

# Visual Analytics via Graph Signal Processing

Alcebiades Dal Col  
Federal University of Espírito Santo  
Department of Mathematics

L. Gustavo Nonato  
University of São Paulo  
Institute of Mathematical and Computer Sciences

**Abstract**—This dissertation presents an overview of the extension of the classical signal processing theory to graph domains. Furthermore, we introduce in this dissertation a novel method for visual analysis of dynamic networks, which relies on the graph wavelet theory. Our method enables the automatic analysis of a signal defined on the nodes of a network. We use a fast approximation of the graph wavelet transform to derive a set of wavelet coefficients, which are then used to identify activity patterns on large networks, including their temporal recurrence. The wavelet coefficients naturally encode spatial and temporal variations of the signal, leading to an efficient and meaningful representation. This method allows for the exploration of the structural evolution of the network and their patterns over time. The effectiveness of our approach is demonstrated using different scenarios and comparisons involving real dynamic networks.<sup>1</sup>

## I. INTRODUCTION

The emerging field of signal processing on graphs has been growing substantially in the last years. The theory is based on irregular structures called graphs. Any data whose elements have relations with each other can be modeled as a graph. Elements are named graph nodes and relations are edges. The ability to analyze a signal defined on such an irregular environment is a remarkable characteristic of this theory, extending the classical signal processing on regular domains to a more complex scenario.

Graph signal processing has a well-established theoretical and computational framework. The theory formulation is strongly based on mathematical principles with the concern of being computationally efficient, which is of unprecedented importance today since we are living the advent of new technologies. When working with the tools of graph signal processing, the construction of the graph and choice of an appropriate signal is of paramount importance.

Within the universe of graph signal processing, we focus on spectral graph wavelets. The concept of graph wavelets is strongly related to the classical version on regular domains. However, some operations quite simple on that domain, such as the shift operation, are harder to define precisely on the graph scenario. Thus, the definition of the graph wavelets is based on the graph spectral theory, where, for instance, the shift operation can be interpreted as a multiplication.

The use of graph wavelets is still incipient with few applications [1]–[3]. In this sense, we contribute to the visual analysis of dynamic networks. The spectral graph wavelets are in the background of an interface that helps users to gain knowledge about behavior of elements forming a dynamic network.

## II. SPECTRAL GRAPH THEORY

Spectral graph theory refers to the analysis of graphs by means of linear algebra concepts. In this section, we review the spectral graph theory and spectral graph wavelet transform.

### A. Basic Definitions

A *graph*  $G$  is a structure composed of a set of nodes and a set of edges connecting pairs of nodes. Consider a graph  $G = (V, E)$  with a node set  $V = \{\tau_1, \dots, \tau_n\}$  and an edge set  $E$ , where  $n$  is the number of nodes. A weight  $w_{ij} \in \mathbb{R}_+$  is associated to each pair of nodes  $\tau_i$  and  $\tau_j$  in  $V$ , encoding how strong the relationship between them is. Since we do not consider self-loops  $w_{ii} = 0$ ,  $i = 1, \dots, n$ . We set  $w_{ij} = 0$  if there is no edge connecting nodes  $\tau_i$  and  $\tau_j$ ,  $i \neq j$ . Assigning weights to the relations between nodes makes the graph  $G$  a *weighted graph*.

The *adjacency matrix* of a graph  $G$  is a  $n \times n$  matrix  $W = (w_{ij})$  with entries  $w_{ij}$ . We assume that the relation between two nodes is always symmetric, then  $w_{ij} = w_{ji}$  and the graph  $G$  is called *undirected graph*. The *graph Laplacian*, or *combinatorial graph Laplacian*, is a matrix given by  $L = D - W$ , where  $W$  is the adjacency matrix and  $D = (d_{ii})$  is the *degree matrix*, a diagonal matrix with entries  $d_{ii} = \sum_k w_{ik}$  (sum of the weights of all edges incident to node  $\tau_i$ ). Since we consider only undirected graphs, both adjacency matrix and graph Laplacian are symmetrical.

The graph Laplacian  $L$  is real, symmetric, and semi-positive definite, then  $L$  has a complete set of orthonormal eigenvectors  $u_\ell$ ,  $\ell = 1, \dots, n$ , with corresponding non-negative real eigenvalues  $\lambda_\ell$ ,  $\ell = 1, \dots, n$ , which we assume ordered in non-decreasing order  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ .

