

Wavelet-based Visual Analysis of Dynamic Networks

Alcebiades Dal Col, Paola Valdivia, Fabiano Petronetto, Fabio Dias,
 Claudio T. Silva, *Senior Member, IEEE* and L. Gustavo Nonato, *Member, IEEE*

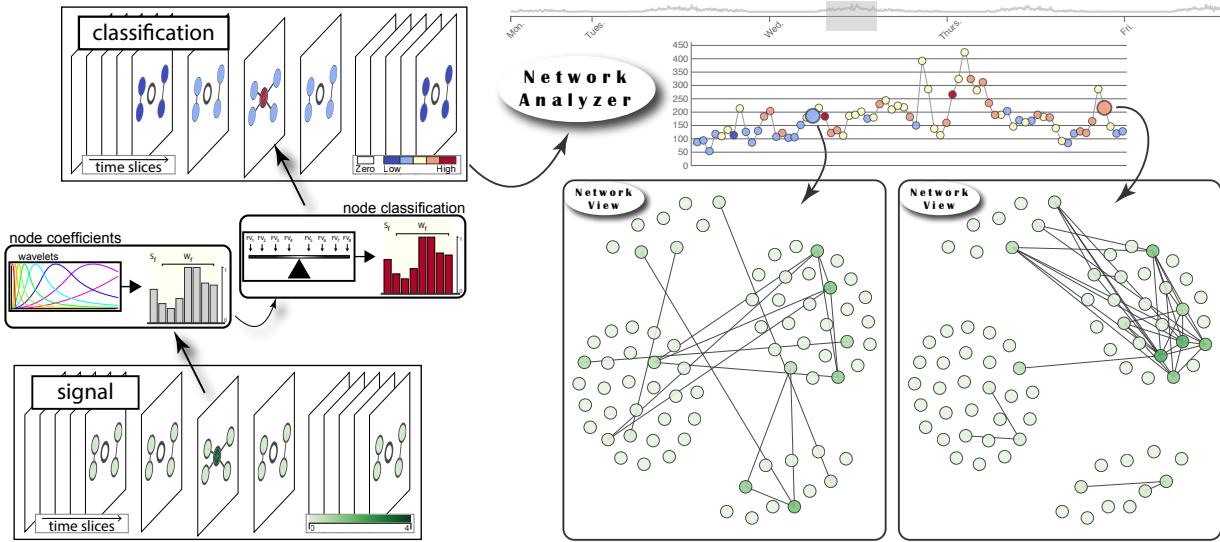


Fig. 1. Visual summarization of the temporal evolution of a dynamic network using spectral graph wavelet theory. Each node is classified using the wavelet coefficients from the decomposition of a signal associated to the nodes. Different levels of local change in the signal under analysis are revealed by the classification. The classification of all nodes of a time slice is used to classify the time slice itself, which are represented in the network analyzer. The position (height) of each circle corresponds to the level of activity in that time slice and the color to the predominant class. The user can gain insights about the evolution of the network without directly inspecting each time slice.

Abstract—

Dynamic networks naturally appear in a multitude of applications from different fields. Analyzing and exploring dynamic networks in order to understand and detect patterns and phenomena is challenging, fostering the development of new methodologies, particularly in the field of visual analytics. In this work, we propose a novel visual analytics methodology for dynamic networks, which relies on the spectral graph wavelet theory. We enable the automatic analysis of a signal defined on the nodes of the network, making viable the robust detection of network properties. Specifically, we use a fast approximation of a 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 coefficients naturally encode the spatial and temporal variations of the signal, leading to an efficient and meaningful representation. This methodology allows for the exploration of the structural evolution of the network and their patterns over time. The effectiveness of our approach is demonstrated using usage scenarios and comparisons involving real dynamic networks.

Index Terms—Dynamic networks, spectral graph wavelets, visual analytics.

1 INTRODUCTION

Networks naturally appear in scenarios where elements interact with each other. For instance, interpersonal contacts constitute a scenario

- A. Dal Col is with ICMC - USP, Brazil. E-mail: alcebiades_dalcol@usp.br.
- P. Valdivia is with ICMC - USP, Brazil and INRIA, France. E-mail: paolalv@icmc.usp.br.
- F. Petronetto is with Federal University of Espírito Santo, Brazil. E-mail: fabiano.carmo@ufes.br.
- F. Dias and C. Silva are with NYU, USA. E-mail: \{fabio.dias,csilva\}@nyu.edu.
- L. G. Nonato is with ICMC - USP, Brazil. E-mail: gnonato@icmc.usp.br.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
 For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
 Digital Object Identifier: <xx.xxxx/TVCG.201x.xxxxxxx>

where each person can be seen as an element and distinct elements are connected if they come into contact. Moreover, interpersonal contacts evolve with time, so the network is dynamic, changing its structure from time to time. Further, elements of the network can have properties associated to them, which can also change over time. The concomitant exploration of element properties and their dynamic relation in the network is a challenging problem whose complexity is usually heightened by the large number of elements to be processed.

Due to the importance of the theme, mathematical and computational tools for handling networks and their content have been progressing substantially. An important example is the recent extension of signal processing theory to networks [29, 31]. In particular, spectral graph wavelet theory [24] has gained attention due to its effectiveness in revealing patterns, removing noise, and compressing information. While graph wavelets have been considered in the literature, mostly for pattern recognition and machine learning, they were not fully explored as a framework for the visual analysis of dynamic networks.

In this work, we propose the use of graph wavelets to support vi-

sual analysis of dynamic networks, exploiting particularly the capability of graph wavelets to detect local changes in the data [36]. Moreover, we rely on an approximation mechanism to compute wavelet coefficients [24], reducing computational costs to enable the analysis of large dynamic networks. We show that the analysis derived from wavelet coefficients is able to reveal patterns of interactions, regardless of the involved elements and their temporal distance. By leveraging the wavelet coefficients, our method can summarize the temporal behavior of the network into a simple visualization, which also acts as a visual index, guiding users directly to relevant time slices for further exploration. In this aspect, we surpass the current state-of-the-art, representing more information and reducing the amount of necessary interactions to explore a dynamic network. The effectiveness of the proposed method is demonstrated, along with other important traits, in a set of usage scenarios and comparisons involving real dynamic networks.

The proposed methodology requires a signal properly defined on the nodes of the network. Two different forms of defining a signal so as to uncover interesting phenomena present in dynamic networks are discussed in Section 4 and employed in real usage scenarios in Section 5. Although the signal associated to nodes plays a crucial role in our methodology, we do not focus our presentation on specific signals. Rather, we devote most of our discussion to describe a general approach that supports any properly designed signal or measure defined on the nodes of a network.

In summary, the main contributions of this paper are:

- A wavelet-based methodology for dynamic network exploration that allows for the identification of both subtle and abrupt signal variation across the irregular topology of dynamic networks;
- An interactive visual analytics tool guided by our methodology and tailored to efficiently direct users to patterns and phenomena in dynamic networks, bridging the gap between exploring the network as a whole and specific elements;
- Two usage scenarios that demonstrate the capabilities of our methodology in revealing gist information for dynamic networks.

Our work is related to a previous paper [36], the main difference lies in the fact that it was only able to handle static graphs, while our methodology builds upon numerical mechanisms to make viable the processing of networks whose structure changes over time. Moreover, we propose a new visual analytics tool, and a torque-based classification mechanism, further we enable the visual identification of prominent and similar elements of networks.

The paper is organized as follows. Section 2 discusses related works. Section 3 provides theoretical background about spectral graph theory. Section 4 describes how we proceed to perform wavelet-based visual analysis of dynamic networks. In Section 5, we investigate two real-world datasets. In Section 6, we discuss the method, limitations, and future works. Finally, we conclude the article in Section 7.

2 RELATED WORK

In order to better contextualize our approach, we organize the following discussion as two separate parts, dynamic network visualization and graph wavelets data analysis.

Visualizing Dynamic Networks. The literature about dynamic network visualization is extensive [1, 5, 6, 23, 39], with the vast majority of approaches focusing on different manners of visually representing the network, mainly accounting for temporal variations. These efforts can be roughly grouped into animation and timeline-based techniques.

Animation techniques represent dynamic networks through a sequence of frames. Networks are usually represented with node-link diagrams [13, 15, 18], with a stable layout in order to preserve context. Tracking temporal changes is facilitated by transitions, such as highlighting differences on the network [3]. While changes in adjacent times are evident, animation-based methods demand greater cognitive effort to track far apart changes in the network, mainly when many time slices are involved.

Timeline-based techniques are static representations that map time into space, illustrating the evolution of a network as an image. Some

methods are able to change the information associated to each node in each time [8, 12, 37], visually representing the information through glyphs [20, 30] or matrices [34, 40, 42]. Small multiples depicting different temporal moments have also been used as visual resource [16, 21, 30]. Visual representation of interactions between entities [12] is also possible with timeline-based methods. MultiPile [2] allows access to subtle changes in specific periods of time by piling adjacency matrices that represent snapshots of the network. These piles are able to capture irregular periods of time, which can be used to detect changes in the network. However, scalability is an issue for timeline-based techniques. Alternatives to mitigate the problem have been proposed, as for example the aggregation of elements [22], but at the expense of hiding relevant information.

