Wavelet-based Visualization of Time-Varying Data on Graphs

Paola Valdivia* University of São Paulo
Fabio Dias* University of São Paulo
Fabiano Petronetto† UFES
Cláudio T. Silva‡ New York University
L. G. Nonato* University of São Paulo

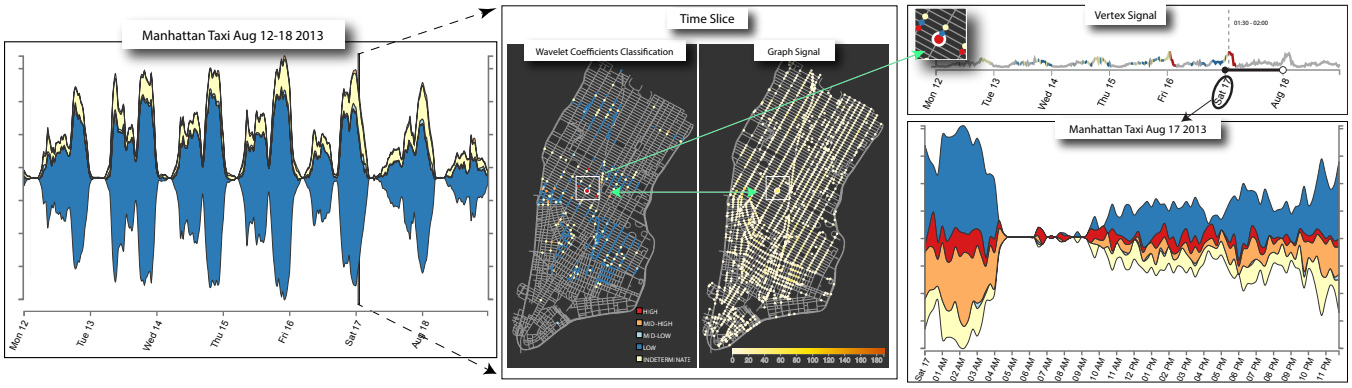


Figure 1: The proposed graph wavelets-based visualization allows to identify, in each time slice, regions from low to high variation (middle left) of a function on the nodes of a graph (middle-right), while still enabling to analyze how those variation evolve over time (left). High frequency variation indicates abrupt changes in the function, attracting users attention to relevant events, which can demand further investigation (right).

Abstract

Visualizing time-varying data defined on the nodes of a graph is a challenging problem that has been faced with different approaches. Although techniques based on aggregation, topology, and topic modeling have proven their usefulness, the visual analysis of smooth and/or abrupt data variations as well as the evolution of such variations over time are aspects not properly tackled by existing methods. In this work we propose a novel visualization methodology that relies on graph wavelet theory and stacked graph metaphor to enable the visual analysis of time-varying data defined on the nodes of a graph. The proposed method is able to identify regions where data presents abrupt and mild spacial and/or temporal variation while still been able to show how such changes evolve over time, making the identification of events an easier task. The usefulness of our approach is shown through a set of results using synthetic as well as a real data set involving taxi trips in downtown Manhattan. The methodology was able to reveal interesting phenomena and events such as the identification of specific locations with abrupt variation in the number of taxi pickups.

Keywords: Time-varying data, graph wavelets, stacked graph visualization

1 Introduction

With the proliferation of sensors and monitoring tools, ever increasing amounts of data are made available for analysis and exploration. Often, this data can be expressed as a signal over a graph structure. Such abstraction is very flexible, capable of expressing complex spatial relationships with time-varying information. However, identifying relevant events and phenomena from spatio-temporal data is not straightforward, rendering the visual analysis indispensable for uncovering hidden patterns and their evolution over time.

An example of such task is the exploration of taxi activities data, revealing general spatial and temporal patterns of behavior as well as regions of disrupture such as an abrupt increase or decrease in the number of pickups in certain areas, indicating the occurrence of some event in those areas. Both patterns of information can be useful to optimize the taxi service and monitoring the city as a whole.

Although many techniques have been proposed for analyzing time-varying data, there are important aspects not properly tackled by existing methods. For instance, identifying regions and time intervals of abrupt and/or smooth variation of a function while visually tracking the evolution of those patterns over time is not an easy task for most visualization techniques. A sudden temporal variation in a function can indicate the appearance of an unexpected event or a change in the behavior of a given phenomenon. Analyzing function variations over time allows for understanding the periodicity of such variations. Therefore, providing users with effective tools for visualizing functions variations is of great importance.

Wavelet theory is a powerful mechanism for analyzing local variations of functions, widely employed in pattern recognition, data compression, and signal filtering. Recently, it has been extended to graphs, *e.g.* [10, 17], opening a multitude of application possibilities, from transport network analysis [13] to community detection [19]. However, directly interpreting wavelets coefficients is not trivial, requiring expert knowledge. Combining wavelet theory with visual analytic tools is a practical alternative for non-specialized users, allowing the effective analysis of time-varying data.

In this work we propose a novel visual analytic methodology for the analysis of time-varying data that combines graph wavelets, pattern recognition/classification, and stacked graph visual metaphor in a linked view visualization that allows for the analysis of local and global data variation. By properly handling wavelet coefficients as feature descriptors for classification, the method removes the need for specific knowledge of the particularities of the transform and is able to reveal regions and time intervals with similar dynamic, regardless of their spatial and temporal distance. The provided results and case study show the effectiveness of the proposed

*e-mail: {paolalv,fabio.dias,gnonato}@icmc.usp.br
†e-mail: fabiano.carmo@ufes.br
‡e-mail: csilva@nyu.edu

methodology in highlighting regions of abrupt and mild data variation as well as how those variation evolve over time.

In summary, the main contributions of this paper are:

- A novel method for the visual analysis of time-varying data defined on the nodes of a graph which combines graph wavelets, pattern recognition/classification, and stacked graph metaphor.
- A methodology to classify graph nodes based on their wavelet coefficients that enables spatial and temporal visual analysis of data variation.
- Results using synthetic data and a real case study that showcase the capability and potential of the proposed visual analytic tool in revealing interesting phenomena and events from massive time-varying data, including Citi Bike rents (in the supplementary material) and taxi trips in downtown Manhattan.