Additional information can be integrated into the graph through a real-valued function  $f : V \rightarrow \mathbb{R}$ . The function  $f$  assigns a real value  $f(\tau)$  to each node  $\tau \in G$ . We term the function  $f$  as *signal*, or *graph signal*. The set  $f(V) = \{f(\tau) | \tau \in V\}$  can be interpreted as a vector in  $\mathbb{R}^n$ , where the  $i$ -th entry of the vector corresponds to the value of the function in node  $\tau_i$ . The graph Laplacian eigenvectors  $u_\ell : V \rightarrow \mathbb{R}$ ,  $\ell = 1, \dots, n$  can also be seen as signals on  $G$  since they are defined on the node set  $V$ .

Let  $f : V \rightarrow \mathbb{R}$  be a signal defined on a graph  $G$ , the *set of zero crossings* of  $f$  on  $G$  is  $Z_G(f) = \{(\tau_i, \tau_j) \in E | f(\tau_i)f(\tau_j) < 0\}$ . In other words,  $Z_G(f)$  is the set of edges connecting nodes where  $f$  has different signs (positive and negative). The larger the number of zero crossings  $|Z_G(f)|$ , the more  $f$  changes its sign across the graph.

<sup>1</sup>This work relates to a Ph. D. thesis

The eigenvectors associated with larger eigenvalues oscillate more throughout the graph. Therefore, the eigenvalues and eigenvectors of the graph Laplacian can be interpreted as frequencies and Fourier modes, that is, the larger the eigenvalue, the more oscillatory the corresponding eigenvector.

### B. Graph Fourier Transform

Given a signal  $f : V \rightarrow \mathbb{R}$  defined on the nodes of  $G$ , the *graph Fourier transform* of  $f$  is a function  $\hat{f}$  defined by  $\hat{f}(\lambda_\ell) = \langle f, u_\ell \rangle = \sum_{i=1}^n f(\tau_i) u_\ell(\tau_i)$ . The *inverse graph Fourier transform* reconstructs the initial signal  $f$  with the linear combination of the eigenvectors  $u_\ell$  weighted by  $\hat{f}(\lambda_\ell)$ , that is,  $f = \sum_{\ell=1}^n \hat{f}(\lambda_\ell) u_\ell$ .

Similar to the classical Fourier transform, the graph Fourier transform reveals frequencies present in a signal  $f$ . The presence of high frequency Fourier coefficients indicates that a signal varies abruptly in some region(s) of the graph, while the presence of low frequency Fourier coefficients suggests smooth signal variation.

### C. Spectral Graph Wavelet Transform

The *graph wavelet* in node  $\tau$  at scale  $s$  can be obtained by  $\psi_{s,\tau} = \sum_{\ell=1}^n u_\ell(\tau) \hat{g}(s\lambda_\ell) u_\ell$ . The graph wavelet coefficients are formally given by the inner product between the signal and the graph wavelets  $W_f(s, \tau) = \langle f, \psi_{s,\tau} \rangle$ . The wavelet coefficient in node  $\tau$  at scale  $s$  can be calculated by  $W_f(s, \tau) = \sum_{\ell=1}^n \hat{g}(s\lambda_\ell) \hat{f}(\lambda_\ell) u_\ell(\tau)$ . The function  $\hat{g}(s\lambda)$  is called *wavelet kernel*. The graph wavelet transform is then determined by the kernel  $\hat{g} : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ , which is selected as

$$\hat{g}(x) = \begin{cases} x^2 & \text{for } 0 \leq x < 1 \\ -5 + 11x - 6x^2 + x^3 & \text{for } 1 \leq x \leq 2 \\ 4x^{-2} & \text{for } 2 < x \end{cases} \quad (1)$$

The scales used to generate the wavelet kernels are logarithmically sampled between  $s_1$  and  $s_r$  ( $s_1, s_2, \dots, s_r$ ), where  $r$  is the number of scales,  $s_1 = 2/\lambda_n$ , and  $s_r = 40/\lambda_n$ .

**Scaling function:** It is introduced an additional function to stably represent low frequency eigenvectors, the *scaling function*. This waveform is constructed by a real-valued function  $\hat{h}$  that acts as a low-pass kernel with  $\hat{h}(x) = \gamma \exp\left(-\left(\frac{10x}{0.3\lambda_n}\right)^4\right)$ .

The parameter  $\gamma$  is chosen such that  $\hat{h}(0)$  is equal to the maximum value of  $\hat{g}$ . The *scaling function coefficient* is then given by  $S_f(\tau) = \sum_{\ell=1}^n \hat{h}(\lambda_\ell) \hat{f}(\lambda_\ell) u_\ell(\tau)$ .

For each node  $\tau \in V$ , the spectral graph wavelet transform produces a scaling function coefficient  $S_f(\tau)$  and a sequence of wavelet coefficients  $W_f(s_1, \tau), \dots, W_f(s_r, \tau)$ . We assumed the wavelet coefficients ordered from lowest to highest wavelet frequency ( $S_f(\tau), W_f(s_r, \tau), \dots, W_f(s_1, \tau)$ ). In our implementation, we considered the scaling function and seven wavelet kernels (Figure 1).

