



Delft University of Technology

Spectral Modelling for Spatial Network Analysis

Nourian, Pirouz; Rezvani, Samaneh; Sariyildiz, Sevil; van der Hoeven, Franklin

Publication date

2016

Document Version

Final published version

Published in

Proceedings of the Symposium on Simulation for Architecture and Urban Design (simAUD 2016)

Citation (APA)

Nourian, P., Rezvani, S., Sariyildiz, S., & van der Hoeven, F. (2016). Spectral Modelling for Spatial Network Analysis. In R. Attar, A. Chronis, S. Hanna, & M. Turrin (Eds.), *Proceedings of the Symposium on Simulation for Architecture and Urban Design (simAUD 2016)* (pp. 103-110). simAUD.

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Spectral Modelling for Spatial Network Analysis

Pirouz Nourian¹, Samaneh Rezvani², Sevil Sariyildiz³, Frank van der Hoeven³

¹TU Delft, A+BE
Delft, the Netherlands
P.Nourian@tudelft.nl

²123dv architectuur & consult
Rotterdam, the Netherlands
S.Rezvani@123dv.nl

^{3,4} TU Delft, ³Design Informatics, ⁴Urban Design
Delft, the Netherlands
{I.S.Sariyildiz, F.D.vanderHoeven}@tudelft.nl

ABSTRACT

Spatial Networks represent the connectivity structure between units of space as a weighted graph whose links are weighted as to the strength of connections. In case of urban spatial networks, the units of space correspond closely to streets and in architectural spatial networks the units correspond to rooms, convex spaces or star-convex spaces. Once represented as a graph, a spatial network can be analysed using graph theory and spectral graph theory. We present four steps of modelling a spectrum for an urban spatial network; present an implementation of a state-of-the-art spectral graph-drawing algorithm and showcase a Spatial Eigenvector Centrality index, which is based on a novel definition of spatial networks based on Fuzzy Closeness indicators computed using Easiest Path distances.

Author Keywords

Spatial Network Analysis; Spectral Graph Theory; Spatial Eigenvector Centrality; Spectral Graph Drawing; Dominant Eigenvectors; Generalized Power Iteration

ACM Classification Keywords

I.6.1 SIMULATION AND MODELING (Model Development); G2 Discrete Mathematics as in G2.2 Graph Theory (Graph Algorithms, Network Problems)

See: <http://www.acm.org/about/class/1998/> for more information and the full list of ACM classifiers and descriptors.

1 INTRODUCTION

In the same way Social Networks affect people's social status, actions and choices, Spatial Networks have been found to affect people's spatial actions, such as mobility and accessibility. Social Network Analysis, originated in the areas of Sociometry and Psychometry, is a relatively mature field of study, with seminal publications, which can be traced back to 1950's and 1960's (e.g. [1], [2]). Graph Theory has been applied to study the structure of social network, remarkably in absence of concretely manifested social networks such as those popular nowadays like Facebook or LinkedIn. The field of Spatial Network Analysis in comparison is somewhat younger and arguably less well-structured, mainly due to the discrepancies of three lineages of work in modelling spatial networks, namely Geography, Transport Planning, and Spatial Analysis. In geographical and transport-related analyses, using a Junction-to-Junction graph model is pervasive, mainly due to the ease of metric distance measurements;

however, in the field of spatial analysis (as in Space Syntax [3] and similar approaches such as [4], [5] and [6] there is another approach to model spatial networks as adjacency representations of type Street-to-Street. This latter approach is fitter to human perception of space in that it corresponds to our intuition: its *nodes* are meaningful spaces such as streets or rooms; and its *links* can represent the difficulty of navigation from one space to another in wayfinding. Once a graph is constructed as an abstract representation of the spatial connectivity structure, it can be analysed using Graph Theory and Spectral Graph Theory.

Spectral Graph Theory (see e.g. [7]) studies the structural properties of graphs and networks (weighted directed graphs) by inspecting the eigenvectors and eigenvalues of some typical matrices associated with graphs, namely, Adjacency Matrix, Laplacian Matrix, Markovian Matrix (a.k.a. Transition Probability Matrix or Random Walk Matrix). We first show an intuitive application of Spectral Graph Theory in drawing large spatial graphs, by embedding their nodes in a low-dimensional Euclidean Space using eigenvectors of Laplacian and Random Walk matrices, implementing and extending the Power Iteration Method after [8]. Then, following our interpretation of a spatial network as an n -dimensional Hilbert space of random vectors $x \in \mathbb{R}^n$ where n is the number of nodes (spaces) in the network; we extend the notion of a Spatial Network to a graph that describes how close (similar) are the nodes (spaces) to one another.