Recent efforts try to reduce the shortcomings of both representations by considering more information from the network. Cui et al. [11] rely on static flow visualization to show changes in particular properties of the nodes, e. g. degree and centrality. Their method is able to reveal interesting patterns in the data, however, finding repetitive patterns is not easy and the relations between nodes are also not properly exploited.

Projections have been used to convey information about dynamic network structures [4, 9, 33, 38, 41], considering mainly Multidimensional Scaling. Those methods can be used to project feature vectors that characterize network structures, specific periods of time, or other information of interest, and the result can be used to identify clusters and anomalies. The performance and effectiveness of those methods are dependent on the definition of the feature vectors.

Our methodology allows the automatic detection of local variation, establishing a relationship between each element and its contact partners. In contrast, animation and timeline-based techniques that enable similar investigation require the user to mentally compare objects and their neighbors, tending to require more cognitive effort.

Graph Wavelets. With the extension of signal processing to graphs [29, 31], spectral approaches have been considered for the analysis of networks [28], including subgraph detection [26] and point cloud compression [43]. In particular, graph wavelets [24] are receiving special attention, because they enable the interpretation of spectral properties as in Fourier analysis, while making viable a local analysis of the network. Applications include community mining [35], network monitoring [10], traffic event detection [27], and geometry processing [14].

Hamon et al. [25] derive a signal from the graph, which is processed and converted back into graph form, aiming to detect network structures. While this method avoids the mathematical differences of operating directly in the graph, the performance is dependent on the order of the nodes.

In the context of visual analysis of dynamic graphs, graph wavelets are particularly useful to detect local changes in the network, which are reflected in the wavelet coefficients. This fact can be used to direct users to patterns and changes in the behavior of network elements. Valdivia et al. [36] consider the coefficients of the graph wavelets to enable the visual analysis of urban data, allowing the identification of patterns and outliers from a limited amount of information. However, that formulation is only able to handle static graphs where information associated to nodes varies over time. Tremblay and Borgnat [35] use graph wavelets for multiscale detection of communities, that is, groups of nodes well connected together. When dealing with dynamic networks, the data is aggregated in a preprocessing step, not considering the temporal evolution of information. Additionally, the grouping is performed on each wavelet scale individually, while our method builds feature vectors that encompass all scales simultaneously.

Our method exploits the interesting properties of graph wavelets and relies on a mechanism with a low computational cost, thus allowing the analysis of large dynamic networks. While other methods are capable of highlighting specific changes in the network, the sensitivity of our methodology to changes is unmatched. Indeed, it has been demonstrated [36] that the analysis derived from wavelet coefficients can reveal both network wide changes and subtle local changes, a trait not typically present in dynamic network visual analytics methods.

3 SPECTRAL GRAPH THEORY

Our method relies on graph wavelets to explore data represented as a dynamic network, where the network topology can change over time. The exploration is guided by a signal associated to nodes of the network, crafted to highlight specific features for the considered application. As further discussed in the following sections, the signal associated to network nodes can be obtained in several ways, from attributes present in the nodes to values computed from the network topology, providing great flexibility to the method.

The graph wavelet theory is derived from the graph Fourier transform, which uses spectral graph theory to define Fourier modes and frequencies on graphs. By analyzing the wavelet coefficients in specific frequencies/scales, our method is able to characterize the behavior of the network over time. Before presenting details of our method, we briefly review the mathematical foundations of graph signal processing [24, 28]. Readers not interested in the mathematical foundations of graph wavelets can skip Subsections 3.1, 3.2, and 3.3, going directly to Subsection 3.4, where we provide a high-level interpretation of wavelet coefficients and the intuition behind our approach.

3.1 Graph Fourier Transform

Let $G = (V, E)$ be an undirected weighted graph, where V is a set of n nodes and E a set of weighted edges connecting pairs of distinct nodes in V , with non-negative edge weights. The adjacency matrix $A = (a_{ij})$ associated to G is a $n \times n$ matrix with entries $a_{ij} = a_{ji} = w_{ij}$ if there is an edge with weight w_{ij} in E connecting the nodes τ_i, τ_j , $i \neq j$ (we do not consider self-loops), and $a_{ii} = 0$ otherwise. The graph Laplacian is given by $L = D - A$, where $D = (d_{ij})$ is a diagonal matrix with entries $d_{ii} = \sum_k w_{ik}$ (sum of the weights of all edges incident to node τ_i). By construction, the matrix L is real, symmetric, and semi-positive definite, thus it has a complete set of orthonormal eigenvectors $\{u_i\}$, $i \in \{1, \dots, n\}$, with corresponding non-negative real eigenvalues $\{\lambda_i\}$, $i \in \{1, \dots, n\}$, which we assume ordered in non-decreasing order $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$. An interesting property of the graph Laplacian is that the eigenvectors u_i and eigenvalues λ_i play the same role as Fourier modes and frequencies in the continuous case. Small values of λ_i correspond to low frequency Fourier mode u_i while large values indicate high frequency Fourier modes (see [7, 28] for details).

Given a real-valued function $f : V \rightarrow \mathbb{R}$ on the nodes of G , the signal under analysis, the graph Fourier transform of f at the frequency λ_ℓ is defined as:

$$\hat{f}(\ell) = u_\ell^\top \cdot f = \sum_{j=1}^n u_\ell(j) f(j) \quad (1)$$

where u_ℓ^\top is the transpose of u_ℓ , and $u_\ell(j)$ and $f(j)$ are the values of u_ℓ and f in the node τ_j of G . Therefore, the ℓ -th graph Fourier coefficient of f is given by the dot product between f and u_ℓ .

3.2 Spectral Graph Wavelet Transform

The Fourier analysis provided by the spectrum of the graph Laplacian is the basis for the graph wavelet theory proposed by Hammond et al. [24], which we follow in this work. The wavelet coefficients at scale s of a given function f are obtained by modulating each Fourier mode. In mathematical terms, the wavelet coefficient of f in a node τ_j at scale s is given by:

$$W_f(s, j) = \sum_{\ell=1}^n g(s\lambda_\ell) \hat{f}(\ell) u_\ell(j) \quad (2)$$

where g is a filter kernel defined on \mathbb{R}^+ . Among some alternatives, we adopt:

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

Small values of s accentuate high frequency Fourier modes, ensuring good localization, while large values of s compress the function $g(sx)$ around low frequency Fourier modes to encode coarser

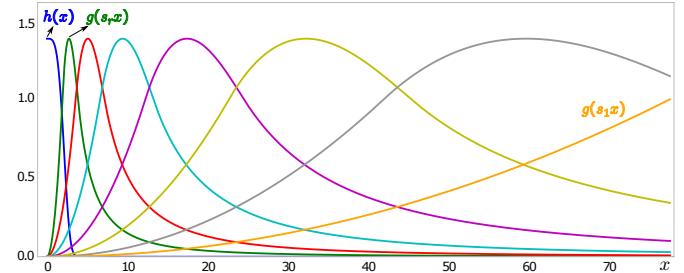


Fig. 2. Scaling function (blue), and seven wavelet kernels.

description of a local neighborhood. In practice, the scales used to generate the wavelet kernels $g(sx)$ are logarithmically sampled between $s_{min} = s_1, s_2, \dots, s_r = s_{max}$, where r is the number of scales, $s_{min} = 2/\lambda_n$, and $s_{max} = 40/\lambda_n$.

Scaling function. Hammond et al. [24] introduce a scaling function to stably represent low frequency Fourier modes. These waveforms are constructed by a real-valued function $h(x) = \gamma \exp(-(\frac{10x}{0.3\lambda_n})^4)$ that acts as a low-pass filter. The parameter γ is chosen such that $h(0)$ is equal to the maximum value of g . The scaling function coefficients are given by:

$$S_f(j) = \sum_{\ell=1}^n h(\lambda_\ell) \hat{f}(\ell) u_\ell(j). \quad (4)$$

In this configuration, the spectral graph wavelet transform produces a scaling function coefficient $S_f(j)$ and a sequence of wavelet coefficients $W_f(s_1, j), \dots, W_f(s_r, j)$ for each node $\tau_j \in V$. The interpretation of the scaling function coefficient is identical to the other wavelet coefficients, but representing the lowest wavelet frequency. For simplicity, we refer to this combination of the scaling function coefficient and the wavelet coefficients as “wavelet coefficients”, which are assumed ordered from lowest to highest wavelet frequency, $\{S_f(j), W_f(s_r, j), \dots, W_f(s_1, j)\}$.

In our implementation, we considered the scaling function and seven wavelet kernels (Figure 2).