## III. ILLUSTRATIVE INTERPRETATION OF GRAPH WAVELETS

A *graph spectral filter*, or *kernel*,  $\hat{g} : \Lambda \rightarrow \mathbb{R}$  is a function defined on the *spectral domain* ( $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ ), which

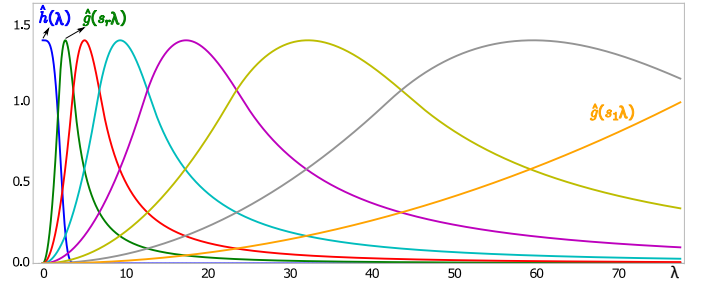


Fig. 1. Scaling function and wavelet kernels.

associates a scalar  $\hat{g}(\lambda_\ell)$  with each eigenvalue  $\lambda_\ell \in \Lambda$ . The graph Fourier transform  $\hat{f} : \Lambda \rightarrow \mathbb{R}$  can be seen as a particular instance of graph spectral filter. A *dictionary of graph spectral filters*  $\{\hat{g}_m \mid m = 1, 2, \dots, M\}$  is a set of graph spectral filters  $\hat{g}_m : \Lambda \rightarrow \mathbb{R}$ ,  $m = 1, 2, \dots, M$ , where  $M$  is the number of graph spectral filters in the dictionary.

Given  $f$  and a dictionary  $\{\hat{g}_m \mid m = 1, 2, \dots, M\}$ , a spectral graph wavelet transform  $W_f : \{1, 2, \dots, M\} \times V \rightarrow \mathbb{R}$  can be defined by  $W_f(m, \tau) = \sum_{\ell=1}^n \hat{g}_m(\lambda_\ell) \hat{f}(\lambda_\ell) u_\ell(\tau)$ . By varying  $m$  while keeping  $\tau$  fixed, we obtain  $M$  wavelet coefficients associated with node  $\tau$ ,  $W_f(1, \tau), W_f(2, \tau), \dots, W_f(M, \tau)$ . Selecting the graph spectral filter  $\hat{g}_m$  as a band-pass filter, where small values of  $m$  correspond to low-pass filters and large values to high-pass filters, makes the coefficients encode the behavior of the signal on each node. For instance, choosing the dictionary of graph spectral filters equal to  $\{\hat{h}(\lambda), \hat{g}(s_r\lambda), \dots, \hat{g}(s_1\lambda)\}$ , where the kernel  $\hat{g}$ , scaling function  $\hat{h}$ , and scales  $s_1, \dots, s_r$  are defined as in Section II, produces exactly the *spectral graph wavelets* (SGW) [4]. This dictionary is called *spectral graph wavelet dictionary*, or succinctly *SGW dictionary*.

### A. Spectrum-Adapted Graph Wavelet Transform

The *uniform translates* are translated versions of each other uniformly distributed on the spectral domain. In more mathematical terms, there are a function  $\hat{g}^U$  and a constant  $a$  such that the graph spectral filters  $\hat{g}_m$  can be written by  $\hat{g}_m(\lambda) = \hat{g}^U(\lambda - ma), \forall \lambda \in [0, \lambda_n]$ , for  $m = 1, 2, \dots, M$ .

The uniform translates are then modified to adapt itself to the spectrum of a given graph. The density of the graph Laplacian eigenvalues is estimated using a fast approximation mechanism, and the larger the number of eigenvalues in a certain region of the spectral domain, the narrower the graph spectral filters are in that region (Figure 2).

The dictionary of uniform translates adaptable to the spectrum of a specific graph forms the *spectrum-adapted graph wavelets* (SAGW) [5]. This dictionary is called *spectrum-adapted graph wavelet dictionary*, or briefly *SAGW dictionary*.

### B. Practical Differences between the Spectral Graph Wavelet Transforms

A *comet graph* is a graph formed combining a path graph and a star graph, the latter centered in one of the extremities of the path graph (Figure 3). The graph Laplacian eigenvalues

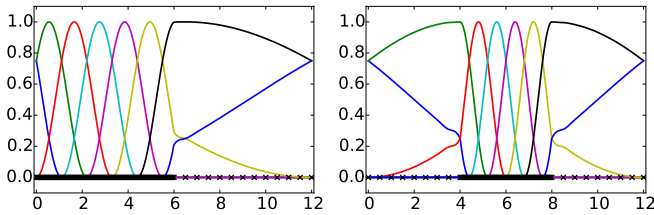


Fig. 2. Uniform translates adapting to graph Laplacian spectra. The graph Laplacian eigenvalues are indicated by  $x$  markers on the horizontal axes.

