} \quad \text{in the subspace: } x^T D 1_n = 0. \tag{6}$$

It is straightforward to show that a solution of (5) is also a solution of (6).

In problem (6) the denominator moderates the behavior of the numerator, as we are showing now. The numerator strives to place those nodes with high degrees at the center of the drawing, so that they are in proximity to the other nodes. On the other hand, the denominator also emphasizes those nodes with high degrees, but in the reversed way: it strives to enlarge their scatter. The combimation of these two opposing goals, helps in making the drawing more balanced, preventing a situation in which nodes with lower degrees are overly separated from the rest nodes.

Another observation is that degree-normalized eigenvectors unify the two common spectral techniques: the approach that uses the Laplacian and the approach that uses the adjacency matrix.

Claim. The generalized eigenvectors of (L, D) are also the generalized eigenvectors of (A, D), with a reversed order.

Proof. Utilize the fact that $L = D - A$. Take u, a generalized eigenvector of (L, D). The vector u satisfies $(D - A)u = \mu Du$, or equivalently, by changing sides, $Au = Du - \mu Du$. This implies that

$$Au = (1 - \mu)Du\,.$$

The proof in the other direction is performed similarly.
Thus, A and L have the same D-normalized eigenvectors, although the order of eigenvalues is reversed. □

In this way, when drawing with degree normalized eigenvectors, we can take either the low generalized eigenvectors of the Laplacian, or the top generalized eigenvectors of the adjacency matrix, without affecting the result. (Remark: In this paper when referring to "top" or "low" eigenvectors, we often neglect the topmost (or lowest) degenerate eigenvector $\alpha \cdot 1_n$.)

The degree-normalized eigenvectors are also the (non-generalized) eigenvectors of the matrix $D^{-1}A$. This can be obtained by left-multiplying the generalized eigen-equation $Ax = \mu Dx$ by D^{-1}, obtaining the eigen-equation

$$D^{-1}Ax = \mu x. \tag{7}$$

Note that $D^{-1}A$ is known as the *transition matrix* of a random walk on the graph G. Hence, the degree-normalized eigen-projection uses the top eigenvectors of the transition matrix to draw the graph.

Regarding drawing quality, for graphs that are close to being regular, we have observed not much difference between drawing using eigenvectors and drawing using degree-normalized eigenvectors. However, when there are marked deviations in node

degrees, the results are quite different. This can be directly seen by posing the problem as in (6). Here, we provide an alternative explanation based on (7). Consider the two edges e_1 and e_2. Edge e_1 is of weight 1, connecting two nodes, each of which is of degree 10. Edge e_2 is of weight 10, connecting two nodes, each of which is of degree 100. In the Laplacian matrix, the entries corresponding to e_2 are 10 times larger than those corresponding to e_1. Hence we expect the drawing obtained by the eigenvectors of the Laplacian, to make the edge e_2 much shorter than e_1 (here, we do not consider the effect of other nodes that may change the lengths of both edges). However, for the transition matrix in (7), the entries corresponding to these two edges are the same, hence we treat them similarly and expect to get the same length for both edges. This reflects the fact that the *relative* importance of these two edges is the same, i.e. $\frac{1}{10}$.

In many kinds of graphs numerous scales are embedded, which indicates the existence of dense clusters and sparse clusters. In a traditional eigen-projection drawing, dense clusters are drawn extremely densely, while the whole area of the drawing is used to represent sparse clusters or outliers. This is the best way to minimize the weighted sum of square edge lengths, while scattering the nodes as demanded. A better drawing would allocate each cluster an adequate area. Frequently, this is the case with the degree normalized eigenvectors that adjust the edge weights in order to reflect their relative importance in the related local scale.

For example, consider Fig. 2, where we visualize 300 odors as measured by an electronic nose. Computation of the similarities between the odors is given in [2]. The odors are known to be classified into 30 groups, which determine the color of each odor in the figure. Figure 2(a) shows the visualization of the odors by the eigenvectors of the Laplacian. As can be seen, each of the axes shows one outlying odor, and places all the other odors about at the same location. However, the odors are nicely visualized using the degree normalized eigenvectors, as shown in Fig. 2(b).