3.3 Fast Spectral Graph Wavelet Transform

Calculating the wavelet coefficients from Equation (2) can become unfeasible when dealing with large graphs, as the whole set of eigenvalues and eigenvectors has to be computed, a task with cost $O(|V|^3)$. However, the coefficients can be approximated using the Chebyshev polynomials via the recursive formula [24]:

$$T_k(y) = 2yT_{k-1}(y) - T_{k-2}(y) \quad (5)$$

with $T_0 = 1$ and $T_1 = y$, for all $y \in [-1, 1]$. These polynomials can be shifted to adjust their domain to the interval $[0, \lambda_n]$ by simply making $\bar{T}_k(x) = T_k(\frac{x-a}{a})$ with $a = \lambda_n/2$. The wavelet coefficient in a node τ_j at scale s_i can be approximated as:

$$W_f(s_i, j) \approx \left(\frac{1}{2} c_{i,0} f + \sum_{k=1}^m c_{i,k} \bar{T}_k \right) (j) \quad (6)$$

$$c_{i,k} = \frac{2}{\pi} \int_0^\pi \cos(k\theta) g(as_i(\cos(\theta) + 1)) d\theta \quad (7)$$

where m is the truncation parameter, g is the filter kernel as defined in Equation (3), and \bar{T}_k is an n -array that represents $\bar{T}_k(L)f$. The vector \bar{T}_k is given by the shifted Chebyshev formula:

$$\bar{T}_k = \frac{2}{a} (L - aI) \bar{T}_{k-1} - \bar{T}_{k-2}, \text{ for } 2 \leq k \leq m \quad (8)$$

with $\bar{T}_0 = f$, $\bar{T}_1 = \frac{1}{a}(L - aI)f$, and I as the identity matrix.

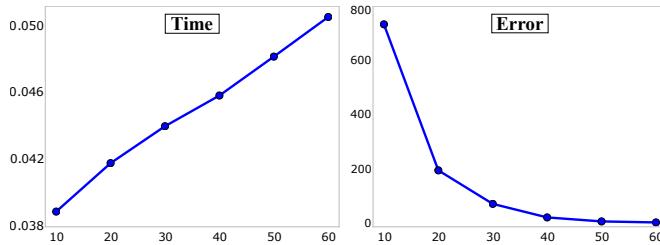


Fig. 3. Computational cost and error in the approximation of the wavelet coefficients as a function of m for a synthetic network. The left graph shows processing times (seconds), while the right graph shows the sum of absolute differences between exact and approximated wavelet coefficients.

Computational Cost. Equation (6) shows that the wavelet coefficient in a node τ_j at scale s_i can be obtained by computing only the largest eigenvalue λ_n of the graph Laplacian L . Moreover, the computation of $(L\bar{T}_{k-1})(j)$ on the right side of Equation (8), the j -th entry in the vector $L\bar{T}_{k-1}$, involves only the neighbors of node τ_j in G , avoiding the explicit construction of the graph Laplacian. Therefore, the computational cost to approximate the wavelet coefficients at a given scale is $O(m|E| + mn)$, where $|E|$ is the number of edges in G .

The truncation parameter m dictates the quality of the approximation, and should be tuned for a good trade-off between accuracy and computational cost. Figure 3 shows the processing time and error in the approximation of the wavelet coefficients for a synthetic network (see Subsection 4.4), using an i7 processor. As one can see, $m = 40$ is a good trade off. The same trade-off result was observed in other tests we carried out, leading us to assume this parameter value in all of our experiments. Shuman et al. [32] suggested that m equal to 20 results in a close approximation. However, with a slight increase in processing time, it is possible to increase m significantly reducing the error.

3.4 Interpretation of the Wavelet Coefficients

The wavelet coefficients can be interpreted as the energy of the original signal in specific frequencies. As in the classic wavelet transform (continuous case), high frequency regions correspond to neighborhoods where the value of the signal is considerably different with respect to neighboring points, while low frequency regions contain smoother signal differences among neighboring points. The same is true in the context of networks, where neighborhoods are defined from the graph structure.

The graph wavelet transform associates a set of coefficients to each node of a graph, where each coefficient represents the amount of energy in a specific frequency. As illustrated in Figure 4, nodes where the value of the signal is in unison with neighboring values (light green nodes in Figure 4) are low frequency nodes, thus their corresponding low frequency wavelet coefficients tend to be more pronounced. In contrast, nodes containing signal values that differ considerably from neighboring values correspond to high frequency nodes, being characterized by pronounced high frequency coefficients (middle dark green node in Figure 4). A more significant signal difference leads to more energy on higher frequencies. Therefore, considering only the wavelet

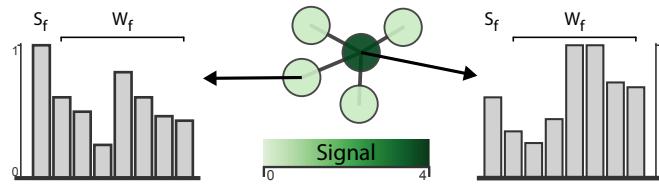


Fig. 4. An example with high and low frequency nodes. The wavelet coefficients allow to infer the signal behavior on and around each node.

coefficients, one can infer how similar the signal on a node and on its neighbors is. Indeed, the wavelet coefficients are perfectly suitable to characterize the behavior of the signal on and around each node.

While the magnitude of the signal is also represented in the wavelet coefficients, its effect can be reduced if the coefficients are normalized, which allows for the comparison of similar signal configurations of different magnitudes.

We used a simple example, depicted in Figure 4, to provide some intuition about wavelet coefficients. It is evident that the graph wavelet transform takes into account a neighborhood that reaches beyond the direct neighbors. For instance, the value of the signal in the light green nodes is in unison with second-order neighbors and the wavelet coefficients gracefully react to this phenomenon.

4 VISUAL ANALYSIS OF DYNAMIC NETWORKS VIA GRAPH WAVELETS

The proposed visual analysis method involves to tackle two main issues: 1) properly define a signal on the nodes of the network and 2) convert the result of the wavelet analysis in a (set of) time series representations whose behavior reflects phenomena present in the network. There are nuances involved in the solution of these steps, which we detail in this section. Precisely, the problem of properly defining a signal on the nodes of the network is discussed in Subsection 4.2 and the problem of converting the output of the wavelet analysis in an enriched time series like representation is detailed in Subsections 4.3 and 4.4.

In the graph wavelet formulation presented in Section 3, the topology of the network does not change and the signal associated with the nodes is a scalar. To adapt that framework to the context of dynamic networks we need to devise a suitable representation, as described in Subsection 4.1.

4.1 Dynamic Network Model

We model a dynamic network as a sequence of N undirected weighted graphs G_1, G_2, \dots, G_N , where, for each time slice $k \in Z = \{1, 2, \dots, N\}$, $G_k = (V \times \{k\}, E_k)$ is a graph with nodes $(\tau_i, k) \in V \times \{k\}$ and edge set $E_k \subseteq (V \times \{k\}) \times (V \times \{k\})$ with weight w_{ij}^k associated to each edge in E_k connecting nodes (τ_i, k) and (τ_j, k) . In order to apply the graph wavelet theory, as described in Section 3, we add (temporal) edges between corresponding nodes in adjacent time slices (blue lines in Figure 5), resulting in a network $G = (V \times Z, E)$ with vertex 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 (τ_i, k) and $(\tau_i, k+1)$ in consecutive time slices, for $i = 1, \dots, n$, $k = 1, \dots, N-1$, with n and N corresponding to the number of nodes in V and the number of time slices, respectively. Since consecutive time slices are connected by temporal edges in G , the graph wavelet theory can be applied to analyze functions defined on the nodes of G , allowing the detection of spatial and temporal variations as well as the identification of important phenomena in the network as discussed in Section 5. Importantly, we use equidistant time-windows to aggregate data in time slices. For instance, in one of our usage scenarios in Subsection 5.1, each time slice corresponds to six minutes of interpersonal contact information.

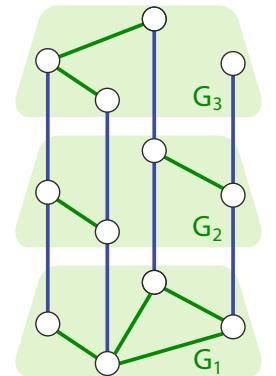


Fig. 5. Dynamic network.

In a similar model, Valdivia et al. [36] incorporate temporal variation of node attributes in the analysis of static networks. Copies of the same graph G are stacked such that each copy holds the node attributes at a time slice. Our dynamic network model, in turn, associates a graph G_k with each time slice $k \in Z$, allowing variation in both node attributes and network structure.

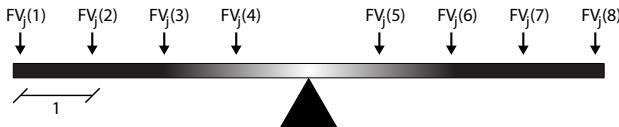


Fig. 6. Torque analogy for node classification. Lowest frequency at the leftmost end and highest frequency at the rightmost end.

4.2 Node Signal and Edge Weights

Signal values associated to nodes of a network play a crucial role in our methodology. Another factor that impacts in our analysis is the weight of the edges, since it 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 network. The information of the edges is encoded in the graph Laplacian.