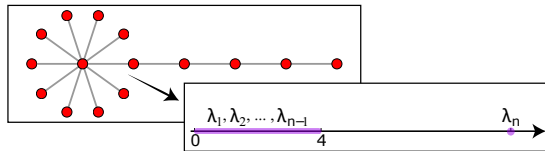


Fig. 3. Comet graph and corresponding spectral distribution.

of a comet graph have a peculiar distribution, as the difference between the two largest eigenvalues is much larger than the difference between the other eigenvalues. Although the second largest eigenvalue is distant from the largest eigenvalue, it corresponds to the second highest frequency in the spectrum of the comet graph.

Figure 4 depicts the spectral graph wavelet and spectrum-adapted graph wavelet dictionaries, both defined in the same spectral domain, the spectrum of the comet graph. The graph wavelet transform derived from each dictionary are quite different, since the SGW dictionary is designed to evenly cover the spectral domain and the SAGW dictionary adapts to the spectral distribution. As one can clearly see in Figure 4, when eigenvalues are not evenly distributed on the spectral domain, as is the case of the comet graph, the SAGW dictionary better fits the frequency bands.

We create a comet graph with 64 nodes whose star extremity contains 30 nodes. The signal is defined as the second largest graph Laplacian eigenvector (signal panel in Figure 5). Figure 5 depicts the wavelet coefficients of the specified node computed using both graph wavelet transforms. The SGW dictionary led to wavelet coefficients with dominant frequencies in the central frequencies (4 and 5), not properly characterizing the high frequency nature of the signal under analysis. In contrast, the SAGW dictionary generated wavelet coefficients with larger intensity in the coefficients related to high frequencies, better capturing the pattern of variation of the signal.

#### IV. VISUAL ANALYSIS OF DYNAMIC NETWORKS

This section explores data modeled as dynamic network, where the network topology can change over time [6]–[12]. Our exploration is guided by a signal associated with the nodes of the network. By analyzing the wavelet coefficients in specific frequencies, our visual analytics method is able to characterize the behavior of the network over time.

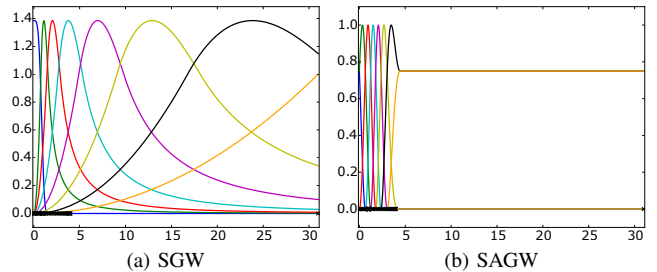


Fig. 4. Graph spectral filters of the SGW and SAGW dictionaries

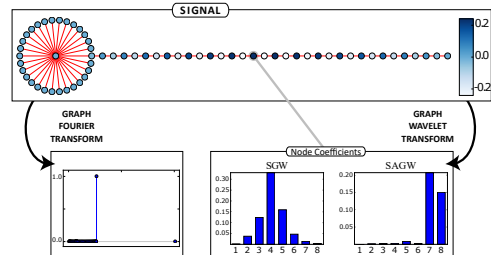


Fig. 5. Second largest graph Laplacian eigenvector defined on a comet graph.

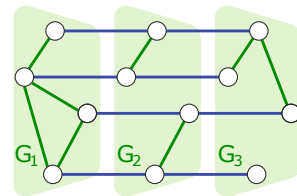


Fig. 6. Dynamic network model.

#### A. Dynamic Network Model

We model a dynamic network as a sequence of  $N$  undirected weighted graphs  $G_1, \dots, G_N$ , where  $N$  is the number of time slices. For each time slice  $k \in Z = \{1, \dots, N\}$ ,  $G_k = (V \times \{k\}, E_k)$  is a graph with nodes  $(\tau, k) \in V \times \{k\}$  and edge set  $E_k \subseteq (V \times \{k\}) \times (V \times \{k\})$ . A weight  $w_{ij}^k$  is associated with each edge in  $E_k$  connecting nodes  $(\tau_i, k)$  and  $(\tau_j, k)$ . In order to apply the graph wavelet theory, as described in Section II, we add *temporal edges* between corresponding nodes in adjacent time slices (blue lines in Figure 6), resulting in a graph  $G = (V \times Z, E)$  with node set  $V \times Z = (V \times \{1\}) \cup (V \times \{2\}) \cup \dots \cup (V \times \{N\})$ , and edge set  $E = (E_1 \cup E_2 \cup \dots \cup E_N) \cup (E_{1,2} \cup E_{2,3} \cup \dots \cup E_{N-1,N})$ , where  $E_{k,k+1}$  is the set of edges connecting nodes  $(\tau, k)$  and  $(\tau, k+1)$  in consecutive time slices.