2 Related Works

We focus this section on visualization methods for time-varying data defined on graphs. The literature about time-varying data visualization is extensive and comprehensive surveys can be found in the works by Aigner *et al.* [1], which approaches spatio-temporal methods for general data visualization, and Kehrer and Hauser [12], which discusses visual analysis methods for multifaceted data. While those surveys are not focused on visualization of functions/signals on graphs, they can provide insights about spatio-temporal methods. The visualization of dynamic graphs [3] is also a related problem, but with different characteristics, since not only the data associated with the graph evolves with time, the structure of the graph is allowed to change as well.

Time-varying visualization A popular application for time-varying information defined on graphs is in the study of urban data. Ferreira *et al.* [8] uses the start and end information of taxi trips to explore urban mobility, introducing an efficient storage manager to deal with large volume of data. The method is capable of handling different queries such as trips that begin on downtown Manhattan and end at the airports. However, the exploration of changes in the information is not fully developed, providing only a plot of total events and the juxtaposition of geographic renderings for different time slices. Doraiswamy *et al.* [7] consider the same kind of data as our case study, proposing a method to support event-guided exploration of large urban data. The problem is approached through topological tools and is able to detect events of different scales and shapes. Detected events can be used as a parameter for querying, however, the method does not provide details on the temporal evolution of the information not detected as an event.

Wang *et al.* [20] also consider taxi tracking information that is filtered and matched to road networks. This information is used to calculate trajectories and create propagation graphs of the traffic information. Velocity in each road can be viewed as a timeline-based layout or using a geographical map. The propagation graphs are used to show concise information about detected events, specifically how the traffic jams propagate. In contrast to our approach, their pixel-based layout does not provide an efficient way to observe patterns of change in the information. Zeng *et al.* [21] study the visualization of interchange patterns between defined locations, introducing the concept of *interchange circos diagrams*. The method uses juxtaposition of diagrams to explore temporal changes. Since the method is based on multi-scale information aggregation, users have to interactively explore the information throughout different scales, that can hinder the detection of small anomalies. Such remarks can also be made for the work proposed by Andrienko and Andrienko [2], where data aggregation methods are used to explore trajectory-oriented and traffic-oriented views of movement data, going beyond the usual spatio-temporal aggregation schemes by considering direction and route information. Since they overlay the pixel-based traffic information into the map, only the coarse geographical information is kept during visualization.

Pu *et al.* [15] present a system for visualizing mobility patterns based on phone call information. The visualization is based on a Voronoi diagram computed from cellphone tower location, allowing the analysis of people migration between different towers, depicted on the edges of the diagram. A similar approach was used by Sun *et al.* [18] to analyze temporal information using roads of a map as a graph. The time-varying information is associated with the edges of the graph rather than the nodes. The main advantage of Pu and Sun methods is the ability to correlate spatial and temporal information. Such methods are not suitable, though, for large graphs. Handling levels of detail is not also a straightforward task.

A quite different approach is proposed by Chu *et al.* [5], where geographic coordinates are transformed into street names, therefore the trajectory of a particular taxi can be expressed as a document and the data set itself as a document corpora, allowing to process the data through nature language processing tools such as *Latent Dirichlet Allocation*. Data variations are expressed through changes in the "taxi topics" and depicted using a set of alternative plots. However, the visualization is focused on the taxi topic information, thus raw information such as the concentration of taxis in specific position and time can not be directly visualized.

Guo *et al.* [9] introduce a visualization system called TripVista, aiming the exploration of microscopic traffic patterns and abnormal behaviors, such as traffic on a particular street junction. The method involves, along with other visualization techniques, a timeline visualization derived from Theme-River [11], but enriched with directional information. While this approach is capable of identifying underlying patterns, outliers, and abrupt changes in the data, it is not directly scalable to macroscopic patterns and can not discriminate changes according to the frequency of the signal.

Graph Wavelets Since our approach relies on graph wavelets, we also review some literature about techniques that relies on graph wavelets to analyze data defined on the nodes of a graph [10]. Crovella and Kolaczyk [6] proposed a methodology based on graph wavelets for monitoring information on a network. However, the problem is approached in a purely spatial manner, disregarding temporal variations. Mohan *et al.* [13] make use of wavelets on graphs to detect disruptive traffic events on transportation networks. Wavelet coefficients are used to detect speed patterns, such as congestions, low traffic, or rush hours, including the duration of such events. However, since the focus of the article is not visual analysis, the visualization provided is limited to simple plots of the magnitude of the coefficients, ignoring the structure of the graph. Indeed, while the authors claim the study comprises several expressways and roads, the provided results consider only a single road at a time, for sake of simplicity in visualization.

In contrast to methods above, our approach makes use of graph wavelets theory to explicitly visualize the evolution of data variations in multiple scales, enabling the visualization of low and high frequencies patterns over time. Our method relies on linked views composed of traditional graph plots and stacked graph metaphor, being able to handle a large amount of data. The provided linked views allow the simultaneous visualization of spatial and temporal data variation for both large and small events, a trait not present in most of the time-varying visualization techniques described above.

3 Spatio-Temporal Wavelets Analysis on Graphs

The proposed methodology for graph-based spatio-temporal data analysis comprises four main steps (see Figure 2): graph wavelets transform, feature vector construction, feature vector classification, and visual representation. Before describing the graph wavelets transform theory, the basis of our methodology, we introduce some basic concepts and notations that are important in this context.

Figure 2: Our methodology comprises four main steps: graph wavelets transform which provides the coefficients for feature vector construction. After classifying the feature associated to each node the visual analysis takes place.

3.1 Mathematical Preliminaries

An undirected graph $G = (V,E)$ consists of a set of n nodes V and a set of edges E connecting pairs of nodes in V. A graph G can be represented by an $n \times n$ symmetric matrix A (adjacency matrix) with entries $a_{ij} = 1$ if only if there is an edge in E connecting the nodes τ_i, τ_j, $i \neq j$, and $a_{ij} = 0$ otherwise. The Laplacian operator $\mathscr{L}$ in G can be defined as $\mathscr{L} = D - A$, where D is the diagonal matrix with entries $d_{ii} = \sum_{k=1}^{n} a_{ik}$. Since $\mathscr{L}$ is a real symmetric semi-positive definite matrix it admits an eigendecomposition with real non-negative eigenvalues λ_j and corresponding orthogonal eigenvectors u_j, $j = 1,\ldots,n$ (see [4] for details). Assuming that the nodes are numbered, we denote by $u_j(i)$ the value of u_j in the node τ_i of G. By analogy to the continuous Laplacian operator where eigenfunctions and eigenvalues are the Fourier modes and frequencies respectively, the u_j and λ_j, $j = 1,\ldots,n$ are considered as graph Fourier modes and frequencies. Therefore, small values of λ_j correspond to low frequency modes while large values indicate high frequency modes.