The signal to be associated to the nodes can be naturally present in the data, as an intrinsic characteristic of nodes. For instance, one of our usage scenarios in Subsection 5.2 explores taxi trips in New York City, where regions of the city correspond to nodes in the network and the signal associated to the nodes is obtained from the number of taxi pick-ups within regions. Edges are placed when the pick-up and drop-off locations take place in different regions/nodes, making the weight of each edge equal to the maximum of the number of trips in each direction. This approach allows the analysis of the variation of node attributes across network structure.

As f is a real-valued function, any signal can be used. To explore contact networks in Subsection 5.1, we correspond people to nodes, with associated signal defined as 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.

In summary, if the signal takes into account intrinsic characteristics of nodes, one can understand how such property is distributed across the network structure either spatially or temporally; if the signal is computed from network topology, one can explore the dynamics of the network.

Beyond the edges connecting nodes representing different entities, our dynamic network model also includes edges between nodes that represent the same entity in sequential time slices, the temporal edges. The weight of these edges can also be handled to control the effect of temporal variation on the resulting coefficients.

4.3 Node Classification

Each node of G is classified using a feature vector containing the corresponding coefficients ordered from lowest to highest wavelet frequency, that is, $FV_j = (S_f(j), W_f(s_r, j), \dots, W_f(s_1, j))$. 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.

To reduce the impact of noise, isolated nodes in each time slice are assigned to a specific class, denoted as *Zero* class. The other nodes are then separated into five classes using a simple analogy to mechanical torque. Consider a horizontal bar fixed at the middle but allowed to rotate freely. Each coefficient of the feature vector FV_j corresponds to a force on a specific point of the bar, as illustrated in Figure 6, considering eight coefficients. Forces acting towards the ends of the bar have greater impact on the torque than those close to the fixed point. The resulting torque indicates whether forces tend to rotate the bar to the left or to the right, that is, whether the feature vector FV_j has predominantly low or high frequency.

The torque is then used to classify each node, the values are normalized into the interval $[-1, 1]$ by simply dividing by the maximum

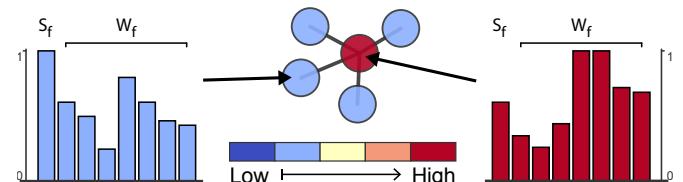


Fig. 7. Node classification performed on the example from Figure 4. The classification represents in a simple manner the information contained in the wavelet coefficients.

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), see Figure 7. 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. A better discrimination of the intervals $[-1, -0.05]$ and $[0.05, 1]$ can be reached considering more classes, however five classes have proved satisfactory results. Moreover, more classes can generate instabilities in the classification, since similar nodes can be classified into different classes due to the narrow intervals.

The classification leads to a straightforward interpretation, considerably easier than interpreting the coefficients directly, where a red node indicates an abrupt change in the signal, considering both spatial and temporal neighborhood. Figure 7 illustrates this classification scheme applied to the example from Figure 4. While there is some energy on the lower frequencies on the feature vector of the middle node, the coefficients are predominantly high frequency, leading to a classification into the red class. For the other nodes, the energy on the lowest frequency (the scaling function coefficient) is partially counterbalanced by the energy on the higher frequencies, resulting in a classification into the light blue class.

4.4 From Wavelet Coefficients to Visualization

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

Goal 1. The analysis of the network as a whole. By combining the classification of each node in each time slice, our method can characterize the behavior and temporal evolution of the whole network.

Goal 2. The analysis of each node. The graph wavelet transform reliably identifies local variations in the network. Nodes are classified according to the abruptness of the signal variation.

Goal 3. The identification of prominent nodes. The wavelet coefficients are used to classify the nodes according to signal variations. Nodes classified as high frequency correspond to nodes where the associated signal faces abrupt variations, thus indicating the node behaves discrepantly from its neighbors. The visualization should efficiently guide users to these nodes spatially and temporally.

Goal 4. The identification of similar nodes. Wavelet analysis enables the identification of nodes with similar behavior over time, regardless of their spatial and temporal location.

The proposed visualization is composed of four linked views: network analyzer, node ranking, node time series, and time slice view, as illustrated in Figure 8.

Network analyzer. This panel depicts 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 value 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

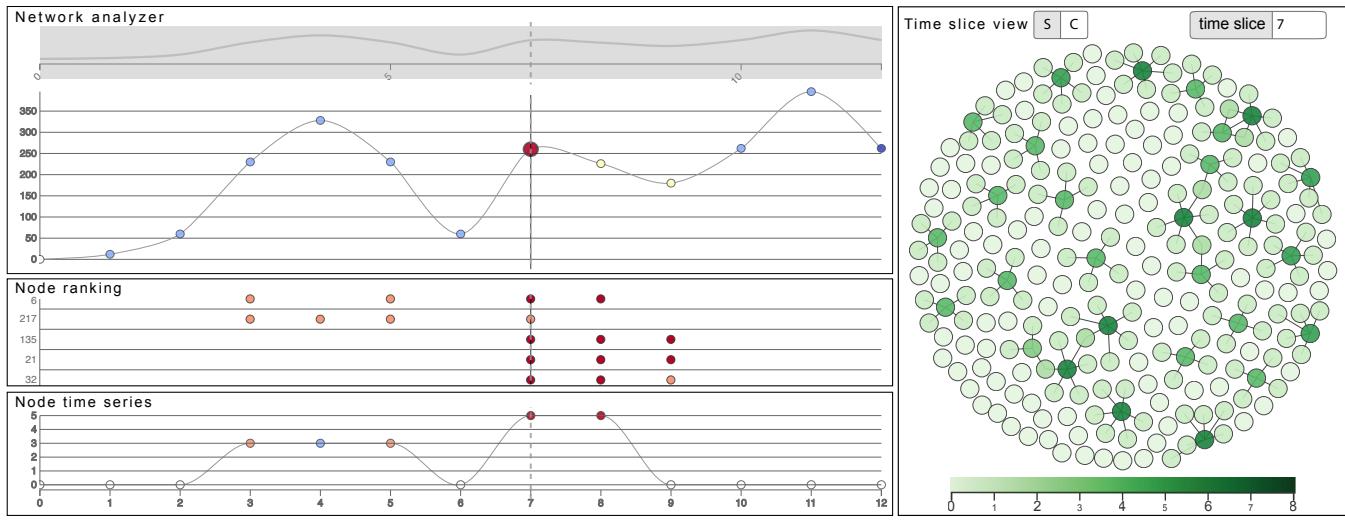


Fig. 8. The proposed visualization includes four linked components: network analyzer, node ranking, node time series, and time slice view.

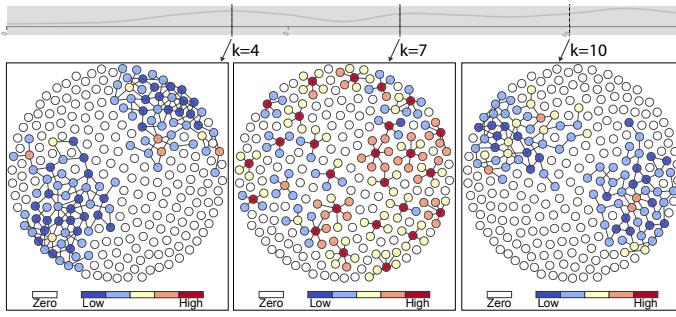


Fig. 9. Node classification for time slices 4, 7, and 10. All these time slices have similar heights on the network analyzer. However, time slices 4 and 10 are classified as light blue, while time slice 7 is classified as red.

network, directing the user to pattern changes. For instance, if in a given time slice the red class has one single node and the light blue class has eight, and the maximum number of nodes in these classes, at all times, is one and ten, respectively, then the circle of this time slice is colored red, to indicate the larger relative presence of red nodes.

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 [19]. 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.

To properly demonstrate the capabilities of our method, we created a synthetic network containing 250 entities and 13 time slices. The signal associated with a node is the number of edges incident to the node in its time slice, and the weights of the edges are all equal to one. There are two large spatial events (leftmost panel in Figure 9) 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 9) appear, with some of them disappearing on the next time slices. At time slice 10, another two large spatial events (rightmost panel in Figure 9) are created by grouping and expanding the small events that still exist, reaching an activity peak at time slice 11 and decreasing at time slice 12.

The network analyzer for the synthetic network is illustrated on the top left of Figure 8, showing that the activity (number of edges) of the network increases and decreases, with three peaks, at time slices

4, 7, and 11. Most time slices are classified as light blue, which indicates smooth changes caused by small variations and/or large events. This is indeed the case as illustrated in the leftmost panel of Figure 9, which depicts the time slice 4. The time slice 10, illustrated in the rightmost panel of Figure 9, has height (activity level) similar to time slice 7 in the network analyzer, but those two time slices differ in their classification.

In fact, time slice 7 is classified as a red time slice. This change in the classification indicates a difference in the current behavior of the network, albeit with almost the same level of activity. The color indicates an increased concentration of nodes classified as high frequency, therefore this time slice contains significant differences in the signal and/or small events. As illustrated in the middle panel of Figure 9, this is indeed the case, corresponding to several small/high frequency events that suddenly appeared at time slice 7. Since the next two time slices are not classified as red, this sudden change became somewhat static, part of the new pattern, leading to a slightly lower frequency classification due to the temporal stability.