Since consecutive time slices are connected by temporal edges in  $G$ , the graph wavelet theory can be applied to analyze a signal defined on the nodes of  $G$ , allowing the detection of spatio-temporal variations of the signal as well as the identification of important phenomena in the dynamic network.

#### B. Edge Weights & Node Signal

The weight of the edges dictates how strongly the graph wavelet transform considers each connection between nodes.

Larger values correspond to stronger connections, with more impact on the coefficients. An edge weight equal to zero corresponds to an edge that is not present on the graph. The weight of the temporal edges can also be handled to control the effect of temporal variation on the resulting wavelet coefficients.

To explore interpersonal contact networks in Section IV-E, we correspond people to graph nodes, with associated signal defined by the number of interpersonal contacts over a period of time. Since each edge represents contact as well, this signal is equivalent to the degree of the node in the corresponding time slice, which is a purely topological measure. This approach enables the analysis of the topological evolution of dynamic networks.

### C. Node Classification

Each node  $\tau$  of  $G$  can be classified using its *wavelet coefficient feature vector*, a feature vector containing the wavelet coefficients of node  $\tau$  ordered from lowest to highest wavelet frequency,  $FV_\tau = (S_f(\tau), W_f(s_r, \tau), \dots, W_f(s_1, \tau))$ . Each scale is normalized independently, dividing by the standard deviation and applying a log scale, dividing then by the log of the maximum coefficient. Such transformation brings each coefficient to the range  $[0,1]$ , while making the classification less sensitive to the amplitudes of the original signal.

Isolated nodes in each time slice are assigned to a specific class, denoted as *Zero* class and represented by the white color. The other nodes are then separated into five classes using a simple analogy to mechanical torque. The torque indicates whether the feature vector  $FV_\tau$  has predominantly low or high frequency.

The torque values are normalized into the interval  $[-1,1]$  by simply dividing by the maximum possible torque. Nodes with torque values below  $-0.3$  are classified as low frequency nodes (dark blue), between  $-0.3$  and  $-0.05$  are considered medium-low (light blue), between  $-0.05$  and  $0.05$  are indefinite (yellow), between  $0.05$  and  $0.3$  correspond to medium-high (orange), and above  $0.3$  are high frequency nodes (red). Albeit empirically defined, these limits are robust enough to provide reliable results without fine tuning. The classification step is performed using five classes, where the yellow class corresponds to a balance between low and high frequencies.

The classification leads to a straightforward interpretation, considerably easier than interpreting the wavelet coefficients directly, where a red node indicates an abrupt change in the signal and a blue node a smooth change in the signal, considering both spatial and temporal neighborhood.

### D. From Wavelet Coefficients to Visualization

The proposed visualization aims to provide analytical resources to identify patterns in the dynamics of the network and show how these patterns evolve over time. Specifically, the visualization should enable:

**Goal 1.** *The analysis of the network as a whole.*

**Goal 2.** *The analysis of each node.*

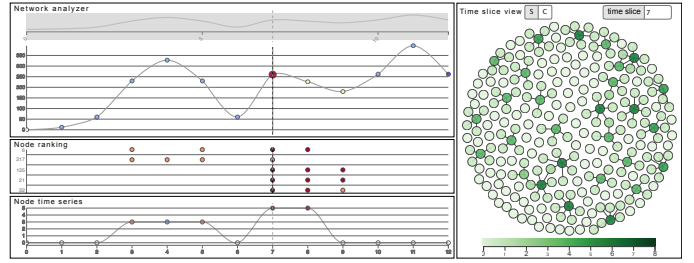


Fig. 7. Proposed interface for dynamic network exploration.

**Goal 3.** *The identification of prominent nodes.*

**Goal 4.** *The identification of similar nodes.*

The proposed visualization is composed of four linked views: network analyzer, node ranking, node time series, and time slice view, as illustrated in Figure 7.

**Network analyzer:** This panel summarizes the general behavior of the network over time via a time series. Each time slice is represented as a circle, whose vertical position is given by the sum of signal of all nodes in that time slice. The coloration of the circle corresponds to the color of the node class with more nodes in that time slice, relative to the maximum number of nodes of that class considering all time slices. This scheme highlights differences in the classification of the network, directing the user to pattern changes.