Any real valued function $f : V \to \mathbb{R}$ that assigns a scalar to each node of G can be interpreted as a vector in $\mathbb{R}^n$. The scalar value in the node τ_i is given by the ith entry $f(i)$ of the vector $f \in \mathbb{R}^n$. The graph Fourier transform of f at the frequency λ_j is defined as:

$$\widehat{f}(j) = u_j^\top f = \sum_{i=1}^{n} u_j(i) f(i) \tag{1}$$

where $u_j^\top$ is the transpose of u_j. In words, Equation (1) says that the jth graph Fourier coefficient of f is given by the dot product between f and u_j. We refer readers to [16] and [22] for a detailed description of graph Fourier transform and its properties.

3.2 Spectral Graph Wavelets

The central idea of graph wavelet transform is to reinterpret a function as a composition of different functions of known behavior. By analyzing their contributions, we can gain insights regarding the behavior of the original function. For instance, if the largest contribution is made by a slowly changing function, we can infer that the original function is mostly smooth. Conversely, a rapidly changing function would have a significant contribution of similarly abrupt functions. Moreover, it can identify abrupt changes in a milder function, potentially discerning different phenomena.

The graph wavelet transform decomposes a function f in terms of basis functions $\{\psi_{s,1},\ldots,\psi_{s,n}\}$ where each $\psi_{s,i}$ depends on a scale s and a location τ_i. An interesting aspect of wavelets is that projecting the function f onto spaces with different scales is equivalent to band-pass filtering the function f, that is, each scale corresponds to a specific band-pass filter.

Recalling that the eigenvalues λ_j correspond to frequencies in the graph Fourier domain, we can define band-pass filters by properly handling the eigenvalues λ_j according to scales s. Denoting by g a filter kernel defined on $\mathbb{R}^+$, the graph wavelet basis functions at a positive scale s and location τ_i (ith node) can be defined as [10]:

$$\psi_{s,i} = U D_g U^\top \delta_i \tag{2}$$

where U is the matrix whose columns are given by the eigenvectors u_j, δ_i is a vector with 1 in the ith entry and zero in all other entries, and $D_g = \mathtt{diag}(g(s\lambda_1), g(s\lambda_2), \ldots, g(s\lambda_n))$ is a diagonal matrix representing the band-pass filter. Wavelet coefficients at scale s and node τ_i are obtained through the dot product

$$\omega_f(s,i) = \psi_{s,i}^\top f \tag{3}$$

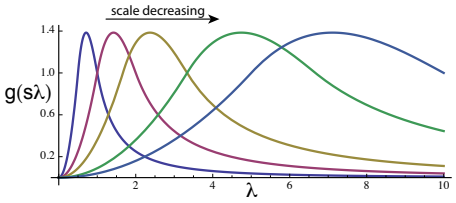


Figure 3: Band-pass filters at five scales.

The rationale behind Equation (2) is that for small values of s (small scales), the filter g stretches out so as to favor high frequency modes essential to good localization. Large values of s (large scales) compress the function around low frequency modes to encode coarser description of a local neighborhood. Formally, g is defined as follows:

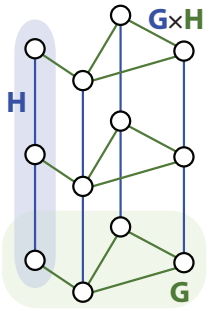
$$g(x) = \begin{cases} x_1^{-\alpha} x^\alpha & \text{for } x < x_1 \\ p(x) & \text{for } x_1 \leq x \leq x_2 \\ x_2^\beta x^{-\beta} & \text{for } x > x_2 \end{cases} \tag{4}$$

where α, β, x_1, x_2 are parameters of the filter that can be tuned to change the behavior of g. Function $p(\cdot)$ is a cubic polynomial satisfying $p(x_1) = p(x_2) = 1$, $p'(x_1) = \alpha/x_1$, and $p'(x_2) = -\beta/x_2$. The scales are logarithmically sampled between $s_{min} = s_1, s_2, \ldots, s_m = s_{max}$, where $s_{min} = x_2/\lambda_n$ (λ_n is the largest frequency in the spectrum) and $s_{max} = 20x_2/\lambda_n$. The provided formulation is based on the work by Hammond et al. [10], but we are presenting it using a more linear algebra notation for easier interpretation. Interested readers should refer to [10] for a thorough mathematical description of graph wavelets and their properties. In our implementation we set all the parameters as suggested in [10], $\alpha = \beta = 2, x_1 = 1, x_2 = 2$. Figure 3 illustrates the band-pass filter g in five distinct scales.

3.3 Time-Varying Data

The graph wavelet formulation presented in previous subsection was designed to process real functions with no temporal variation. A natural way to extend the theory to process time-varying data is through the so called Cartesian product graph.

Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two graphs and τ_i and ι_j be nodes in V_G and V_H, respectively. The Cartesian product between G and H is the graph $G \times H$ with node set $V_G \times V_H$ and edges connecting two nodes (τ_i, ι_j) and (τ_k, ι_l) if only if either $\tau_i = \tau_k$ and ι_j is adjacent to ι_l in H, or $\iota_j = \iota_l$ and τ_i is adjacent to τ_k in G. The particular case where H is a linear graph is of special interest in our context, as the Cartesian product $G \times H$ can be seen as copies of G stacked according to the nodes of H.