We process the set of navigable spaces in these steps:

- Geographical Modelling
- Geometrical Modelling
- Topological Modelling
- Graphical Modelling
- Spectral Modelling

In this paper, we address the last two steps and give brief references of our earlier works in the previous steps. Two avoid common confusions between graphs and their drawings, we refer to graphs as comprised of nodes and links; and we reserve the terms vertices and edges for referring to topological constructs such as graph drawings. In the domain of geometry, we shall speak of points and lines analogously. The steps mentioned above transform a concrete set of geographical places gradually to a very abstract set of eigenvectors and eigenvalues.

This abstraction is done by: 1) reducing a real geographical place to a set of geometrical shapes (polylines and lines); 2) representing the topological relations among these shapes as in a Topological Skeleton (e.g. Straight Skeleton [9]), while abstracting topological Incidence Matrices and extracting Primal (Vertex-to-Vertex) and Dual (Edge-to-Edge) Adjacency Matrices as Graphs; 4) computing distances using graph-theoretical geodesics; and 5) studying the spectra (i.e. their Eigen pairs) of graph matrices such as Laplacian Matrix, and Random Walk Matrices (Transition Probability Matrices of Markov Chains) using iterative methods. Novelty reported in this paper are namely the method for extracting the first few important Eigen pairs of large graphs matrices, and introduction of a new graph matrix called Fuzzy Closeness Matrix.

2 GEOGRAPHICAL MODELLING

Official geographical models of transportation networks are usually made by authorities every few years. There are also crowd-sourced models such as OpenStreetMap, sometimes with a more detailed coverage. The key point is that a transportation network model corresponds to a mode of transportation. This sounds rather obvious but in case of street networks, we should note that there are paths that are navigable for pedestrians, which are not necessarily mapped on official street network maps. Missing a few important links can create big disconnected components in the resultant graph. Therefore, it is essential to capture all layers of maps relevant to a mode of transportation. Here we focus on walking and suggest that all relevant layers from OSM be merged together to ensure capturing the whole set of navigable spaces for pedestrians.

3 GEOMETRICAL MODELLING

To ensure a complete capture of the structure of the navigable space, we suggest that road surfaces be modelled, merged, and overlaid; and then the topological skeleton of the road surfaces be extracted. A systematic process as such is introduced in [10].



Figure 1. Topological Skeleton of a road surface map [10].

Similar information is typically available on OSM as polylines describing road segments, steps, bridges, and alike. However, such data models often require tedious steps of filtering to ensure a high quality topological model.

4 TOPOLOGICAL MODELLING

Once we have a set of reasonably clean set of geometric lines describing a topological skeleton of a set of roads

(navigable surfaces) using an error tolerance ε we can find out how the lines are (nearly) incident to one another. To capture these relations, we first model a set of topological vertex points at the intersection of the lines; these points might be geometrically many, but once seen as vertices they should form a set, i.e. they should be distinct as to the error tolerance mentioned before. This is to say the vertices V should be the topological representatives of their ε neighborhoods. Then a set of topological edges E can be constructed to represent how the lines in the map are incident to the mentioned vertices (again using the topological definition of a neighborhood). This is to say we construct a topological graph data structure.

$$G(V, E): v \in V \text{ \& } e \in E \quad (1)$$

Connectivity information of this graph can be captured in *Incidence Matrices*, whose rows correspond to vertex indices, and whose columns correspond to edge indices. We denote this matrix as A_{VE} and its transposed version as A_{EV} , i.e. $A_{VE} = A_{EV}^T$.

$$A_{VE} = [a_{i,j}^{ve}]_{|V| \times |E|} = \begin{cases} 1 & \text{if } V_i \sim E_j \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$A_{EV} = [a_{i,j}^{ev}]_{|E| \times |V|} = \begin{cases} 1 & \text{if } E_i \sim V_j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

5 GRAPHICAL MODELLING

Following [4], we form two types of graphs from the topological incidence matrices and represent them by their *Adjacency Matrices*. We can think of two type of adjacency matrices now, Vertex-to-Vertex and Edge-to-Edge, which we denote respectively as A_{VV} and A_{EE} .

$$A_{VV} = [a_{i,j}^{vv}]_{|V| \times |V|} = \begin{cases} 1, & \text{if } i \neq j \text{ \& } V_i \sim V_j \\ Deg(v), & \text{if } i = j \end{cases} \quad (4)$$

$$A_{EE} = [a_{i,j}^{ee}]_{|E| \times |E|} = \begin{cases} 1, & \text{if } i \neq j \text{ \& } E_i \sim E_j \\ Deg(e), & \text{if } i = j \end{cases} \quad (5)$$

It can be shown that:

$$A_{VV} = A_{VE}A_{EV} \quad (6)$$

$$A_{EE} = A_{EV}A_{VE} \quad (7)$$

In the above equations $Deg(v)$ denotes the number of vertices intermediately (through a single intermediary edge) adjacent to a vertex v and $Deg(e)$ denotes the number of edges intermediately (through a single intermediary vertex) adjacent to an edge e . We denote A_p as the adjacency matrix corresponding to the primal graph $\Gamma_p(N, L)$ and A_d as the adjacency matrix corresponding to the dual graph $\Gamma_d(N, L)$; in addition, we consider diagonal matrices D_p and D_d , whose diagonal entries are respectively equal to degrees of vertices and edges in G .