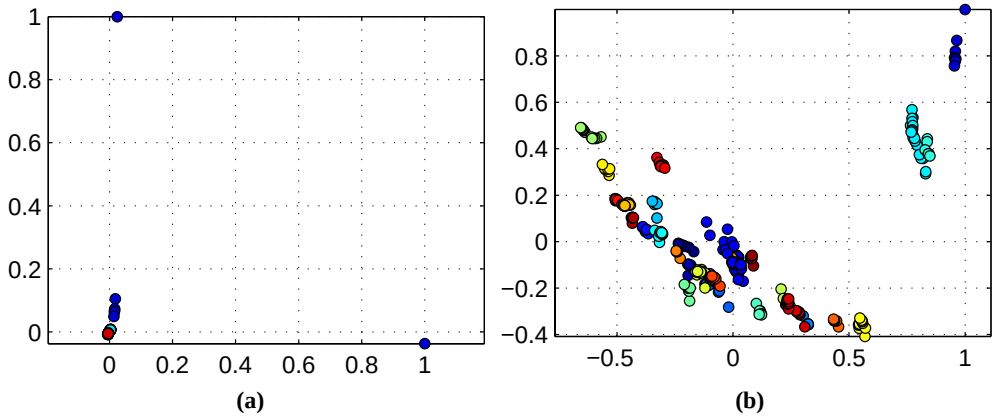

Fig. 2. Visualization of 300 odor patterns as measured by an electronic nose. **(a)** A drawing using the eigenvectors of the Laplacian. **(b)** A drawing using the degree-normalized eigenvectors.

5 An Optimization Process

An attractive feature of the degree-normalized eigenvectors is that they can be computed by an intuitive algorithm, which is directly related to their aesthetic properties. This is unlike the (non generalized) eigenvectors, which are computed using relatively complicated methods that are difficult to interpret in aesthetic terms. Now we will derive the algorithm.

Let i be a node. Differentiating $E(x)$ with respect to $x(i)$ gives

$$\frac{\partial E}{\partial x(i)} = 2 \sum_{j \in N(i)} w_{ij}(x(i) - x(j)).$$

Equating this to zero and isolating $x(i)$ we get

$$x(i) = \frac{\sum_{j \in N(i)} w_{ij} x(j)}{\deg(\mathrm{i})} \ .$$

Hence, when allowing only node i to move, the location of i that minimizes $E(x)$ is the weighted centroid of i's neighbors.

This induces an optimization process that iteratively puts each node at the weighted centroid of its neighbors (simultaneously for all nodes). The aesthetic reasoning is clear. A rather impressive fact is that when initialized with a vector D-orthogonal to 1_n, this algorithm converges *in the direction* of a non-degenerate degree-normalized eigenvector of L. More precisely, this algorithm converges either in the direction of u_2 or that of u_n.

We can prove this surprising fact by observing that the action of putting each node at the weighted centroid of its neighbors is equivalent to multiplication by the transition matrix — $D^{-1}A$. Thus, the process we have described can be expressed in a compact form as the sequence

$$\begin{cases} x_0 & = \text{random vector, s.t. } x_0^T D 1_n = 0 \\ x_{i+1} & = D^{-1} A x_i \ . \end{cases}$$

This process is known as the Power-Iteration [5]. In general, it computes the "dominant" eigenvector of $D^{-1}A$, which is the one associated with the largest-in-magnitude eigenvalue. In our case, all the eigenvectors are D-orthogonal to the "dominant" eigenvector — 1_n, and also the initial vector, x_0, is D-orthogonal to 1_n. Thus, the series converges in the direction of the next dominant eigenvector, which is either u_2, which has the largest positive eigenvalue, or u_n, which possibly has the largest negative eigenvalue. (We assume that x_0 is not D-orthogonal to u_2 or to u_n, which is nearly always true for a randomly chosen x_0)