By considering the position (height) and the color of the circles, we can infer consistent states of the network, visually identifying instants with similar configuration and signal distribution. The capability of identifying time slices with similar network activity level while discriminating patterns of behavior is one of the traits that differentiate our approach from others in the literature, as for example the method proposed by van den Elzen et al. [38], called Reducing Snapshots to Points (RSP), where similar states are also visually grouped, but not discriminated. As we shall see in more detail in Section 5, our method not only visually identifies similar states, but also provides additional information about it. Indeed, users gain insights about the general behavior of the network at each time slice without directly inspecting it.

Figure 10(a) illustrates the RSP method applied to the synthetic network. The circles corresponding to time slices 4 and 10 are projected distant from each other, indicating different network configurations. From the viewpoint of our method, these times are similar, almost the same activity level and two large events. The RSP method differentiates them because it takes into account which nodes are involved. At a glance, our network analyzer reveals all these information, while the RSP method require additional views to bring meaning to the difference visually asserted. In addition, we use another state-of-the-art technique, called TimeArcs [12], to explore the synthetic network. The TimeArcs method also reveals different nodes involved in two major events in time slices 4 and 10, as shown in Figure 10(b). However, the activity pattern presented in the middle panel of Figure 9, namely small events with central elements, is not easily perceived, while our tool makes it clear in advance thanks to the network analyzer.

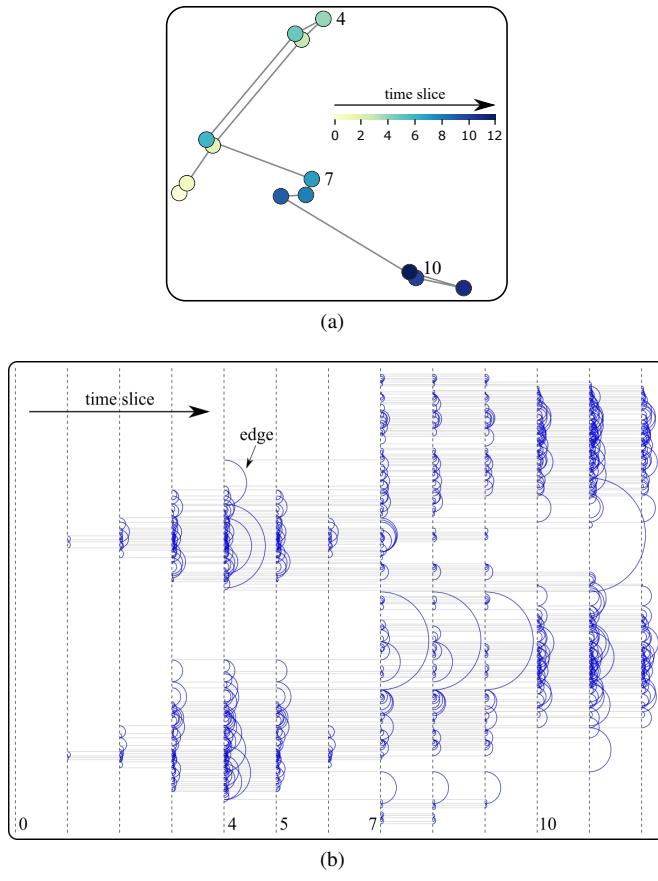


Fig. 10. (a) RSP and (b) TimeArcs methods applied to the synthetic network.

Node ranking. This panel shows the nodes ordered according to their relevance. Since relevance is application dependent, we adopted a simple mechanism to rank nodes, in which 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 order the nodes according to the total number of time slices in which it is considered relevant, on a given time frame. Considering the whole synthetic network, as illustrated on the middle left of Figure 8, the most relevant entity corresponds to the node identified as the number six, that was classified as red/orange in four time slices. Indeed, this node was involved in the large events of the first time slices, near the border of the event, which lead to a higher frequency, and was the center of a small event later.

By clicking on a node label 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 displayed relevance vectors. Considering the synthetic network, this option would quickly reveal nodes similar to node six and therefore nodes that were the central nodes of small events.

Since this definition of similarity is the number of time slices where two entities are classified as relevant, which is not related to the activity curves of the entities, the results may seem counterintuitive. This measure identifies entities that had significant change at the same times, regardless of the overall behavior or format of the activity curves.

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. Further, different definitions can

be used for the relevance and similarity of nodes, increasing the potential use cases of the method.

Node time series. Once an entity is selected in one of the panels, the corresponding time series is depicted. The height of the dots represents the signal value and the colors indicate the node classification in each time slice, as illustrated on the bottom left of Figure 8.

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 the 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.

5 USAGE SCENARIOS

To demonstrate the capabilities of the proposed method, we consider two distinct real dynamic networks. The first dataset contains human interaction information over time, while the second contains georeferenced information about taxi trips in New York City. The prototype visualization interface was implemented in JavaScript. A Python script converts the information from its original format into the dynamic network model presented in Subsection 4.1, calculates the signal and edge weights, performs the approximation of the wavelet coefficients and node classification, and saves the results in a format suitable for the visualization interface.

5.1 High School Dynamic Contact Network

The high school dataset contains face-to-face contact information between 180 students from a school in France [17] (<http://www.sociopatterns.org>), during nine days in November of 2012. 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 approximation of the coefficients took approximately 7 seconds, and the whole preprocessing step took approximately 20 seconds on a regular i7 computer, using around 1Gb of memory.

We start the exploration using the network analyzer, which shows the total number of contacts made by students (Figure 11). As expected, there are seven periods of activity, corresponding to the seven weekdays where the students interacted. Interestingly, 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 [17, 38].

The predominance of light blue nodes indicates that the vast majority of time slices belongs to the medium-low frequency class (Goal 1), where changes in the number of contacts between children are smooth, but not enough to be classified as dark blue. Indeed, this dataset is naturally biased to low frequencies, as no node was classified as red. However, there are time slices classified as orange, mostly on Monday, suggesting sudden changes in the number of contacts. The network analyzer allows for zooming particular time-intervals, in particular the zoomed-in view at the bottom in Figure 11 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 12.

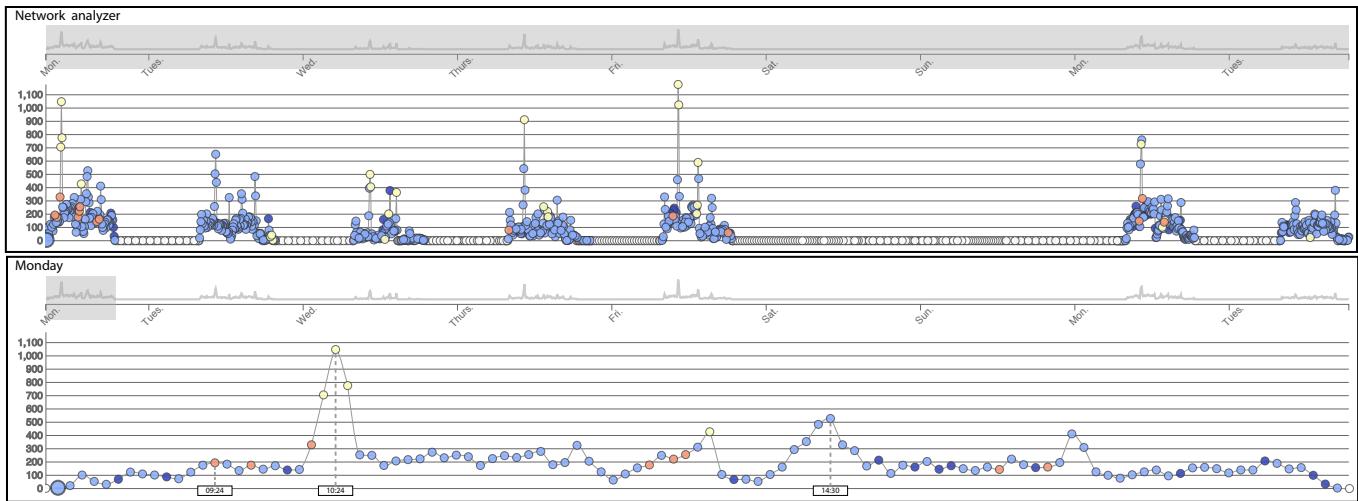


Fig. 11. Network analyzer for the high school dataset, showing the whole dataset at the top and only Monday (zoomed-in view) at the bottom.