$$A_{VV} = A_p + D_p | D_p := \text{row sums of } A_{VE} \quad (8)$$

$$A_{EE} = A_d + D_d | D_d := \text{row sums of } A_{EV} \quad (9)$$

It is of course quite straightforward to obtain A_p and A_d from A_{VV} and A_{EE} computationally. So far, we showed that based on the same topological model two graphical models could be constructed. We choose our dual graph A_d as the basis for representation of the spatial network for walking (and cycling). However, this graph is yet not a network (i.e. it only captures topological information) because it is not weighted yet. If we assign costs/impedances to each link in this graph, then we can find optimal paths in this graph. It is interesting to note that this graph, as long as not weighted is undirected; however once weighted as to the ease of walking (or cycling) from one street space to another, it can become undirected, i.e. having an asymmetric adjacency matrix.

We hereby give a very short overview of our methods on how we attribute costs to the links of the abovementioned graph. We consider that an optimal path for walking or cycling should minimize the cost of traversal from one node to another, i.e. from one space to an adjacent space. This cost might be physical, e.g. as to the slope of the route or cognitive, e.g. as to the difficulty of navigating towards the next node. Inspired by [11], [12], [13] and [14]; we have made a dual graph representation that finds the **Easiest Paths** for walking or cycling [15]. These paths are as short, flat, and straightforward as possible. These paths are found by searching the network space whose links are inversely biased by attributing costs of traversal in terms of **minutes of travel time**. We show an exemplary graph as such on a hypothetical space network shown in Figure 2. Spaces are represented as lines (at right), modelled as a dual graph and then weighed asymmetrically by impedances of going from each street to another considering elevation angle (the slope lowering walking speed), azimuth angle (the steering angle complicating navigation) and the path lengths from/to the black dots as midpoints of the spatial nodes. The matrix plot in black and white shows the adjacency matrix that is symmetric, suggesting the underlying graph is undirected. However, once the impedances are assigned, the network graph model becomes directed; because its adjacency matrix is asymmetric, i.e. it is a directed [dual] graph models.

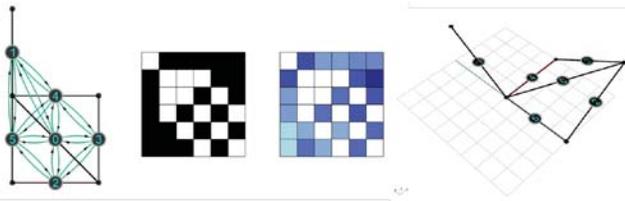


Figure 2. a hypothetical street space network, impedances of the links are asymmetric due to the differences between downhill and uphill traversals.

In order to model the cognitive difficulty of navigation ζ^A we formulate cognitive impedance as a function of the

azimuth angle between to streets that ranges between a maximum confusion time τ and 0.

$$\zeta^A := \begin{cases} h(\theta), & \text{if } Deg(n_{junction}) > 2 \\ 0 & \text{otherwise} \end{cases} \quad | \quad h(\theta) = \tau \sin^2 \frac{\theta}{2} \quad (10)$$

The total impedance of traversing a link $\zeta_{i,j}$ is then formulated as below:

$$\zeta_{i,j} = \begin{cases} \zeta_{(i,j)}^W + \zeta_{(i,j)}^A, & \text{if walking} \\ \zeta_{(i,j)}^C + \zeta_{(i,j)}^A, & \text{if cycling} \end{cases} \quad (11)$$

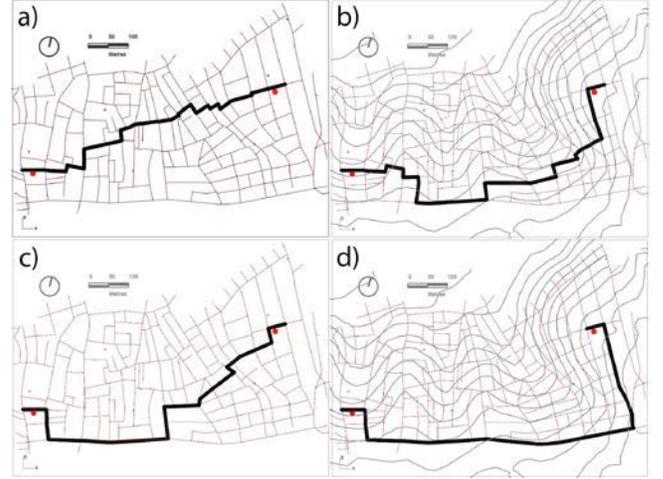


Figure 3. a) A Shortest Path without considering the terrain and difficulty of navigation on an example network from "Tarlabasi", Istanbul, data set provided by Ahu Sokmenoglu; b) Easiest Path geodesic found considering the terrain and $\tau=0$ for angular confusion (thereby no cognitive impedance; c) Easiest Path geodesic computed not considering the terrain and $\tau=15$ seconds; d) Easiest Path geodesic computed considering the terrain and $\tau=15$ seconds. The parameter τ (tau), introduced in equation (10) determines the maximum time wasted for making a navigation choice at a junction.