In practice, we want to ensure convergence to u_2 (avoiding convergence to u_n). We use the fact that all the eigenvalues of the transition matrix are in the range $[-1, 1]$. This can be proved directly using the Gershgorin bound on eigenvalues [5], since in $D^{-1}A$ all entries on the diagonal are 0, and the sum of each row is 1. Now it is possible to shift the eigenvalues by adding the value 1 to each of them, so that they are all positive, thus preventing convergence to an eigenvector with a large negative eigenvalue. This is

done by working on the matrix $I + D^{-1}A$ instead of the matrix $D^{-1}A$. In this way the eigenvalues are in the range $[0, 2]$, while eigenvectors are not changed. In fact, it would be more intuitive to scale the eigenvalues to the range $[0, 1]$, so we will actually work with the matrix $\frac{1}{2}(I + D^{-1}A)$. If we use our initial "intuitive" notions, this means a more careful process. In each iteration, we put each node at the average between its old place and the centroid of its neighbors. Thus, each node absorbs its new location not only from its neighbors, but also from its current location.

The full algorithm for computing a k-D drawing is given in Fig. 3. To compute a degree-normalized eigenvector u_j, we will use the principles of the power-iteration and the D-orthogonality of the eigenvectors. Briefly, we pick some random x, such that x is D-orthogonal to $u_1, \ldots, u_{j-1}$, i.e. $x^T D u_1 = 0, \ldots, x^T D u_{j-1} = 0$. Then, if $x^T D u_j \neq 0$, it can be proved that the series $\frac{1}{2}(I + D^{-1}A)x, (\frac{1}{2}(I + D^{-1}A))^2 x, (\frac{1}{2}(I + D^{-1}A))^3 x, \ldots$ converges in the direction of u_j. Note that in theory, all the vectors in this series are D-orthogonal to $u_1, \ldots, u_{j-1}$. However, to improve numerical stability, our implementation imposes the D-orthogonality to previous eigenvectors in each iteration. The power iteration algorithm produces vectors of diminishing (or exploding) norms. Since we are only interested in convergence *in direction*, it is customary to re-scale the vectors after each iteration. Here, we will re-scale by normalizing the vectors to be of length 1.

Function SpectralDrawing (G – the input graph, k – dimension)
% This function computes $u_2, \ldots, u_k$, the top (non-degenerate) eigenvectors of $D^{-1}A$.
  const $\epsilon \leftarrow 10^{-7}$ % tolerance
  for $i = 2$ to k **do**
    $\hat{u}_i \leftarrow$ random % random initialization
    $\hat{u}_i \leftarrow \frac{\hat{u}_i}{\|\hat{u}_i\|}$
    do
      $u_i \leftarrow \hat{u}_i$
      % D-Orthogonalize against previous eigenvectors:
      for $j = 1$ to $i - 1$ **do**
        $u_i \leftarrow u_i - \frac{u_i^T D u_j}{u_j^T D u_j} u_j$
      end for
      % multiply with $\frac{1}{2}(I + D^{-1}A)$:
      for $j = 1$ to n **do**
        $\hat{u}_i(j) \leftarrow \frac{1}{2} \cdot \left(u_i(j) + \frac{\sum_{k \in N(j)} w_{jk} u_i(k)}{\deg(j)} \right)$
      end for
      $\hat{u}_i \leftarrow \frac{\hat{u}_i}{\|\hat{u}_i\|}$ % normalization
    while $\hat{u}_i \cdot u_i < 1 - \epsilon$ % halt when direction change is negligible
    $u_i \leftarrow \hat{u}_i$
  end for
  return $u_2, \ldots, u_k$

Fig. 3. The algorithm for computing degree-normalized eigenvectors

The convergence rate of this algorithm when computing u_i is dependent on the ratio μ_i/μ_{i+1}. In practice, we embedded this algorithm in a multi-scale construction, resulting in extremely fast convergence. Further details on the multi-scale scheme are given in [9].

6 A Direct Characterization of Spectral Layouts

So far, we have derived spectral methods as solutions of optimization problems, or as a limit of a drawing process. In this section we characterize the eigenvectors themselves, in a rather direct manner, to clarify the aesthetic properties of the spectral layout. Once again the degree-normalized eigenvectors will appear as the more natural way for spectral graph drawing.