**Time Slice View:** The main motivation of this panel is to visually represent the relationships of a node at a given time slice. The nodes are positioned using Fruchterman-Reingold force-directed algorithm. The user is allowed to choose between signal and node classification visualizations. By positioning the mouse over a node its contact partners and edges connecting them are highlighted. Further, additional information is revealed, such as node identification and class.

**Synthetic Dynamic Network Exploration:** We generated a synthetic dynamic network containing 250 nodes and 13 time slices. The signal associated with a node  $(\tau, k)$  is the number of edges in  $G_k$  incident on the node. The weights of the edges are all equal to one. There are two large spatial events (leftmost panel in Figure 8) in the network, corresponding to an activity peak (many edges) at time slice 4, which decrease in size until time slice 6. At time slice 7, several small spatial events (central panel in Figure 8) appear, with some of them disappearing on the next time slices. At time slice 10, another two large spatial events (rightmost panel in Figure 8) are created, which reach an activity peak at time slice 11 and decrease at time slice 12.

Figure 7 depicts the network analyzer for the synthetic dynamic network. It shows that the activity level (number of edges) of the synthetic dynamic network increases and decreases, with three peaks, at time slices 4, 7, and 11. Most time slices are classified as light blue, which indicates the predominance of nodes facing smooth signal variation. A high concentration of this kind of node suggests a large event happening on the dynamic network, which is the case for time slices 4 and 10, as illustrated in the leftmost and rightmost

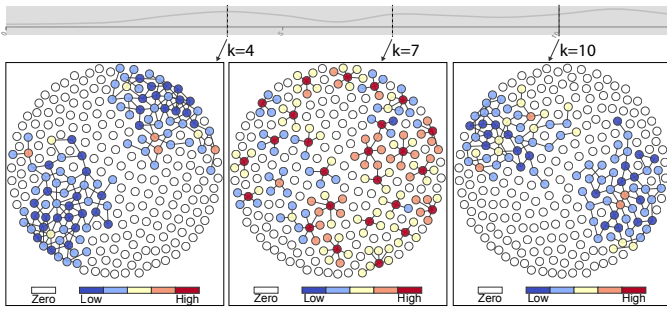


Fig. 8. Node classification for the synthetic dynamic network.

panels of Figure 8. Time slice 10 has height similar to time slice 7 in the network analyzer, indicating similar activity level, but these two time slices differ in their classification. In fact, time slice 7 is classified as red and time slice 10 is classified as light blue. This change in the classification indicates a difference in the current behavior of the network, albeit with almost the same level of activity. A red time slice indicates the predominance of high frequency nodes. Therefore time slice 7 contains nodes facing abrupt signal variation, revealing small events probably with central nodes since the signal corresponds to the degree of the nodes. As illustrated in the middle panel of Figure 8, this is indeed the case, corresponding to several small events that suddenly appeared at time slice 7.

**Node ranking:** This panel depicts the nodes ordered according to their relevance. Since relevance is application dependent, we adopted a simple mechanism to rank nodes, where relevance is related to the higher frequencies. Therefore, we consider a node as relevant in a time slice if it is classified as orange or red, and we rank a node according to the total number of time slices in which the node is considered relevant, on a given time frame. Considering the whole synthetic dynamic network, the most relevant entity corresponds to the node identified by the number six, which was classified as orange/red in four time slices (middle left of Figure 7). Indeed, this node was involved in one of the large events in the early time slices, near the border of the event, which led to a higher frequency, and was the center of a small event later.

By clicking on a node in the node ranking, the order changes to organize the nodes according to their similarity to the selected node. Similarity is defined as the intersection between the two corresponding relevance arrays, displayed in the node ranking. Considering the synthetic dynamic network, this option would quickly reveal nodes similar to node six, namely, nodes that were center of small events.

This panel acts as a visual index of the relevant nodes, allowing users to quickly identify nodes and time slices of most relevant changes. This is a new and important feature for the visualization of dynamic networks, bridging the gap between exploring the network as a whole and specific nodes, reducing the amount of direct inspections required to find relevant information.

**Node time series:** Once an entity is selected in one of the panels, the corresponding time series is depicted below the

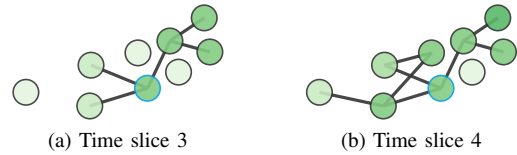


Fig. 9. Neighborhood of node six in the synthetic dynamic network

node ranking panel. The height of the circles represents the signal and the coloration indicates the node classification in each time slice, as illustrated on the bottom left of Figure 7.