Walking time and cycling time when traversing i^{th} street to j^{th} street are denoted as $\zeta_{(i,j)}^W$ and $\zeta_{(i,j)}^C$ respectively. It is notable that these values are parametric and can be adjusted to represent motor assisted bikes. The easiest path is then the path π that minimizes the following sum over all possible paths.

$$\min_{(i,j) \in L} \sum_{(i,j) \in A} \zeta_{(i,j)}, \quad A = \{(i,j) | (i,j) \in L \cap \pi\} \quad (12)$$

5.1 Revisiting Network Distance

Any notion of distance is based on a corresponding geodesic or optimal path of some minimum cost or distance; this is because otherwise the notion of distance will be subject to different interpretations. While many studies take it for granted that shortest path is the basis of network distance, we argue that network distance should be defined for each mode of transportation; hence, we redefine network distances for pedestrians or cyclists as travel times experienced through Easiest Paths.

5.2 Geodesic Centrality Models

Using the Easiest Path (EP) algorithm, we can compute* a class of ‘directed geodesic centrality’ indices, namely (generalized) Betweenness Centrality [16] and Closeness centrality [2]. We call them geodesic centrality measures because they are directly computed using geodesic paths or geodesic distances. We compute EP Betweenness as an indicator of how many times a certain street space happens to be part of a geodesic (Easiest Path), provided that the destination of the path is not further than a threshold radius of search (inspired by Local Integration Centrality of Space Syntax). EP Closeness Centrality is then computed as the following: for each node we compute the average distances of all nodes reachable within a search radius and then Fuzzify (as in Fuzzy Logics [17]) that distance to assign a closeness index.

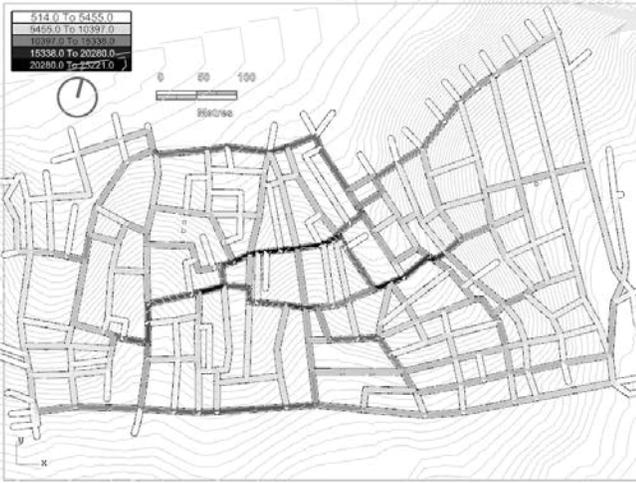


Figure 4. Easiest Paths Betweenness Centrality, $\tau = 15'$, Directed Graph, Search Radius is 10 Minutes Walking



Figure 5. Easiest Paths Closeness Centrality, $\tau = 15'$, Directed Graph, Search Radius is 5 Minutes Walking, revealing a polycentric structure in the neighbourhood

6 SPECTRAL MODELLING AND SIMULATION

Spectral analyses begin with inspecting a few eigenvectors of a graph matrix. One of the most intuitive evidences of the usefulness of spectral methods in studying networks is its application in drawing the undirected graph associated with the network. The point is that although eigenvectors seems to be very abstract, they turn out to be capable of reconstructing a concrete topological embedding that is often a ‘good’ graph drawing. Other applications of spectral methods can be found in forming a measure of centrality called Eigenvector Centrality that also has an intuitive interpretation in spite of its sophisticated name. Eigenvector Centrality (based on [1], [18], and [19]) assumes that the centrality of a node is determined by the centrality of the nodes that are immediately linked to it. A variant of this centrality index is used in the Google PageRank algorithm for ranking webpages as to their importance [20]. We hereby show the application of dominant eigenvectors of some matrices associated with graphs in spatial analysis. In doing so, we focus on some subtleties and issues in using eigenvectors in analysing (potentially large) spatial networks. To have a smooth transition to the topic we begin by the intuitive topic of Spectral Graph Drawing.