As we have seen, the quadratic form $E(x) = \sum_{\langle i,j\rangle \in E} w_{ij}(x(i) - x(j))^2$, which motivates spectral methods, is tightly related to the aesthetic criterion that calls for placing each node at the weighted centroid of its neighbors. When the graph is connected, it can be strictly achieved only by the degenerate solution that puts all nodes at the same location. Hence, to incorporate this aesthetic criterion into a graph drawing algorithm, it should be modified appropriately.

Presumably the earliest graph drawing algorithm, formulated by Tutte [15], is based on placing each node on the weighted centroid of its neighbors. To avoid the degenerate solution, Tutte arbitrarily chose a certain number of nodes to be anchors, i.e. he fixed their coordinates in advance. Those nodes are typically drawn on the boundary. This, of course, prevents the collapse; however it raises new problems, such as which nodes should be the anchors, how to determine their coordinates, and why after all such an anchoring mechanism should generate nice drawings. An advantage of Tutte's method is that in certain cases, it can guarantee achieving a planar drawing (i.e., without edge intersections).

Tutte treats in different ways the anchored nodes and the remaining nodes. Whereas the remaining nodes are located exactly at the centroid of their neighbors, nothing can be said about anchored nodes. In fact, in several experiments we have seen that the anchored nodes are located quite badly.

Alternatively, we do not use different strategies for dealing with two kinds of nodes, but rather, we treat all the nodes similarly. The idea is to gradually increase the deviations from centroids of neighbors as we move away from the origin (that is the center of the drawing). This reflects the fact that central nodes can be placed exactly at their neighbors' centroid, whereas boundary nodes must be shifted outwards.

More specifically, node i, which is located in place $x(i)$, is shifted from the center toward the boundary by the amount of $\mu \cdot |x(i)|$, for some $\mu > 0$. Formally, we request the layout x to satisfy, for every $1 \leqslant i \leqslant n$

$$x(i) - \frac{\sum_{j \in N(i)} w_{ij} x(j)}{\deg(\mathrm{i})} = \mu \cdot x(i)\,.$$

Note that the deviation from the centroid is always toward the boundary, i.e. toward $+\infty$ for positive $x(i)$ and toward $-\infty$ for negative $x(i)$. In this way we prevent a collapse at the origin.

We can represent all these n requests compactly in a matrix form, by writing

$$D^{-1}Lx = \mu x\,.$$

Left-multiplying both sides by D, we obtain the familiar generalized eigen-equation

$$Lx = \mu Dx\,.$$

We conclude with the following important property of degree-normalized eigenvectors:

Proposition 1. *Let u be a generalized eigenvector of (L, D), with associated eigenvalue μ. Then, for each i, the exact deviation from the centroid of neighbors is*

$$u(i) - \frac{\sum_{j \in N(i)} w_{ij} u(j)}{\deg(\mathrm{i})} = \mu \cdot u(i)\,.$$

Note that the eigenvalue μ is a scale-independent measure of the amount of deviation from the centroids. This provides us with a fresh new interpretation of the eigenvalues that is very different from the one given in Subsection 3.1, where the eigenvalues were shown as the amount of energy in the drawing.

Thus, we deduce that the second smallest degree-normalized eigenvector produces the non-degenerate drawing with the smallest deviations from centroids, and that the third smallest degree-normalized eigenvector is the next best one and so on.

Similarly, we can obtain a related result for eigenvectors of the Laplacian:

Proposition 2. *Let v be an eigenvector of L, with associated eigenvalue λ. Then, for each i, the exact deviation from the centroid of neighbors is*

$$v(i) - \frac{\sum_{j \in N(i)} w_{ij} v(j)}{\deg(\mathrm{i})} = \lambda \cdot \deg(\mathrm{i})^{-1} \cdot v(i)\,.$$

Hence for eigenvectors of the Laplacian, the deviation between a node and the centroid of its neighbors gets larger as the node's degree decreases.