Let $G = (V_G, E_G)$ be a graph and $f : V_G \times [a,b] \to \mathbb{R}$ be a time-varying function that assigns, for every time t in the interval $[a,b]$, a real scalar to each node $\tau_i \in V$. Assuming a discretization $a = t_1 < t_2 < \cdots < t_r = b$ for the interval $[a,b]$, we can define the Cartesian product graph $G \times H$, where H is a linear graph with nodes $V_H = \{\iota_1, \iota_2, \ldots, \iota_r\}$ and edges $E_H = \{e_{ii+1} = \iota_i\iota_{i+1}\}$, $i = 1,\ldots,r-1$. Therefore, the time-varying function f can naturally be extended to $G \times H$ through the function $f_{G\times H} : V_G \times V_H \to \mathbb{R}$ such that $f_{G\times H}\left((\tau_i, \iota_j)\right) = f(\tau_i, t_j)$.

The main advantage of extending a time-varying function as described above is that $f_{G\times H}$ is a "steady" function on the nodes of $G \times H$, therefore, the graph wavelet theory presented in subsection 3.2 can be directly employed to analyze $f_{G\times H}$. However, the number of nodes in $G \times H$ increases as nr, thus, even for moderate values of n and r, the size of $G \times H$ and the corresponding Laplacian matrix can be considerable, hampering the computation of eigenvalues and eigenvectors. Nevertheless, the spectrum of Cartesian product graphs has the particular property of being derived from the spectrum of G and H, making unnecessary the construction of the Laplacian matrix associated to $G \times H$. Precisely, let u_j, λ_j be the eigenvectors and eigenvalues of G and v_k, μ_k be the eigenvectors and eigenvalues of H, then $\lambda_j + \mu_k$ is an eigenvalue of $G \times H$ and $w_{jk} = u_j \otimes v_k$ the corresponding eigenvector, where $\otimes$ is the Kronecker product [4]. Computing the spectrum of $G \times H$ from the eigenvalues and eigenvectors of G and H makes the use of graph wavelet theory feasible for handling large time-varying data.

3.4 Node Classification and Stacked Graphs

Let $f_{G\times H} : V_G \times V_H \to \mathbb{R}$ be the extension of a function to a Cartesian product graph $G \times H$. We denote by $\omega_{f_{G\times H}}(s,i,j)$ the wavelet coefficient of $f_{G\times H}$ at scale s, and location (τ_i, ι_j), as described in equation (3). If scales are discretized as $s_{min} = s_1, s_2, \ldots, s_m = s_{max}$, then each node (τ_i, ι_j) of $G \times H$ can be associated to an m-dimensional feature vector with attributes given by the wavelets coefficients $w_{ij} = (\omega_{f_{G\times H}}(s_m,i,j), \ldots, \omega_{f_{G\times H}}(s_1,i,j))$, where large scales are placed on the left and small scales on the right of w_{ij}.

For small scales, wavelet coefficients tend to be high in regions of the graph where f varies abruptly. On the other hand, for large scales, wavelet coefficients with high magnitude tend to appear in regions where f has a "smoother" behavior. Therefore, one can characterize each node of $G \times H$ according to the distribution of wavelet coefficients in its feature vector. More precisely, if wavelets coefficients of a node (τ_i, ι_j) assume larger values in small scales (on the right part of w_{ij}) then that node is located in a region of abrupt variation of $f_{G\times H}$, and we will call (τ_i, ι_j) a *high frequency node*. If the coefficients are more concentrated in large scales (on the left part of w_{ij}) then the node is located in a region of smoother variation and it will be called *low frequency node*. In our experiments we have noticed that four scales is enough to reveal interesting phenomena, being the number of scales employed henceforth.

Node classification Each node (τ_i, ι_j) can be classified according to the distribution of coefficients in its feature vector w_{ij}, allowing the identification of high and low frequency regions in the graph and the visualization of how those regions evolve over time.

Classification can be performed through pattern recognition mechanisms commonly employed by the machine learning community. In this work we choose the pattern recognition neural network (PRNN) classifier [14], due to its reduced computational times and satisfactory performance in terms of classification. PRNN is a supervised classification method, thus demanding a training set in order to fit the classification model. Assuming that the feature vectors are four dimensional, each w_{ij} is built from four scale levels, we generate a training set made up of five classes, as illustrated in Figure 4. Feature vectors in the low frequency class has $w_{ij}(1)$ as the dominant attribute (Figure 4, dark blue bars); $w_{ij}(2)$ is the dominant attribute in the mid-low frequency class (Figure 4, light blue bars); $w_{ij}(3)$ and $w_{ij}(4)$ are the dominant attributes in the mid-high (Figure 4, orange bars) and high frequency (Figure 4, red bars) classes, respectively. Feature vectors with no dominant attribute characterize the indeterminate class (Figure 4, yellow bars). We simulate feature vector patterns in each class to serve as training data for the classification method. The training set was generated with one thousand samples per class, thus comprising a data set with five thousand elements. Figure 4 illustrates ten samples from each class generated by our training data simulator. The simulator was designed to mimic coefficient patterns we observed in real and synthetic data sets. Figure 5 shows a specific time-frame of a synthetic time-varying data (left-top) and regions of low, mid-low, mid-high, high and indeterminate frequencies (right) classified with the PRNN after a training step with simulated training data.

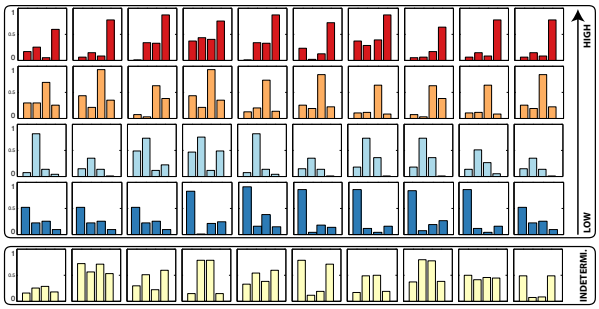


Figure 4: Four dimensional feature vectors: from top to bottom high, mid-high, mid-low, low and indeterminate frequency classes. Each row depicts ten examples of feature vectors in each training class and the bars correspond to the magnitude of each attribute.