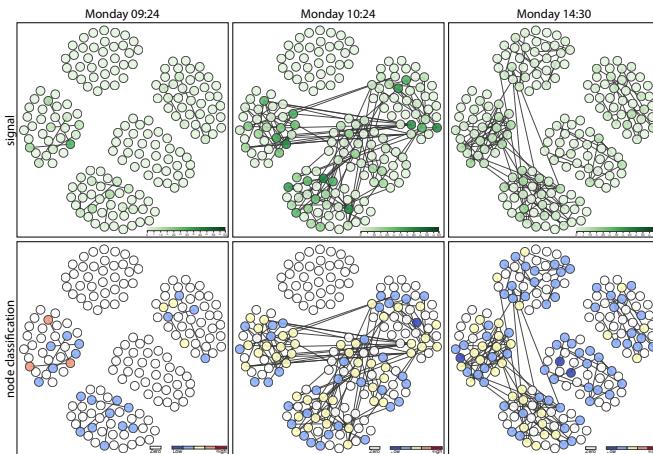


Fig. 12. Signal and node classification for Monday at 9:24, 10:24, and 14:30 time slices.

The 9:24 time slice is the first classified as orange on this dataset. From the network analyzer only, one can infer that this time slice does not contain much activity, its height is well below the peaks for this day; it is part of a consistent level of activity that covers the morning period. While most of its neighboring time slices are classified as light or dark blue, it was classified as orange. This classification implies a difference in the contact pattern, but not abrupt enough to be considered red. Since the dataset is predominantly low frequency, the presence of a few higher frequency nodes would be sufficient to lead to a higher frequency classification.

By inspecting the 9:24 time slice, as illustrated in Figure 12 (leftmost column), it is clear that this time slice mostly contains pairs of contacts, with low signal (number of contacts). 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 other words, this activity peak is different from the pattern of behavior of the network at this time. This change is correctly detected and represented by the network analyzer, reducing the amount of direct inspection and comparisons with respect to other methods in the literature for dynamic network analytics.

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 moderate signal differences. Indeed, more pronounced signal differences and/or smaller events would cause a number of the nodes to be classified into a higher frequency class, which would classify the time slice into a higher frequency class as well. Similarly, a large event without somewhat significant signal changes would be predominantly light or dark blue. In other words, this time slice contains nodes with moderate signal connected to several nodes with lower signal, leading to a classification as yellow. As illustrated in Figure 12 (middle column), this is clearly the case. This time slice includes interactions between four of the five classes of students.

The 14:30 time slice also contains a considerable amount of activity, indicated by the height in the network analyzer, however, the time slice is classified as light blue. The combination of increased height and light blue classification means this time slice contains a large event, without signal peaks. Smaller events or peaks would lead to higher frequencies, as was the case for the 10:24 time slice. Indeed, as illustrated in Figure 12 (rightmost column), the signal varies smoothly across a large event, with some nodes classified as yellow, but with the relative majority of light blue nodes.

These results highlight the potential of our network analyzer, illustrating that it represents relevant information concerning the signal under analysis and the topology of the associated dynamic network, 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 network.

The main idea behind our network analyzer - to present a simple summary of the temporal evolution of the network to guide the user through the exploration - is not new. In fact, our work was inspired by Reducing Snapshots to Points (RSP) [38], but the use of wavelet theory allows our method to leverage more information efficiently. This dataset was chosen exactly because it was used as a case study in RSP paper. We perform a direct comparison, illustrated in Figure 13, by implementing RSP following all guidelines given by the authors in the original paper.

Importantly, the RSP method considered a different aggregation of edges, using 6 minutes intervals with an overlap window of 60 minutes, centered at the considered time, while our method uses 6 minutes intervals. Since this aggregation can be seen as a low pass filter, running our method with overlapping windows results in forcibly lower frequencies. Conversely, if RSP is applied to data without an overlap window, the paths corresponding to each day do not appear, because each time slice is considerably different from the previous. Therefore, Figure 13 shows the best results of both methods. We consider that the smaller temporal aggregation is beneficial, since more details can be

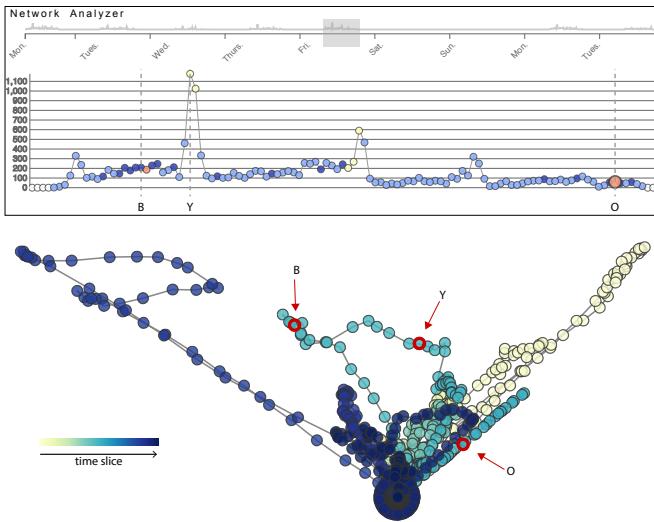


Fig. 13. Comparison with a state-of-the-art technique considering the high school dataset. Our network analyzer, focusing on Friday (top). Projection view of the RSP method (bottom); circles are colored to encode the time, darker markers correspond to older snapshots. The *B*, *Y*, and *O* letters create a time match in the views.

revealed, thus better depicting the evolution of the network.

From Figure 13 one can see that, in our method, both the position and color of each point have a clear meaning, while for RSP, the color represents time and the position is derived from dimensionality reduction techniques, which cannot be as easily interpreted, especially when using non-linear methods. While there are other possible layouts for RSP, the only insight available for the user is that there are consistent states, where time slices are grouped at an arbitrary position. Our method represents these states and provides information regarding the activity and general behavior of the network at each time slice.

Specifically, consider the three time slices indicated in Figure 13. While RSP's layout correctly separated the three points, it is not clear how these time slices are different or even the temporal relationship between them. Our method naturally places these points along the time axis, representing the activity level and the information about the time slice. Contrarily to RSP, using our network analyzer, the user can obtain valuable information, without inspecting the corresponding time slice. For instance, the level of activity at time *Y* is higher than at times *B* and *O*, time *O* contains nodes with moderate signal variation, and so on.

Moreover, while RSP is used to explore the topology of the network, our method can be used to simultaneously explore the topology and an associated signal. Indeed, the same network can yield different results, by changing the associated signal. This enables the exploration of different facets of the data, which is not easily done in other methods from the literature.

To further relate our method with the state-of-the-art, we apply the TimeArcs [12] method to the high school dataset by using the source scripts made available by the authors. Figure 14 depicts the students as horizontal lines and their interpersonal contacts as blue arcs. Although the method vertically group nodes according to similarity, there is a large amount of edge cluttering. Further, long arcs interfere with the visualization of other time slices, they come from student pairs that have low interpersonal contact along all time frame.

5.2 Manhattan Taxi Pick-Ups

This dataset contains taxi trips in Manhattan in December 2009, created from NYC Taxi and Limousine Commission (TLC) Trip Record Data (http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml). Each taxi zone, as defined by TLC, is represented as a node, and the associated signal corresponds to the number of taxi pick-ups on that zone. We restricted

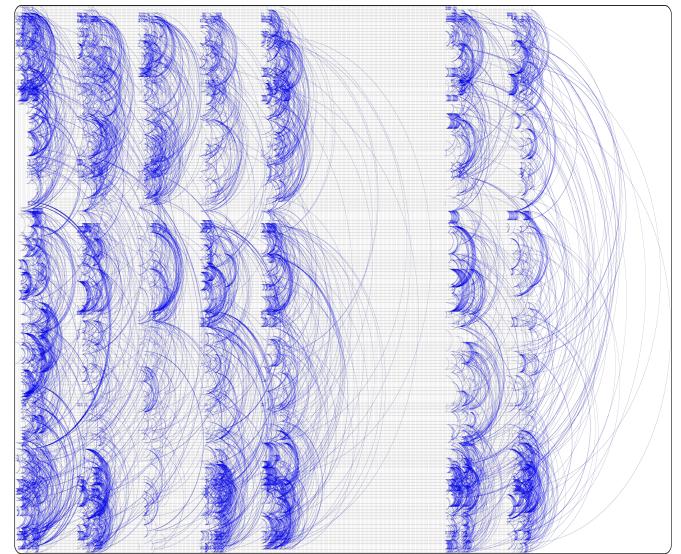


Fig. 14. TimeArcs method applied to the high school dataset. The temporal evolution is represented along the *x* axis.

our attention to the zones in Manhattan and the three nearby airports. Edges are placed between two zones if there are trips between them, and the weight of each edge is the maximum of the number of trips in each direction between the two involved zones. Each time slice corresponds to 10 minutes, for a total of 4,464 time slices. The resulting dynamic network has 5,078,722 edges, not counting the temporal edges. The preprocessing time was around 40 seconds, with the approximation of coefficients taking about 5 seconds, using about 3.7Gb of memory.

Figure 15 illustrates the proposed visual analytics interface on this dataset. The network analyzer shows that the nodes have predominantly light blue classification, indicating smooth taxi pick-ups distributions. However, there are several time slices with red classification, in a periodic pattern, which means a significant difference in the number of taxi pick-ups from neighboring time slices and zones (Goal 1). By selecting the first one, corresponding to December 5th at 02:00, the time slice view changes, showing two nodes with red classification, corresponding to Lower East Side and East Village (Goal 2).