7 Discussion

In this paper we have presented a spectral approach for graph drawing, and justified it by studying three different viewpoint for the problem. The first viewpoint describes a classical approach for achieving graph layouts by solving a constrained energy minimization problem. This is much like force directed graph drawing algorithms (for a survey refer to [4, 8]). Compared with other force-directed methods, the spectral approach has two major advantages: **(1)** Its global optimum can be computed efficiently. **(2)** The energy function contains only $O(|E|)$ terms, unlike the $O(n^2)$ terms appearing in almost all the other force-directed methods.

A second viewpoint shows that the spectral drawing is the limit of an iterative process, in which each node is placed at the centroid of its neighbors. This viewpoint does not only sharpen the nature of spectral drawing, but also provides us with an

aesthetically-motivated algorithm. This is unlike other algorithms for computing eigenvectors, which are rather complicated and far from having an aesthetic interpretation.

We have also introduced a third viewpoint, showing that spectral methods place each node at the centroid of its neighbors with some well defined deviation. This new interpretation provides an accurate and simple description of the aesthetic properties of spectral drawing.

Another contribution of our paper is the introduction of a new spectral graph drawing algorithm, using what we have called *degree-normalized eigenvectors*. We have shown that this method is more natural in some aspects, and has aesthetic advantages for certain kinds of data.

References

1. U. Brandes and T. Willhalm, "Visualizing Bibliographic Networks with a Reshaped Landscape Metaphor", Proc. 4th Joint Eurographics - IEEE TCVG Symp. Visualization (VisSym '02), pp. 159-164, ACM Press, 2002.
2. L. Carmel, Y. Koren and D. Harel, "Visualizing and Classifying Odors Using a Similarity Matrix", Proceedings of the ninth International Symposium on Olfaction and Electronic Nose (ISOEN'02), IEEE, to appear, 2003.
3. F.R.K. Chung, *Spectral Graph Theory*, CBMS Reg. Conf. Ser. Math. 92, American Mathematical Society, 1997.
4. G. Di Battista, P. Eades, R. Tamassia and I.G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall, 1999.
5. G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.
6. K. M. Hall, "An r-dimensional Quadratic Placement Algorithm", *Management Science* **17** (1970), 219–229.
7. M. Juvan and B. Mohar, "Optimal Linear Labelings and Eigenvalues of Graphs", *Discrete Applied Math.* **36** (1992), 153–168.
8. M. Kaufmann and D. Wagner (Eds.), *Drawing Graphs: Methods and Models*, LNCS 2025, Springer Verlag, 2001.
9. Y. Koren, L. Carmel and D. Harel, "ACE: A Fast Multiscale Eigenvectors Computation for Drawing Huge Graphs", *Proceedings of IEEE Information Visualization 2002 (InfoVis'02)*, IEEE, pp. 137–144, 2002.
10. D.E. Manolopoulos and P.W. Fowler, "Molecular Graphs, Point Groups and Fullerenes", *J. Chem. Phys.* **96** (1992), 7603–7614.
11. B. Mohar, "The Laplacian Spectrum of Graphs", *Graph Theory, Combinatorics, and Applications* **2** (1991), 871–898.
12. A. Pothen, H. Simon and K.-P. Liou, "Partitioning Sparse Matrices with Eigenvectors of Graphs", *SIAM Journal on Matrix Analysis and Applications*, **11** (1990), 430–452.
13. J. Shawe-Taylor and T. Pisanski, "Characterizing Graph Drawing with Eigenvectors", Technical Report CSD-TR-93-20 (Royal Holloway, University of London, Department of Computer Science, Egham, Surrey TW20 0EX, England).
14. J. Shi and J. Malik, "Normalized Cuts and Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (2000), 888–905.
15. W. T. Tutte, "How to Draw a Graph", *Proc. London Math. Society* **13** (1963), 743–768.
16. A. Webb, *Statistical Pattern Recognition*, Arnold, 1999.

A Solution of Constrained Quadratic Optimization Problems

In this appendix we study a certain kind of constrained optimization problem, whose solution is a generalized eigenvector.