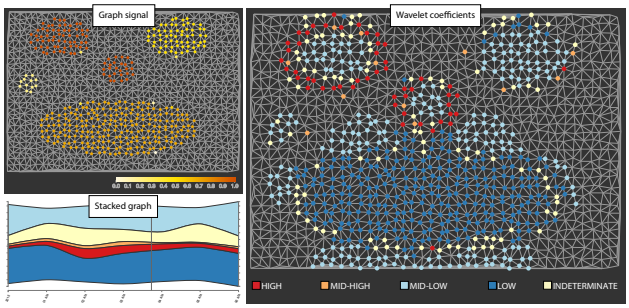


Figure 5: Top row: a function f in a time instant (left) and feature classification (right) regions of low, high and indeterminate. Bottom row: stacked graph illustrating the temporal evolution.

Stacked graph For each time t_i, we can count the number of nodes that belong to a specific class in that time. More specifically, let C be one of the five classes defined above and $(\tau_i, \iota_j)_C$ be a node in C. For each time-instant t_k, the set of nodes $C_{t_k} = \{(\tau_i, \iota_j)_C \,|\, j = k\}$, corresponds to the nodes in $V_G \times \iota_k$ that belongs to C. Therefore, we can apply a stacked graph metaphor to visualize the number of nodes in each class in the time t_i, $i = 1, \ldots, m$, which allows us to visualize how classes evolve over time. Figure 5 (left-bottom) illustrates the proposed visualization. Each horizontal strip represents a class, colors discriminate the class types, and the thickness of each strip accounts for the number of nodes in each class in each time t_i. The highlighted (vertical bar) instant in stacked graph corresponds to the time slice depicted in right figure. The stacked graph metaphor allows for easily identifying time intervals where a function f varies abruptly or presents low levels of variation, assisting in the analysis of f over time.

3.5 Computational Times

The bottleneck of our algorithm is the calculation of wavelets coefficients (Eq. 3). The coefficients are computed from eigenvectors and eigenvalues of the graphs G and H. We use LAPACK to compute the spectrum of both graphs, where the routine used has overall complexity $O(n^3)$. Since the spectrum of $G \times H$ is can be derived from the spectrum of graphs G and H using the Kronecker product and sum of eigenvalues, asymptotic complexity to compute all eigenvectors and eigenvalues of $G \times H$ is $O(n^2r^2 + n^3 + r^3)$, where n and r are the number of nodes in G and H respectively.

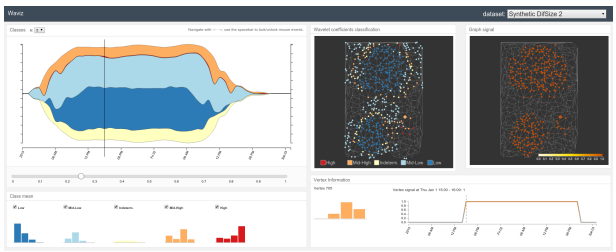


Figure 6: Screenshot of our prototype interface which comprises a stacked graph visualization (top left), threshold slide bar for wavelets coefficients (middle left), average feature vector in each class (bottom left), graph map depicting node labels and function intensity in specific time slice (top right), and time series of a node (bottom right).

4 Visual Elements and Linked Views

Figure 6 shows the interface of our prototype applied to a synthetic data set. The stacked graph-based visualization is depicted on the top left of the interface. From the stacked graph its possible to see that the amount of variation in the data increases initially, remains stable, then decreases. The data does not present time variation in the middle of the time domain, but there are nodes with non-zero coefficients due to spatial variation. By changing the slider under the stacked graph we can control the minimum value of the coefficients to be considered during visualization, hiding noises or events of smaller magnitude, that can be interpreted as less significant.

Nodes with the same classification are grouped and depicted with a specific color that encodes their frequency class. The average pattern of coefficients in each class is shown in the bottom left area of the interface. From the stacked graph, we can see that the variation on the data is predominantly of low-frequency, happening in more than 50% of the nodes. High and mid-high frequencies take place in about 20% of the nodes. From the small amount of vertices classified as high-frequency, we can infer that high-frequency events happen along the time, but they are spatially small. In this specific data set, detailed in Section 5, the time-varying information was designed to alternate between zero and one in specify regions and time intervals of the graph, but the region where the change happens is large and the underlying function remains constant for enough time to be expressed using lower frequencies, making high frequency coefficients to show up only in a small number of nodes.

The stacked graph visualization is linked to a time slice visualization depicting function intensities and nodes classification, as shown on top right in Figure 6. A time slice is selected from the stacked graph by placing the mouse on a specific time. Each node in that time interval with coefficients higher than a threshold (set in the slide bar) is colored according to its classification.

By clicking on a node one can access the corresponding feature vector and function value for that node. Moreover, the time series associated with the selected node is presented on the bottom right area of the interface, allowing the users to visualize how the data varies in that node over the whole period of time. The color of the line corresponds to the classification of the node at that time, or gray when the magnitude of the coefficient is below the threshold.

5 Visual Analytics with Graph Wavelets

In the following we show the usefulness of the proposed combination of graph wavelets, pattern classification, and stacked graph when analyzing the given synthetic data. We consider an event as any portion of the domain where one can notice a variation, even subtle, in the data under analysis. This definition is quite general and can be adapted to many different contexts. In this section, we consider a synthetic data set containing six distinct events of different sizes, time intervals, and durations. These differences were designed to explore the behavior of the method considering events

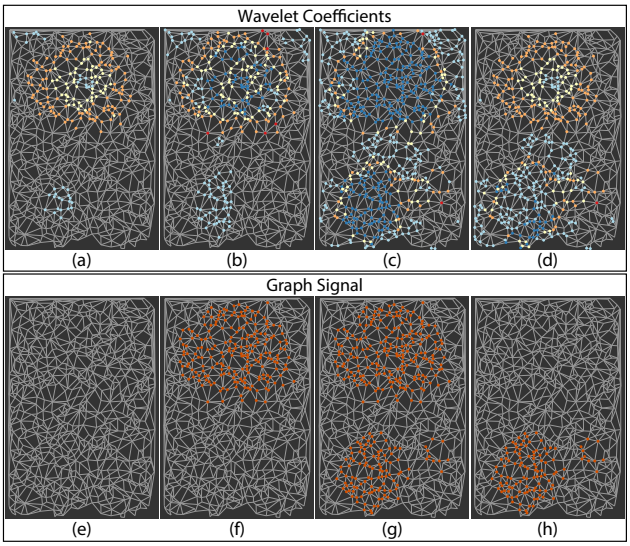


Figure 7: Coefficients and raw data for four distinct time slices. Top row: Coefficients, bottom row: respective raw data.

that are small or large, abrupt or mild, brief or lengthy, and the interaction between nearby events. The time-varying function domain is a graph with 716 nodes and 50 time slices, giving rise to a Cartesian product graph with 35,800 nodes. The system prototype was implemented in JavaScript and the data has been pre-processed in a regular desktop computer, equipped with an i7 Intel processor and 8Gb of RAM, in order to compute wavelets coefficients in each node of the Cartesian product graph. The pre-processing step for this particular synthetic data set took approximately 8 seconds.