By selecting these zones, we can inspect the corresponding time series, illustrated in Figure 16, which confirms that this is a periodic occurrence on the first hours of weekends (Goal 2). Since the East Village is known for its vibrant nightlife, this is indeed expected. By changing the ranking order to similarity to East Village (selecting the East Village node in the node ranking panel), we can explore which zones have similar temporal relevance patterns (Goal 4), which include the Theater District, Clinton East, and Midtown Center.

The most similar region to East Village, in terms of relevance of nodes, is Theater District, which has a different temporal activity pattern, as illustrated in Figure 16. Indeed, the activity for this region is somewhat constant, potentially due to the Port Authority Bus Terminal and the attractions of the region, particularly in this time of the year. While the activity in the East Village and Lower East Side decreased on the holidays, it significantly increased in December 5th.

Because the Theater District is classified as relevant for most of the time slices in each day, which is also the case with East Village, there is significant overlap of relevant time slices between those two regions, which leads to the high similarity value in the node ranking panel. Clinton East and Midtown Center also have a similar behavior, but with different distributions of high frequency among time slices. Lower East Side, while more distant in the ranking, is classified as high frequency on approximately the same time slices as East Village. Indeed, the corresponding time series presents a similar activity pattern, with significant peaks on the first hours of weekends. However,

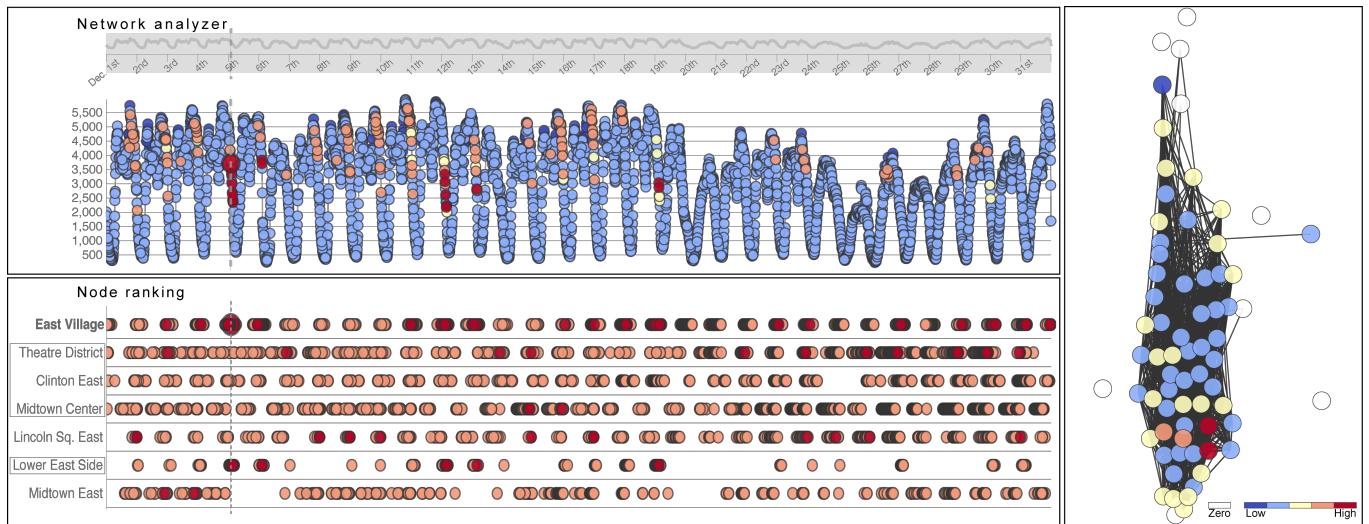


Fig. 15. Proposed interface considering taxi trips in Manhattan in December 2009. The network analyzer presents the total number of taxi pick-ups. The nodes in the time slice view (right) represent taxi zones and they are connected in case of trips between the corresponding zones. The node ranking was adjusted to order zones according to their similarity to East Village.

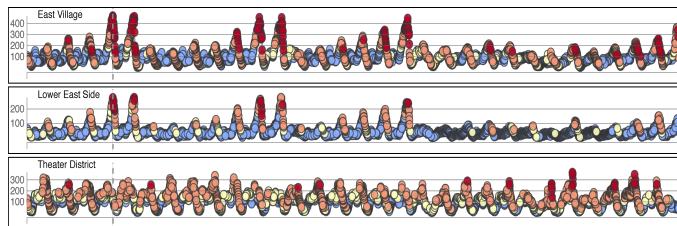


Fig. 16. Node time series to East Village, Lower East Side, and Theater District zones. The vertical dotted lines correspond to December 5th at 02:00.

this region has less overlaps of relevant time slices, because of less activity on the start of the week and during the holidays, which reduced the similarity value between them.

Back to the network analyzer, the number of taxi pick-ups for the whole city was reduced by the holidays, which is expected. The daily patterns are clearly visible, with Sundays having less taxi activity than the weekdays, and December 25th was the day with lowest activity of the month. However, there is a small gap around December 20th, which corresponds to a blizzard that arrived in New York City around 13:30. The afternoon peak in taxi trips, that usually occurred around 20:00 on the nearby days, happened at 19:00 on December 19th, and the morning of the next day, a Sunday, also shows a slower increase in trips, compared to other Sundays of this month. Interestingly, this whole period is classified as light or dark blue, indicating that the patterns of taxi pick-ups were similar over most of the city.

Figure 17 illustrates the proposed interface considering the period between December 24th and December 27th. The most relevant region for this time period is the Theater District (Goal 3), with three pronounced peaks of activity, two red periods around December 25th and 26th, both at 22:30, and one orange in December 25th at 18:30. While the exact reason for these significant peaks is unknown, we conjecture that it can be caused by people returning to the city from nearby locations, on a day with reduced options for transportation, possibly combined with tourists leaving the holiday attractions in the area.

6 DISCUSSION AND LIMITATIONS

While the proposed method has presented great results in the visual analysis of dynamic networks, including georeferenced urban data, we have not fully explored its capabilities. Indeed, we considered only

simple definitions for the signal. Different aspects of the network can be highlighted by more exotic functions, including functions that combine sophisticated network metrics such as closeness and betweenness, for instance. Moreover, this is a concept that provides great flexibility for our method and the exploration of these different options can lead to a multitude of potential new developments and applications.

The representation of the most relevant class in the network analyzer does not enable access to the evolution of the frequency classes over time. A stacked graph metaphor can be used to highlight this information, as in [36]. However, our approach is still able to guide users to interesting phenomena along time, accomplishing this in a simpler way that allows to include more information, the activity level.

An important part of the current interface, the node-link approach for the time slice view does not scale naturally to larger datasets. However, the method itself is compatible with any other method that supports the encoding of at least one information, alternating the representation of the signal or the class.

Importantly, when exploring dynamic networks, usually there is interest in the detection and evolution of communities. While we believe that this problem can be approached using the graph wavelet transform, we considered it beyond the scope of this work; an interesting future work, now that we have established this methodology as a viable alternative for the visual analytics of large dynamic networks.

The most significant weakness of our work resides in the node ranking. While it did provide acceptable results, there is much room for improvements here. We adopted this simple version just to illustrate the concept and potential usefulness.

Another worthwhile future work is the application of our method in the analysis of dynamic multivariate networks, scenario where there are many attributes associated with the nodes and edges of a dynamic network. Once the graph wavelet transform we rely on has been developed based on unique attributes on nodes and edges, it will be necessary the development of new strategies to efficiently include additional attributes in the graph wavelet transform, since it is not straightforward to adapt the current methodology to multivariate attributes.

7 CONCLUSION

We have introduced a methodology for the visual analysis of dynamic networks, leveraging the properties of graph wavelets, including an approximation mechanism to estimate the wavelet coefficients, which has linear complexity in relation to the number of nodes of a network.