We use two matrices: **(1)** A — an $n \times n$ real symmetric positive-semidefinite matrix. **(2)** B — an $n \times n$ diagonal matrix, whose diagonal entries are real-positive. (In fact, it is enough to require that matrix B is positive-definite.) We denote the generalized eigenvectors of (A, B) by $u_1, u_2, \ldots, u_n$, with corresponding eigenvalues $0 \leqslant \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$. Thus, $Au_i = \lambda_i B u_i,\ i = 1, \ldots, n$. To uniquely define $u_1, u_2, \ldots, u_n$, we require them to be B-normalized, i.e. $u_i^T B u_i = 1,\ i = 1, \ldots, n$.

Clearly, for every $1 \leqslant i \leqslant n$, $B^{\frac{1}{2}} u_i$ and λ_i are an eigen-pair of the matrix $B^{-\frac{1}{2}} A B^{-\frac{1}{2}}$. Note that $B^{-\frac{1}{2}} A B^{-\frac{1}{2}}$ is a symmetric positive-semidefinite. Thus, all the eigenvalues are real non-negative, and all the generalized eigenvectors are B-orthogonal, i.e. $u_i^T B u_j = 0$, $\forall i \neq j$.

Now we define a constrained optimization problem

$$\begin{aligned} &\min_x x^T A x \\ &\text{given: } x^T B x = 1 \\ &\text{in the subspace: } x^T B u_1 = 0, \ldots, x^T B u_{k-1} = 0. \end{aligned} \tag{8}$$

Claim. The optimal solution of problem 8 is $x = u_k$, with an associated cost of $x^T A x = \lambda_k$.

Proof. By using the B-orthogonality of $u_1, \ldots, u_n$, we can decompose every $x \in \mathbb{R}^n$ as a linear combination where $x = \sum_{i=1}^n \alpha_i u_i$. Moreover, since the solution is constrained to be B-orthogonal to $u_1, \ldots, u_{k-1}$, we can restrict ourselves to linear combinations of the form $x = \sum_{i=k}^n \alpha_i u_i$.

We use the constraint $x^T B x = 1$ to obtain

$$1 = x^T B x = \left(\sum_{i=k}^n \alpha_i u_i\right)^T B \left(\sum_{i=k}^n \alpha_i u_i\right) = \left(\sum_{i=k}^n \alpha_i u_i\right)^T \left(\sum_{i=k}^n \alpha_i B u_i\right) =$$
$$= \sum_{i=k}^n \sum_{j=k}^n \alpha_i u_i \alpha_j B u_j = \sum_{i=k}^n \sum_{j=k}^n \alpha_i \alpha_j u_i B u_j = \sum_{i=k}^n \alpha_i^2.$$

The last equation stems from the B-orthogonality of $u_1, u_2, \ldots, u_n$, and from defining these vectors as B-normalized.

Hence, $\sum_{i=k}^{n} \alpha_i^2 = 1$ (a generalization of Pythagoras' Law). Now, we expand the quadratic form $x^T Ax$

$$x^T Ax = \left(\sum_{i=k}^{n} \alpha_i u_i\right)^T A \left(\sum_{i=k}^{n} \alpha_i u_i\right) = \left(\sum_{i=k}^{n} \alpha_i u_i\right)^T \left(\sum_{i=k}^{n} \alpha_i Au_i\right) = \tag{9}$$

$$= \left(\sum_{i=k}^{n} \alpha_i u_i\right)^T \left(\sum_{i=k}^{n} \alpha_i \lambda_i Bu_i\right) = \sum_{i=k}^{n}\sum_{j=k}^{n} \alpha_i u_i \alpha_j \lambda_i Bu_j =$$

$$= \sum_{i=k}^{n}\sum_{j=k}^{n} \alpha_i \alpha_j \lambda_i u_i Bu_j = \sum_{i=k}^{n} \alpha_i^2 \lambda_i \geqslant \sum_{i=k}^{n} \alpha_i^2 \lambda_k = \lambda_k.$$

Thus, for any x that satisfies the constraints, we have $x^T Ax \geqslant \lambda_k$. Since $u_k^T Au_k = \lambda_k$, we can deduce that the minimizer is $x = u_k$. □