Figure 7 shows two sets of time slices from our synthetic data set. Specifically, Figures 7(a) and 7(b) correspond to the classification of nodes in two adjacent time slices t_5 and t_6, while Figures 7(c) and 7(d) show node classification in non-consecutive time slices t_i and t_j. Figures 7(e), 7(f), 7(g), 7(h) depict the function in the same time slices as in Figures 7(a), 7(b), 7(c), and 7(d), respectively.

Figure 7(a) shows the nodes classified according to their wavelet coefficients in time slice t_5. Even though the function has no value in that time slice (all nodes are gray in Figure 7(e)), wavelets coefficients are non zero, indicating that some temporal variation should take place in adjacent time slices. Moreover, nodes are mostly in orange class, meaning that a mid-high frequency variation is about to happen in that region. In summary, Figures 7(a) and 7(e) point out that the data should present an abrupt temporal change from time slice t_5 to the adjacent slices, what is attested when analyzing Figures 7(b) and 7(f) referring to time t_6. Notice from Figure 7(b) that several nodes in the middle of the event becomes low-frequency (blue nodes), surrounded by yellow and orange nodes, indicating a spatial and temporal stability in the center of the event from time slice t_6 to the following slices. The nodes on the border of the event are mostly medium-high frequency, with some high frequency nodes. Those high, mid-high frequency nodes are mainly caused due to spacial variation of the data. Nodes classified as indeterminate indicates that both low (mid-low) as well as high (mid-high) frequencies are present in those nodes due to the simultaneous temporal and spacial variability. The reason is that the event is large enough to be captured by low frequency wavelets.

It is also easy to see that a new region with light-blue nodes appears in Figure 7(b), indicating a variation from t_6 to the following time slices in that region. However, those nodes belong to the mid-low frequency class, meaning that either the function variation is not so "severe" in that region or the data variation will not take place exactly in t_7 but in the following time slices. By clicking in one of the nodes in the new blue region we can visualize its cor-

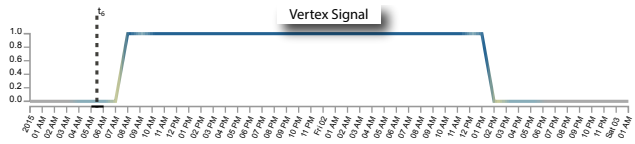


Figure 8: Time series of a node in the blue region in Figure 7(b).

responding time series, with the current time depicted as a dashed vertical line, as depicted in Figure 8. From the time series it is easy to see a function variation from time slice t_8.

Figures 7(c) and 7(g) show another time slice. Coefficients in the middle of the largest events are dominated by low frequency nodes, while nodes in the border tend to be mid-high frequency nodes. Some nodes that are never part of an event are also classified as mid-low frequency, indicating their spatial proximity to other events. However, not all nearby vertices have coefficients above the chosen threshold, so they remain hidden. The time slice just after the end of the largest event is presented in Figures 7(d) and 7(h). Similarly to what happened in Figure 7(a), mid-high frequency nodes indicate a new temporal change related to the vanishment of event.

The discussion above shows the effectiveness of our methodology in revealing events in a time-varying data. The usefulness of our tool in analyzing real data is discussed in the next section.

6 Case Study

In this section, we present a case study where our methodology is used to explore real urban data. The data was provided by the NYC Taxi and Limousine Commission, containing information about the taxis themselves, the drivers, the amount of passengers, the pickup and drop-off times, the duration and distance of the trips, and geographical information for the pickups and drop-offs. We consider only the location and time of the taxi pickups, from August 11th, 2013 to August 18th, 2013. The study considers downtown Manhattan - NY, which is expressed as a graph, where nodes represent street intersections and edges represent the streets connecting them.

In this context, nodes classified as low frequency correspond to locations where the amount of taxi pickups around that intersection changes mildly, in space and time. In contrast, a higher frequency node implies that there is an abrupt change in a nearby location or time. Notice that this relationship does not involve the number of pickups, but the relative change. Consider, for instance, two distinct locations, one with an average of 100 pickups per interval, another with an average of 5. If both locations experience an increase of 10 taxis per interval, the first location would be considered as low frequency, and the second would be classified as high frequency, because of the abrupt relative change in the number of taxi pickups.

Figure 9 depicts the stacked graph of one week of taxi trips, from August 12th to 18th, 2013. Each day was discretized in periods of thirty minutes, or 336 time slices, leading to a Cartesian graph with $1,577,184$ nodes. The pattern of pickups of each day is clearly represented, with a significant decrease on the 18th. Moreover, there is a clear difference between night and day periods, for during the night the number of taxi pickups increases considerably.

Comparison between day and night patterns reveals a decrease between 4pm and 5pm, but the amount varies for each day of the week, from a slight reduction on Monday to a very accentuated valley on Friday. The time in which the number of taxi pickups declines at the end of the night also varies, going from 11pm on Monday to 4:30am on Friday and 4am on Saturdays. Although the decline happens later on Friday, our linked view reveals that the amount of locations with significant taxi pickups at the same period decreases, thus keeping the total of affected nodes roughly the same, as one can clearly see by comparing the shape of Friday against Thursday of Wednesday. Sunday has less trips, evenly distributed among the period.

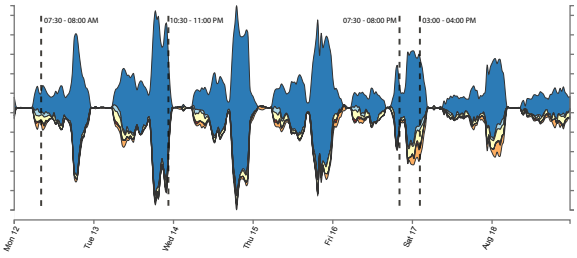


Figure 9: Stacked graph visualization of taxi pickups from August 12th to 18th. A large number of taxi trips happen at night.