Figure 7 depicts the node time series of node six in the synthetic dynamic network. The time series indicates two periods of activity, from time slice 3 to 5 and from time slice 7 to 8. The first one may seem curious to the reader, since the node has the same number of contacts with different classifications, but the pattern hidden in the first period reveals the strength of the graph wavelets. Indeed, Figure 9 shows the neighborhood of node six in time slices 3 and 4. The way node six interacts with its neighbors does not change over time, however some neighbors of this node expand their contact beyond node six, making node six a member of a slightly larger contact network and no longer a central node of a small group.

The panels comprising our visual analytics tool are interactive and linked, allowing users to explore the information in several ways. The network analyzer guides the user through the temporal evolution of the network (Goal 1), including consistent network states. The node time series and time slice view, combined, allow for the exploration of the signal and classification of each node (Goal 2). The node ranking panel summarizes which entities are more prominent (Goal 3), and allows for the identification of entities with similar patterns of relevance over time (Goal 4).

Further, the use of graph wavelet transform, enabled by these panels, creates a visualization tool whose performance surpasses the current state-of-the-art, allowing users to explore and discover gist information and patterns efficiently, with less interaction and mental effort.

#### E. Usage Scenario: High School Dynamic Contact Network

The high school dataset contains face-to-face contact information between 180 students from a school in France, during nine days in November of 2012 (<http://www.sociopatterns.org>). Each student belongs to one of five different classes. The original data provides contact information between students in intervals of 20 seconds, which we aggregated further by creating a time slice every 6 minutes, for a total of 2,027 time slices. The resulting dynamic network has 14,788 edges, not counting the temporal edges. The signal associated with each node is the number of face-to-face contacts made by the corresponding person in the corresponding time slice, and the weight of each edge corresponds to the number of face-to-face contacts between the two involved people. The calculation of the wavelet coefficients took approximately 7 seconds, and the

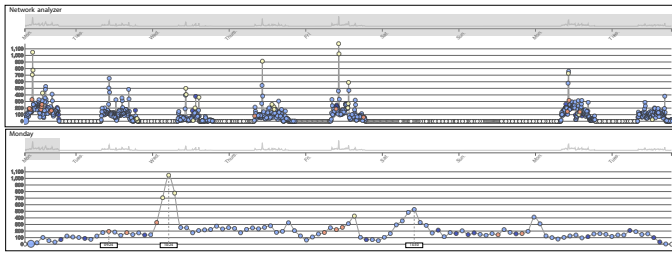


Fig. 10. Network analyzer for the high school dataset.

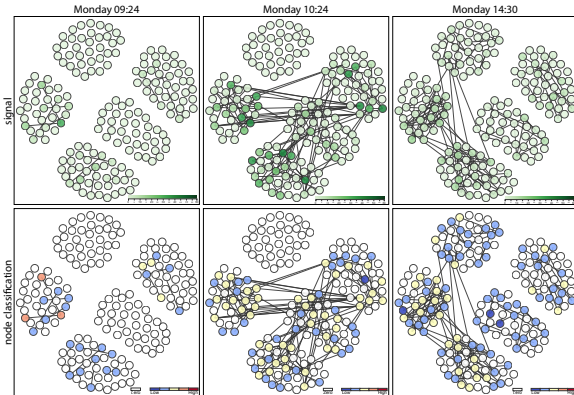


Fig. 11. Signal and node classification for the high school dataset on the first Monday at 9:24, 10:24, and 14:30.

whole preprocessing step took approximately 20 seconds on a regular i7 computer, using around 1 Gb of memory.

We start the exploration of the high school dynamic network using the network analyzer, which shows the total number of contacts made by students (Figure 10). There are seven periods of activity, corresponding to the seven weekdays where the students interacted. Wednesday has a much lower level of activity than the other weekdays, which indicates a lower level of interactions between the subjects. Indeed, on this day, the students had exams, which lessen the level of interaction.

The predominance of light blue nodes indicates that in most of the time slices the signal faces smooth variation, meaning that children interact primarily in balanced contact groups. Some time slices, however, they are classified as orange, mostly on Monday, suggesting sudden changes in the number of contacts and directing our attention to a pattern different from the one mentioned above. The network analyzer allows for zooming particular time-intervals, in particular the zoomed-in view at the bottom in Figure 10 shows a view of activities on the first Monday. To demonstrate the usefulness of the classification scheme used in the network analyzer, we selected, in the zoomed Monday, three time slices with different classification, corresponding to times 9:24 (orange), 10:24 (yellow), and 14:30 (light blue), illustrated in Figure 11.