6.1 Spectral Graph Drawing

A graph is an abstract construct that captures the relations between a set of elements (nodes) as in their pair-wise relations. We usually have a tangible idea of a graph, as a spatial network because of the history of graph theory that was remarkably started by Leonhard Euler in studying a spatial network (the famous 7 bridges of Königsberg, the current city of Kaliningrad). However, note that this form of a spatial network representation is only one way to capture connectivity of spaces, which is in our terminology a Junction-to-Junction adjacency representation. The point is that a graph per se needs not to have a geometric or topological representation to exist. Once abstracted, it will be simply a matrix of adjacencies without any direct reference to geometric space. If we later decide to draw a graph, we can do it in a number of different ways, such as assigning a set of geometric points to the set of nodes and drawing geometric lines so as to represent links between the nodes. This will always be by definition an arbitrary choice, for a single graph can have infinitely many correct drawings as such. This way of representing a graph is called topological embedding. In the context of graph drawing one usually speaks of goodness of a drawing in terms of such things as good distinction between vertices (representing nodes), i.e. to avoid crossings between edges (representing links). While there are many methods for making ‘good’ graph drawings, there is one method that is scientifically very interesting as it has a unique topological solution using only a matrix associated with the graph. Historically, the first matrix used for this purpose was the Laplacian Matrix $L = D - A$ [21], in which D is a diagonal matrix whose diagonal entries equal node degrees (row sums of the adjacency matrix in case of undirected graphs).

The idea is to use ‘the first few eigenvectors’ of the Laplacian Matrix to place some vertices for embedding the graph in a low-dimensional Euclidean space, usually 2D or 3D. The first subtle issue to be noted is that this notion of first few eigenvectors might be confusing in the sense that it depends on how one defines the sorting of eigenvalues. Some textbooks use a convention of sorting the eigenvalues in an ascending order and some others in a descending order. The important issue is that intuitively, the eigenvectors associated with the largest eigenvalues are most explanatory for the variances between nodes in a graph; this is based on a view of a graph as a construct that captures similarity between nodes [8]. In that sense, some works also use the first few eigenvectors of the Adjacency Matrix for Spectral Embedding, specifically those associated with the largest eigenvalues. Note that using the first eigenvectors associated with the smallest eigenvalues of the Laplacian corresponds to the minimizers of the potential energy stored in imaginary springs representing the edges of the embedding. The following is not a proof, but a mathematical explanation of how the eigenvectors of the Laplacian matrix can be used to obtain a good drawing.

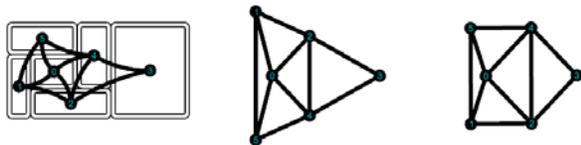


Figure 6. spectral drawing of a graph representing spatial connectivity of a hypothetical configuration of rooms (left), using Laplacian Matrix (middle) and using the Lazy Random Walk Matrix (right)

We know that the potential energy stored in a spring is proportionate to its squared length. Assuming a position vector $\mathbf{x}(i) \in \mathbb{R}^k$ for the i^{th} node of a graph (i.e. a vertex in \mathbb{R}^k), while k is typically 2 or 3, then we are interested in minimizing the sum of squared spring lengths:

$$\min_{(i,j) \in L} \sum_{(i,j) \in L} (\mathbf{x}(i) - \mathbf{x}(j))^2 \quad (13)$$

Which can be shown to be the same as the ‘quadratic form’ associated with the Laplacian Matrix L , meaning:

$$\mathbf{x}^T L \mathbf{x} = \sum_{(i,j) \in L} (\mathbf{x}(i) - \mathbf{x}(j))^2 \quad (14)$$

This minimization problem has a degenerate trivial solution of $\mathbf{x} = \mathbf{0}$, which is not interesting at all. To avoid this, we can impose a constraint, which keeps the variance of the vertices equal to a constant. The variance of the vertices, assuming the average vertex to be centered at $\bar{\mathbf{x}} = \mathbf{0}$ can be written as below:

$$\mathbf{x}^T \mathbf{x} = \sum_{i=1}^n (\mathbf{x}(i))^2 \quad (15)$$

These two equations together mean that we can minimize the following:

$$\min_x \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \quad (16)$$

This quotient is widely known as the **Rayleigh Quotient**, whose minimizers are indeed eigenvectors associated with its minimum values that are the lowest eigenvalue[†], while of course eigenvectors are linearly independent, i.e. $\langle v_i, v_j \rangle = 0: \forall i, j \in [1, n]$:

$$\lambda_i = \min\{R_L(\mathbf{x}) | \mathbf{x} \neq 0\} = \min_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \quad (17)$$

$$v_i = \{\mathbf{x} | R_L(\mathbf{x}) = \lambda_i\} = \arg \min_{\mathbf{x} \neq 0} \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \quad (18)$$