Considering the classification of nodes, we see a predominance of low frequency, representing smooth spatio-temporal changes in the number of taxi pickups, even at 5pm, where we have a clear temporal change. Therefore, we can assume that this temporal variation on the number of pickups also occurs as a large, uniform event, that can be characterized as low frequency. In other words, while the number of taxi trips rapidly increases at 5pm, it does so in a large part of the Manhattan graph domain, in a mild manner, without many locations with more abrupt changes. However, there are also nodes classified as high or mid-high frequency, pointing some abrupt variation in the number of pickups at specific locations of the map. The stacked graph reveals that the number of such locations is small compared to the number of low frequency nodes. In less mathematical terms, the number of regions facing an abrupt change in the number of pickups is small compared to regions where the number of pickups changes smoothly.

Our interactive exploration tool allows users to select a time interval in the stacked graph and the linked view reveals the density and the variation of pickups in that particular time interval on Manhattan map. Following the patterns displayed in the stacked graph, we start by exploring the start of the Monday morning rise, which has particular interest, representing a potential rush hour. Figure 10 illustrates the classification of nodes and the density pickup function for four different time slices. The first column represents the time slice from 7:30am to 8am, August 12, 2013. There are some taxi pickups in most of the downtown Manhattan, but the method identified regions of particular changes around the Port Authority bus terminal on 42th street (purple marker 1), Penn Station (2), and at the corner of Liberty street and South End Avenue (3). While the large amount of taxi pickups around two the busy public traffic hubs of New York City are expected, and clearly visible in the density map (top row), the third identified location is curious. The density value associated to that node is lower than in most nodes in the map, but different enough of the local pattern to be identified as a more significant change. We can select this node by clicking on it, which will highlight the node and display the corresponding time series, as is illustrated in Figure 11, where the color of the line corresponds to the classification of the node in each time slice, or gray if below the current threshold. There are visible peaks in the number of taxi pickups during the morning in weekdays. While the value of the function at such peaks is not very high, corresponding to less than a taxi pickup per minute, they represent a clear change in the local dynamics of this particular region.

We identified that this location is close to the WFC Ferry Station, which explains the increase of taxi pickups during the early morning in weekdays. However, this is not the closest street to the Ferry Station, representing a walk of approximately 500 meters, whereas the distance between the Ferry Station and River Terrace, or Vesey Street, is less than 300 meters. Contrarily of River Terrace, the corner of Liberty street and South End avenue has more convenient parking space for taxis waiting for fares. We postulate that passengers from the ferry prefer a longer walk on the esplanade to arrive in a location where they know there would be available taxis.

Figure 10: Four different time slices for the taxi pickup data, corresponding to August 12, 2013, 7:30am to 8am (left); August 13, 2013, 10:30pm to 11pm (mid-left); August 16, 2013, 7:30pm to 8pm (mid-right); and August 17, 2013, 1:30am to 2am (right).

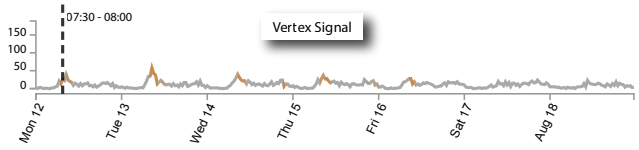


Figure 11: Time series associated to the node representing the intersection of Liberty street and South End avenue.

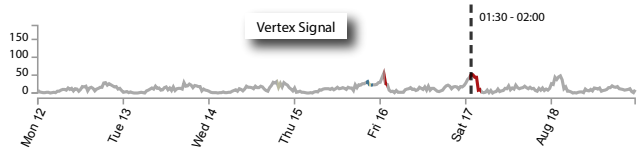


Figure 12: Time series associated to the node representing the node highlighted in the fourth column of Figure 10.

The second column in Figure 10 illustrates the period from 10:30pm to 11pm, Tuesday, August 13, 2013, corresponding to second peak of taxi pickups on Tuesday night. There is an increase around Times Square (4), reflecting the nightlife activity. The highlighted node corresponds to the intersection of 50th street and 8th avenue, near several theaters. Our visual analytics method also identified two locations where the number of pickups changed in a more abrupt manner, one in Tribeca and another near Nolita (orange nodes in the south). While there is a change in these locations, we found no obvious explanation for those peaks.

The third column in Figure 10 shows the period from 7:30pm to 8pm, Friday, August 16, 2013, corresponding to the peak in the number of pickups that happens separating the afternoon and the evening. While the number of pickups around Port Authority bus terminal (1) and Penn Station (2) remain significant, there is an increase in the Midtown east area (5), an area with several high-end hotels. Therefore, we believe that such increase is related to the guests of such hotels leaving to enjoy the nightlife on Friday.

The fourth column in Figure 10 illustrates the period from 1:30am to 2am, Saturday, August 17, 2013, corresponding to an increase in the amount of nodes classified as mid-high frequency, depicted in orange. We can identify regions with high concentration of bars and restaurants, such as South Chelsea (6) and the Lower East Side (7), along with four orange locations on 7*th* avenue, following the 1 and 2 subway lines (red line). We see some nodes with more abrupt changes identified in the middle of the map, located in the intersection of 21st street and 6th avenue (8). The time series of the highlighted node is shown in Figure 12, from which one can see an increase on the number of taxi pickups after midnight, on Fridays, Saturdays and Sundays, with almost two taxi trips per minute starting nearby. We found no obvious explanation for such change during the weekend. It is worth pointing out the effectiveness of the proposed exploratory tool, allowing users to identify/select particular phenomena from the stacked graph which are then revealed in detail on the maps by a linked view. Visual analysis of time series associated specific node can also be accomplished by selecting it.

This case study showcases an important feature of our method, its ability to identify significant changes in the function, regardless of the magnitude of the associated data. Therefore, it can direct the attention of users to the important changes without "contaminating" the visualization with higher magnitude data whose behavior is more uniform. In other words, the method is capable of separat-

ing interesting changes from the usual evolution of the data, which would otherwise require expert knowledge or a deeper analysis using behavioral models. This property render our methodology a competing alternative for time-varying data analysis.