The 9:24 time slice is the first to be classified as orange on the dynamic network. From the network analyzer only, one can infer that this time slice does not contain much activity. The 9:24 time slice is part of a consistent level of activity that covers the morning period. While most of the neighboring

time slices are classified as light or dark blue, the 9:24 time slice was classified as orange. This classification implies a difference in the contact pattern. Since the graph nodes are predominantly low frequency, the presence of a few higher frequency nodes would be sufficient to lead the time slice to a higher frequency classification.

By inspecting the 9:24 time slice, as illustrated in Figure 11, it is clear that this time slice mostly contains pairs of contacts, with low signal. The nodes are mostly light blue with a few yellows. However, three connected nodes on the leftmost student class present a higher signal, leading to an orange classification, which leads to the classification of this time slice as orange as well. In more practical terms, this phenomenon probably represents a transmission of a large amount of information, where the center node corresponds to the child who transmitted the information.

The activity peak for Monday happened at 10:24, where the sum of the signal surpassed a thousand. Since the sum of the signal corresponds to the total number of contacts and the time slice is classified as yellow, this time slice contains a large event, with some moderated signal differences. As illustrated in Figure 11, this is clearly the case.

The 14:30 time slice also contains a considerable amount of activity, indicated by the height in the network analyzer, and beyond that this time slice is classified as light blue. The combination of increased height and light blue classification means that this time slice contains a large event, without signal peaks. Indeed, as illustrated in Figure 11, the signal varies smoothly across a large event, with some nodes classified as yellow, but with the relative majority of light blue nodes. This large event is harder to notice when compared to the synthetic dynamic network where we had a controlled environment, highlighting the potential of the graph wavelet transform in the exploration of dynamic networks with the aid of visual analysis.

These analyses highlight the potential of the network analyzer, greatly reducing the necessity for inspection of individual time slices. While the proposed interface allows for closer inspection of each time slice and node, the network analyzer efficiently provides overall information about the evolution of the dynamic network.

The interface and source codes are available on a GitHub page located at <https://github.com/AlcebiadesDalColJunior/>.

## V. PUBLICATIONS

As a result of the Ph.D. thesis, we have two articles [13], [14] and a book chapter [15].

## REFERENCES

- [1] N. Tremblay and P. Borgnat, "Graph wavelets for multiscale community mining," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5227–5239, 2014.
- [2] P. Valdivia, F. Dias, F. Petronetto, C. T. Silva, and L. Nonato, "Wavelet-based visualization of time-varying data on graphs," in *IEEE Conference on Visual Analytics Science and Technology*. Institute of Electrical and Electronics Engineers, 2015.

- [3] D. M. Mohan, M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Wavelets on graphs with application to transportation networks," in *IEEE International Conference on Intelligent Transportation Systems*. Institute of Electrical and Electronics Engineers, 2014, pp. 1707–1712.
- [4] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [5] D. I. Shuman, C. Wismeyer, N. Holighaus, and P. Vandergheynst, "Spectrum-adapted tight graph wavelet and vertex-frequency frames," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4223–4235, 2015.
- [6] B. Bach, E. Pietriga, and J.-D. Fekete, "GraphDiaries: Animated transitions and temporal navigation for dynamic networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 5, pp. 740–754, 2014.
- [7] B. Bach, N. Henry-Riche, T. Dwyer, T. Madhyastha, J.-D. Fekete, and T. Grabowski, "Small MultiPiles: Piling time to explore temporal patterns in dynamic networks," *Computer Graphics Forum*, 2015.
- [8] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic, "Time Curves: Folding Time to Visualize Patterns of Temporal Evolution in Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, 2016.
- [9] W. Cui, X. Wang, S. Liu, N. H. Riche, T. M. Madhyastha, K. L. Ma, and B. Guo, "Let it flow: a static method for exploring dynamic graphs," in *IEEE Pacific Visualization Symposium*. Institute of Electrical and Electronics Engineers, 2014, pp. 121–128.
- [10] T. N. Dang, N. Pendar, and A. G. Forbes, "Timearcs: Visualizing fluctuations in dynamic networks," *Computer Graphics Forum*, vol. 35, no. 3, pp. 61–69, 2016.
- [11] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, "Dynamic network visualization with extended massive sequence views," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 8, pp. 1087–1099, 2014.
- [12] —, "Reducing snapshots to points: A visual analytics approach to dynamic network exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 1–10, 2016.
- [13] A. Dal Col, P. Valdivia, F. Petronetto, F. Dias, C. T. Silva, and L. G. Nonato, "Wavelet-based visual analysis for data exploration," *Computing in Science & Engineering*, vol. 19, no. 5, pp. 85–91, 2017.
- [14] —, "Wavelet-based visual analysis of dynamic networks," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 8, pp. 2456–2469, 2018.
- [15] —, "Wavelet-based visual data exploration," in *Vertex-Frequency Analysis of Graph Signals*. Springer, 2019, pp. 459–478.