6.2 Finding the Dominant Eigenvectors

The Laplacian matrix is related with the negated Adjacency Matrix, so its eigenvectors are reversely ordered; this is why in Spectral Graph Drawing, some scholars use the top eigenvectors of the adjacency matrix, although not with very nice results [8]. What is interesting about top eigenvectors (i.e. those associated with the largest eigenvalues) is that they can be found quickly for large graphs using iterative methods such as Power Iteration Algorithm. This is the same algorithm used for finding the eigenvector representing Google PageRank. Note that the alternative to iterative methods is finding the Eigenvalue Decomposition (EVD) that is a prohibitively complex computational process for large matrices. While the solution to finding the top one eigenvector is widely known and applied, as in finding Google Page Rank, for finding the first few dominant eigenvectors there is no direct hint to a straightforward intuitive method in the literature. We here give a simple algorithm extracted and generalized from [8] for this purpose:

Inputs: Matrix $[n \times n]$ M , int k , int MaxIter //for Hermitian Matrices
Outputs: Vector[] EVecs, double[] EVals //arrays of results
<pre> vector[n][k] u=new vector[n][k]; //EVecs double[k] lambda=new double[k]; //EVals for (int i = 0; i < k; i++){ double CoDir=0; int counter=0; vector u_i=Vector.Random(n); u_i.Normalize(2); //p-norm do{ u[i] = u_i; for (int j = 0; j < i; j++){ u[i] = u[i] - ((u[i],u[j]) / (u[j],u[j])) * u[j]; //orthogonalise u[i].Normalize(2); } u_i = M * u[i]; u_i.Normalize(2); CoDir = (u_i, u[i]); counter++; } while ((CoDir < 1 - ε) ∩ (counter < MaxIter)) u[i] = u_i; } for (int i = 0; i < k; i++){ lambda[i] = (u[i], M * u[i]); } EVecs = u; EVals = lambda; </pre>

Algorithm 1. Find k Top Eigenvectors via Power Iteration after [8]

Using the first ‘top degree-eigenvectors’ (generalized by Yehuda Koren [8]) of the matrix $M = \frac{1}{2}(I + D^{-1}A)$, a.k.a. the Lazy Random Walk Matrix we can obtain a Spectral Drawing, which is more interesting than the one done by Laplacian for it is more intuitive and scalable because it uses top eigenvectors and so those eigenvectors can be computed by a generalized power iteration method.

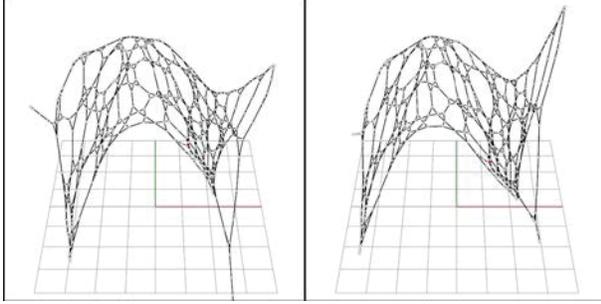


Figure 7. Left: 3D Spectral Graph Drawing of the example spatial network using the eigenvectors of the Laplacian Matrix; **Right:** 3D Spectral Graph Drawing of the example spatial network using the eigenvectors of the Lazy Random Walk Matrix, after Kroen [8]

The reason this algorithm converges can be understood by thinking of a random vector as being defined in terms of a linear combination of eigenvectors. This can be done because eigenvectors are mutually perpendicular to one another and thus for an ‘orthogonal basis’. That is:

$$\mathbf{x} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \dots + \alpha_n \mathbf{v}_n \quad (19)$$

If we multiply both sides by a matrix A , we get:

$$A\mathbf{x} = \alpha_1 A\mathbf{v}_1 + \alpha_2 A\mathbf{v}_2 + \dots + \alpha_n A\mathbf{v}_n \quad (20)$$

By virtue of the fact that vectors \mathbf{v}_i are eigenvectors with corresponding eigenvalues as λ_i we can replace all $A\mathbf{v}_i$ terms by $\lambda_i \mathbf{v}_i$ terms:

$$A\mathbf{x} = \alpha_1 \lambda_1 \mathbf{v}_1 + \alpha_2 \lambda_2 A\mathbf{v}_2 + \dots + \alpha_n \lambda_n A\mathbf{v}_n \quad (21)$$

Therefore:

$$A^2 \mathbf{x} = \alpha_1 \lambda_1^2 \mathbf{v}_1 + \alpha_2 \lambda_2^2 \mathbf{v}_2 + \dots + \alpha_n \lambda_n^2 \mathbf{v}_n \quad (22)$$

Therefore:

$$A^k \mathbf{x} = \alpha_1 \lambda_1^k \mathbf{v}_1 + \alpha_2 \lambda_2^k \mathbf{v}_2 + \dots + \alpha_n \lambda_n^k \mathbf{v}_n \quad (23)$$

This means by multiplying a random vector many times by the matrix in question, the product gradually converges to (is determined by) the direction of the dominant eigenvector; because other terms are attenuated by their lesser eigenvalues.