Another case study involving bike trips in Manhattan is described in the supplementary material accompanying this paper.

7 Discussion and Limitations

The provided results and case study clearly show that our approach is able to deal with large amounts of information, while allowing the visualization of spatio-temporal events, regardless of their size.

The proposed classification scheme, based on simulated training data, was satisfactory, enabling informative visualization and data analysis. We attested the correctness of the classification by inspecting the results against the corresponding feature vector. However, a more thorough verification method should be implemented, specially for applications demanding a higher degree of confidence.

Other training data could also be employed to generate classes with specific patterns. For instance, one could split the indeterminate class in order to further discriminate nodes as to their tendency to low or high frequencies, thus increasing the number of classes. The ideal number of classes is application dependent, and several distinct patterns could be revealed by tuning training data.

While the stacked graph plot was quite informative, it is not entirely appropriate when the classes are unbalanced, where classes with few elements can be difficult to visualize. However, this minority information can indicate outliers. Moreover, when a large temporal interval is considered, the plot can become quite dense, hindering the visualization of smaller variations in the classes.

There are additional issues when handling large data sets, as real-time wavelets coefficients calculation can only be achieved for small graphs. Even with the adopted mechanism to avoid the construction of the Laplacian matrix for of whole Cartesian product graph, computational times can still become a significant factor when dealing with massive data. This cost can be reduced by aggregating time slices in a coarser time discretization, as we did in the case study. However, aggregation operates as a low-pass filter that naturally disguises small temporal and spatial variations. A possible alternative for this issue is to incorporate focus-plus-context data exploration, which we consider a relevant, but non-trivial extension of our methodology.

Another weakness of our approach is the difficulty of discriminating spatial and temporal changes, considering only the graph wavelets coefficients. Wavelets coefficients could be combined with other information to perform such discrimination.

The limitations above are considered avenues for future work. The provided results clearly show the usefulness and flexibility of graph wavelets to support visualization tasks, encouraging a multitude of further developments and investigations.

8 Conclusion

In this work we have proposed a novel visual analytic methodology for analyzing time-varying data that combines graph wavelet theory, pattern classification, and stacked graph visual metaphor in a linked view visualization environment. The proposed method is versatile, robust, and quite effective to reveal important events given by variation in the data. The usefulness of the proposed methodology was attested through a set of tests and a case study, rendering it an attractive methodology for many visualization applications.

Acknowledgements

Grants #2013/14089 – 3, #2013/19760 – 5, and #2014/12815 – 1, São Paulo Research Foundation (FAPESP). The views expressed are those of the authors and do not reflect the official policy or position of the FAPESP. The authors thank Dr. Paulo Pagliosa, UFMS, for the help with the video and the reviewers for their comments.

References

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data - A systematic view. *Computers & Graphics*, 31(3):401–409, June 2007.

[2] G. Andrienko and N. Andrienko. Spatio-temporal aggregation for visual analysis of movements. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, pages 51–58. IEEE, Oct. 2008.

[3] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The state of the art in visualizing dynamic graphs. In *Proceedings of the Eurographics Conference on Visualization (EuroVis 14)State of The Art Reports*, 2014.

[4] T. Bıyıkoglu, J. Leydold, and P. F. Stadler. *Laplacian eigenvectors of graphs*. Lecture notes in mathematics. Springer, 2007.

[5] D. Chu, D. A. Sheets, Y. Zhao, Y. Wu, J. Yang, M. Zheng, and G. Chen. Visualizing hidden themes of taxi movement with semantic transformation. In *IEEE Pacific Vis. Symp.*, pages 137–144, 2014.

[6] M. Crovella and E. Kolaczyk. Graph wavelets for spatial traffic analysis. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 3, pages 1848–1857. IEEE, 2003.

[7] H. Doraiswamy, N. Ferreira, T. Damoulas, J. Freire, and C. T. Silva. Using Topological Analysis to Support Event-Guided Exploration in Urban Data. *IEEE Trans. Vis. Comput. Graphics*, 20(12):2634–2643, Dec. 2014.

[8] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: a study of New York City taxi trips. *IEEE Trans. Vis. Comput. Graphics*, 19:2149–58, 2013.

[9] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan. TripVista: Triple Perspective Visual Trajectory Analytics and its application on microscopic traffic data at a road intersection. In *2011 IEEE Pacific Visualization Symposium*, pages 163–170. IEEE, Mar. 2011.

[10] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, Mar. 2011.

[11] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: visualizing theme changes over time. In *IEEE Symposium on Information Visualization. Proceedings*, pages 115–123. IEEE Comput. Soc, 2000.

[12] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: a survey. *IEEE Trans. Vis. Comput. Graphics*, 19(3):495–513, Mar. 2013.

[13] D. M. Mohan, M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet. Wavelets on graphs with application to transportation networks. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1707–1712. IEEE, Oct. 2014.

[14] M. F. Möller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.

[15] J. Pu, S. Liu, P. Xu, H. Qu, and L. M. Ni. MViewer: mobile phone spatio temporal data viewer. *Front. of Comput. Sci.*, 8:298–315, 2014.

[16] A. Sandryhaila and J. Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31:80–90, 2014.

[17] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013.

[18] G. Sun, Y. Liu, W. Wu, R. Liang, and H. Qu. Embedding Temporal Display into Maps for Occlusion-Free Visualization of Spatio-temporal Data. In *2014 IEEE Pacific Vis. Symp.*, pages 185–192. IEEE, Mar. 2014.

[19] N. Tremblay and P. Borgnat. Graph wavelets for multiscale community mining. *IEEE Trans. on Signal Process.*, 2014.

[20] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. van de Wetering. Visual traffic jam analysis based on trajectory data. *IEEE Trans. on Vis. Comp. Graphics*, 19(12):2159–68, Dec. 2013.

[21] W. Zeng, C.-W. Fu, S. M. Arisona, and H. Qu. Visualizing Interchange Patterns in Massive Movement Data. *Comp. Graph. Forum*, 32(3pt3):271–280, June 2013.

[22] X. Zhu and M. Rabbat. Approximating signals supported on graphs. In *ICASSP*, pages 3921–3924. Citeseer, 2012.