6.3 Fuzzy Closeness and Closeness Graph

We hereby explain the basis of our Fuzzy approach to forming such a Closeness Matrix. We define closeness as a Fuzzy linguistic variable that can be interpreted in view of a factor saying what is absolutely far for the perceiver of closeness. If one is not willing to walk more than 5 minutes, then every destination below 5-minute walk will be

somehow close to them but destinations farther than a 5 minutes’ walk will be absolutely far. Representing the truth level in the statement referring to closeness of a destination, we can formulate it as value between 0 and 1. In Crisp Logic, statements are either true (1) or false (0), but in Fuzzy Logics [17], we speak of the whole range [0,1] as for correctness of statements. We define Fuzzy Closeness as follows, where x represents temporal (travel-time) distance; μ represents an adjustment coefficient; and F denotes the temporal How-Far threshold:

$$C(x) = \frac{1}{1 + e^{\mu(x - \frac{F}{2})}} \quad (24)$$

We intend to obtain a sigmoid function to show the concept of closeness as it is perceived for a person.

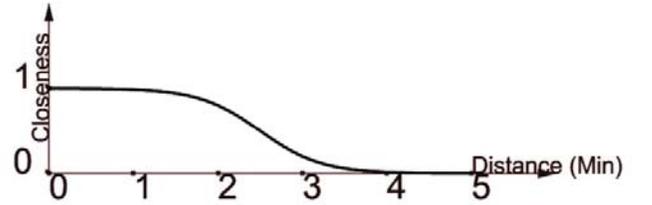


Figure 8. Fuzzy Closeness as a function of temporal distance

To ensure that the Fuzzy Closeness $C(x)$ will have a value smaller than ϵ at the threshold distance F we can set μ to the following:

$$\mu \geq \frac{2}{F} \ln\left(\frac{1}{\epsilon} - 1\right) \quad (25)$$

It is then straightforward to translate each temporal distance value in a distance matrix (whose entries are Easiest Path distances) to a Fuzzy Closeness value and form a Closeness Matrix. This matrix will be the representative of a graph that can be seen as a literal translation of the famous expression, a.k.a. the First Law of Geography, by Waldo Tobler: ‘‘Everything is related to everything else, but near things are more related than distant things’’ [22].

6.4 Spatial Eigenvector Centrality

On our undirected adjacency matrix, we can find eigenvector centrality rankings. Eigenvector centrality is a natural generalization of the intuitive notion of degree centrality, which is usually the first thing that comes to mind when speaking of centrality. If we say a more important node (say a person in a social network or a street space) is the one with more links (a person with many connections or a street where many other streets meet) then we are speaking of degree centrality. However, if we differentiate between connections (neighbours), we can redefine the centrality (importance [18], status [1], or accessibility [19]) relative to the centrality of the neighbours themselves in a recursive manner. This is of course an intuitive notion like ‘‘people are known by their friends’’ or ‘‘an important person is a person who is connected to important people’’. Formally:

$$\mathbf{c}^e(i) \propto \sum_{j \sim i} \mathbf{c}^e(j) \quad (26)$$

$$\mathbf{c}^e(i) = \eta \sum_{j \sim i} \mathbf{c}^e(j) = \eta \sum_{j=1}^n \mathbf{a}_{i,j} \mathbf{c}^e(j) \quad (27)$$

\mathbf{c}^e is a vector holding eigenvector centrality values, we can rewrite the same equation in matrix form:

$$\mathbf{c}^e = \eta \mathbf{A} \mathbf{c}^e \quad (28)$$

This would be more interesting if we reformulate as below where $\lambda=1/\eta$:

$$\mathbf{A} \mathbf{c}^e = \lambda \mathbf{c}^e \quad (29)$$

This centrality can be computed in two different ways on a spatial network:

- Firstly, by literally using the adjacency matrix of the network; and
- Secondly, by computing it on a ‘Fuzzy Closeness’ Matrix

We propose the second approach viewing a graph as a construct that captures similarity (proximity in the tempo-spatial sense) between nodes.



Figure 9. Eigenvector Centrality of the adjacency matrix

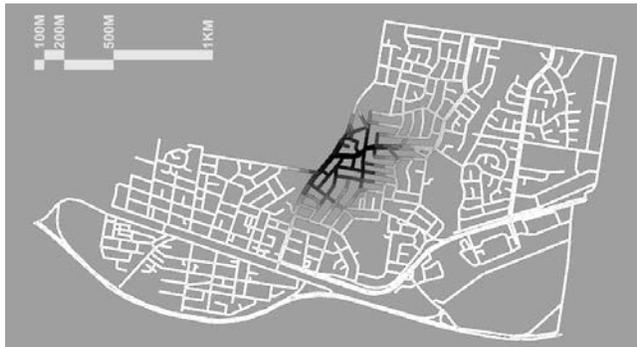


Figure 10. Eigenvector Centrality index of the Closeness matrix, when distances above “10 minutes cycling” are considered ‘far’

We can clearly see that the fuzzy closeness graph is more useful when using spectral methods. Now we show that each farness threshold can generate a different eigenvector centrality as such.



Figure 11. Eigenvector Centrality index of the Closeness matrix, when distances above “15 minutes cycling” are considered ‘far’



Figure 12. Eigenvector Centrality index of the Closeness matrix, when distances above “25 minutes cycling” are considered ‘far’

7 CONCLUSION

In this paper we introduced a number of novel models, methods and algorithms for Spectral Modelling in Spatial Network Analysis. We have generalized the concept of spatial network from the ‘topological connectivity’ (in Adjacency matrices) to the ‘perceived closeness’ (in Fuzzy Closeness matrices). This new definition of the Spatial Network comes closer to a Social Network, where places are all related to one another but near places are more related. Our fuzzy closeness approach, the Easiest Path algorithm, and its underlying graph representation are novel constructs that make spatial analysis more intuitive, understandable, and more easily interpretable; and at the same time connect it to the field of spectral graph theory. The main advantage of spectral analysis in modelling and simulation of large datasets is the fact that it reduces the dimensionality of the data to a few important factors and directions, using which we can transform our n-dimensional dataset to a low-dimensional Euclidean space, within which similarities based on distance would be representatives of similarities in the original space. The success of spectral graph drawing in producing ‘nice’ drawings using only abstract topological information is remarkable and illuminative. Similarly, the fact that we are able to reconstruct closeness-like distributions using eigenvectors on our generalized fuzzy graph definition proves a point about relevance of spectral methods in Spatial Network Analysis.

8 IMPLEMENTATION

The methods reported in this paper are all implemented in the new version of a freeware toolkit for Urban Configuration Analysis by the first authors. The toolkit will be available for download [here](#). We have used [MathNet](#) library for Linear Algebraic data structures and algorithms.

9 REFERENCES

1. Leo Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.
2. G. Sabidussi, "The centrality index of a graph," *Psychometrika*, 1966.
3. B. Hillier and J. Hanson, *The social logic of space*. 1984.
4. M. Batty, "A New Theory of Space Syntax," *CASA Work. Pap.*, vol. 44, no. 0, pp. 1–36, 2004.
5. B. Jiang and C. Claramunt, "Topological analysis of urban street networks," *Environ. Plan. B Plan. Des.*, vol. 31, no. 1, pp. 151–162, 2004.
6. S. Porta, P. Crucitti, and V. Latora, "The network analysis of urban streets: A dual approach," *Phys. A Stat. Mech. its Appl.*, vol. 369, no. 2, pp. 853–866, 2006.
7. F. R. K. Chung, "Spectral Graph Theory," *ACM SIGACT News*, vol. 30, p. 14, 1999.
8. Y. Koren, "On Spectral Graph Drawing," in *Computing and Combinatorics*, 2003, pp. 496–508.
9. P. Felkel and S. Obdrzalek, "Straight Skeleton Implementation," *Proc. Spring Conf. Comput. Graph.*, pp. 210–218, 1998.
10. R. Sileryte, "Analysis of Urban Space Networks for Recreational Purposes based on Mobile Sports Tracking Application Data," 2015.
11. M. Duckham and L. Kulik, "'Simplest' paths: Automated route selection for navigation," *Spat. Inf. Theory, Proc.*, vol. 2825, pp. 169–185, 2003.
12. A. Turner and N. Dalton, "A simplified route choice model using the shortest angular path assumption," *Geocomputation*, 2005.
13. W. Tobler, *Three presentations on geographical analysis and modeling*. 1993.
14. Rhett Allain, "What's the Steepest Gradient for a Road Bike?," *What's the Steepest Gradient for a Road Bike?*, 2013. [Online]. Available: <http://www.wired.com/2013/03/whats-the-steepest-gradient-for-a-road-bike/>.
15. S. Nourian, Pirouz; van der Hoeven, Franklin; Rezvani, Sama; Sariyildiz, "Easiest paths for walking and cycling: Combining syntactic and geographic analyses in studying walking and cycling mobility," in *Proceedings of the 10th Space Syntax Symposium, UCL, London*, 2015.
16. L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.
17. L. a. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
18. P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *J. Math. Sociol.*, 1972.
19. P. R. Gould, "On the Geographical Interpretation of Eigenvalues," *Trans. Inst. Br. Geogr.*, no. 42, pp. 53–86, 1967.
20. L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: bringing order to the web.," *Technical report, Stanford Digital Library Technologies Project*, 1998. [Online]. Available: <http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf>.
21. K. M. Hall, "An r-Dimensional Quadratic Placement Algorithm," *Manage. Sci.*, vol. 17, no. 3, pp. 219–229, 1970.
22. W. R. Tobler, "A computer movie simulating urban growth in the Detroit region," *Econ. Geogr.*, pp. 234–240, 1970.

* This is a newer version of our previous work [15], in which we have considered underlying graphs as directed graphs and obtained different results.

† Since we are dealing with undirected graphs in Spectral Drawing, these eigenvalues correspond to symmetric (Hermitian, Self-Adjoint) matrices and are therefore real-valued.