By leveraging the wavelet coefficients, our method accurately characterizes the behavior of each node in each time slice. The network

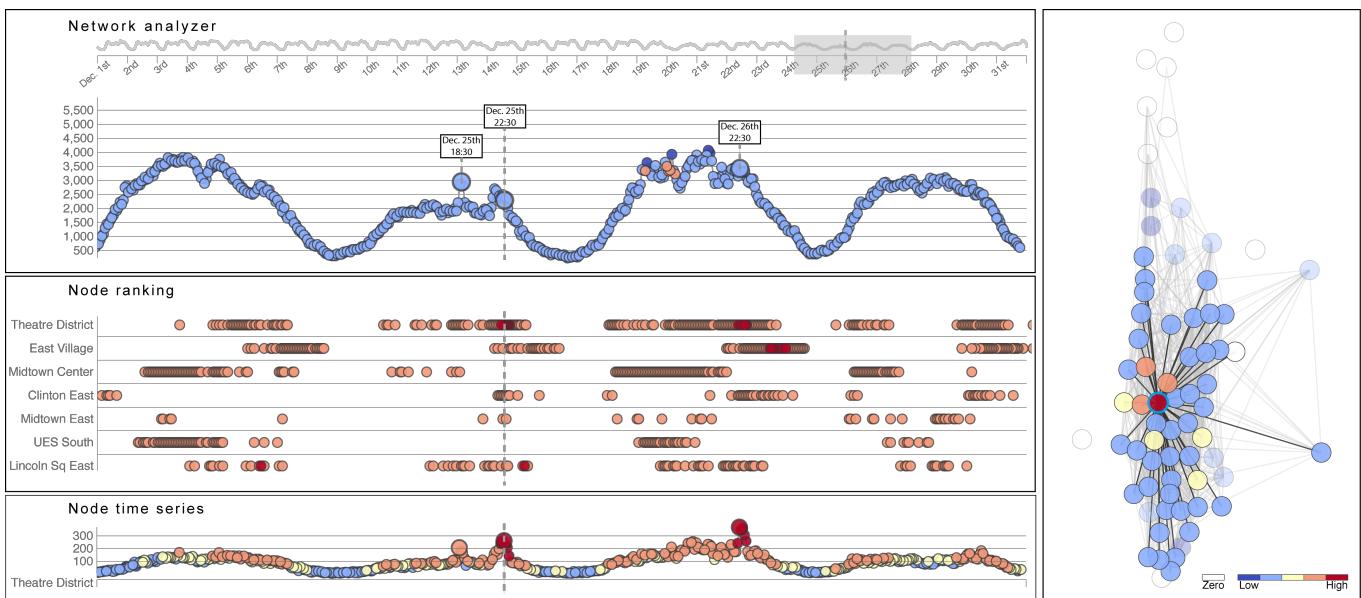


Fig. 17. Proposed interface considering taxi trips in Manhattan in December 2009. The network analyzer presents the total number of taxi pick-ups on the period between December 24th and December 27th (zoomed-in view). In this time period, the Theater District was considered the most relevant region as shown in the node ranking. The other panels, node time series and time slice view (right), highlight this region.

analyzer panel uses this information to summarize the temporal evolution of the network, providing more information than the current state-of-the-art methods for dynamic network exploration. This tool allows for a quick understanding of gist information and also acts as a visual index, greatly reducing the direct inspection of time slices.

Further exploration is enabled by the proposed interface, which contains four linked views that allow the easy identification of consistent network states and subtle/abrupt changes in the associated signal. The nodes are characterized using the wavelet coefficients, this characterization reveals variation patterns that are further analyzed to provide insights for the entire dynamic network, allowing the classification of each time slice into classes that represent the general pattern of variation. The user can use these classes to identify temporal patterns in the evolution of the network and quickly jump between different time slices of same classification to explore similarities.

We have demonstrated the capabilities and effectiveness of the method through experiments and real usage scenarios. Moreover, this initial development can be extended in several ways, potentially leading to enhanced performance and/or applications in different domains. Indeed, the reduced computational cost, due to the approximation mechanism, enables the processing of large graphs in reasonable time.

ACKNOWLEDGMENTS

Grants 2011/22749-8, 2013/14089-3, 2014/12815-1, 2015/03330-7, 2016/04391-2 São Paulo Research Foundation (FAPESP). The views expressed are those of the authors and do not reflect the official policy or position of the São Paulo Research Foundation. Alcebiades acknowledges CAPES (Brazilian government agency for postgraduate studies) for the financial support to the present research work. The authors wish to thank Dr. Harish Doraiswamy for the assistance with the Taxi dataset.

REFERENCES

- [1] D. Archambault, J. Abello, J. Kennedy, S. Kobourov, K.-L. Ma, S. Miksch, C. Muelter, and A. C. Telea. Temporal multivariate networks. In *Multivariate Network Visualization*, pages 151–174. Springer, 2014.
- [2] 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.
- [3] B. Bach, E. Pietriga, and J.-D. Fekete. GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, 2014.
- [4] 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*, 22(1), 2016.
- [5] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The state of the art in visualizing dynamic graphs. In *Eurographics Conference on Visualization - STARs*, pages 83–103. Eurographics Association, 2014.
- [6] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 2016.
- [7] T. Biyikoglu, J. Leydold, and P. F. Stadler. *Laplacian eigenvectors of graphs*. Lecture notes in mathematics. Springer, 2007.
- [8] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf. Parallel edge splatting for scalable dynamic graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2344–2353, 2011.
- [9] N. Cao, C. Shi, S. Lin, J. Lu, Y.-R. Lin, and C.-Y. Lin. Targetvue: Visual analysis of anomalous user behaviors in online communication systems. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):280–289, 2016.
- [10] M. Crovella and E. Kolaczyk. Graph wavelets for spatial traffic analysis. In *Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1848–1857. IEEE, 2003.
- [11] 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*, pages 121–128. IEEE, 2014.
- [12] T. N. Dang, N. Pendar, and A. G. Forbes. Timearcs: Visualizing fluctuations in dynamic networks. *Computer Graphics Forum*, 35(3):61–69, 2016.
- [13] S. Diehl and C. Görg. Graphs, they are changing – dynamic graph drawing for a sequence of graphs. In *Graph Drawing*, GD, pages 23–31. Springer-Verlag, 2002.
- [14] V. N. Ekambaran, G. C. Fanti, B. Ayazifar, and K. Ramchandran. Spline-like wavelet filterbanks for multiresolution analysis of graph-structured data. *IEEE Transactions on Signal and Information Processing over Networks*, 1(4):268–278, 2015.
- [15] C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. V. Yee. GraphAEL: Graph animations with evolving layouts graph drawing. In *Graph Drawing*, GD, pages 98–110. Springer, 2004.
- [16] M. Farrugia, N. Hurley, and A. Quigley. Exploring temporal ego networks using small multiples and tree-ring layouts. *International Conference on Advances in Computer-Human Interactions*, 2011:23–28, 2011.

- [17] J. Fournet and A. Barrat. Contact patterns among high school students. *PLoS ONE*, 9(9):e107878, 2014.
- [18] C. Friedrich and P. Eades. Graph drawing in motion. *Journal of Graph Algorithms and Applications*, 6(3):353–370, 2002.
- [19] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.
- [20] J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg. Evaluation of alternative glyph designs for time series data in a small multiple setting. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 3237–3246. ACM, 2013.
- [21] R. Gove, N. Gramsky, R. Kirby, E. Sefer, A. Sopan, C. Dunne, B. Shneiderman, and M. Taieb-Maimon. Netvisia: Heat map & matrix visualization of dynamic social network statistics & content. In *International Conference on Privacy, Security, Risk and Trust, IEEE International Conference on Social Computing (SocialCom)*, pages 19–26. IEEE, 2011.
- [22] S. Hadlak, H. Schumann, C. H. Cap, and T. Wollenberg. Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2267–2276, 2013.
- [23] S. Hadlak, H. Schumann, and H.-J. Schulz. A survey of multi-faceted graph visualization. In *Eurographics Conference on Visualization - STARs*. EuroVis. The Eurographics Association, 2015.
- [24] D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.
- [25] R. Hamon, P. Borgnat, P. Flandrin, and C. Robardet. Extraction of temporal network structures from graph-based signals. *IEEE Transactions on Signal and Information Processing over Networks*, PP(99):1–1, 2016.
- [26] B. A. Miller, M. S. Beard, P. J. Wolfe, and N. T. Bliss. A spectral framework for anomalous subgraph detection. *IEEE Transactions on Signal Processing*, 63(16):4191–4206, 2015.
- [27] 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*, pages 1707–1712. IEEE, 2014.
- [28] 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(5):80–90, 2014.
- [29] A. Sandryhaila and J. M. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013.
- [30] P. Saraiya, P. Lee, and C. North. Visualization of graphs with associated timeseries data. In *IEEE Symposium on Information Visualization*, pages 225–232. IEEE, 2005.
- [31] 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, 2013.
- [32] D. I. Shuman, P. Vandergheynst, and P. Frossard. Chebyshev polynomial approximation for distributed signal processing. In *IEEE International Conference on Distributed Computing in Sensor Systems and Workshops*, pages 1–8. IEEE, 2011.
- [33] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer Graphics Forum*, volume 33, pages 401–410. Wiley Online Library, 2014.
- [34] K. Stein, R. Wegener, and C. Schlieder. Pixel-oriented visualization of change in social networks. In *IEEE International Conference on Advances in Social Networks Analysis and Mining*, pages 233–240. IEEE, 2010.
- [35] N. Tremblay and P. Borgnat. Graph wavelets for multiscale community mining. *IEEE Transactions on Signal Processing*, 62(20):5227–5239, 2014.
- [36] P. Valdovina, 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*. IEEE, 2015.
- [37] 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*, 20(8):1087–1099, 2014.
- [38] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1–10, 2016.
- [39] C. Vehlow, F. Beck, and D. Weiskopf. The state of the art in visualizing group structures in graphs. In *Eurographics Conference on Visualization - STARs*. EuroVis. The Eurographics Association, 2015.
- [40] C. Vehlow, F. Beck, and D. Weiskopf. Visualizing dynamic hierarchies in graph sequences. *IEEE Transactions on Visualization and Computer Graphics*, 22(10):2343–2357, Oct 2016.
- [41] T. von Landesberger, S. Diel, S. Bremm, and D. W. Fellner. Visual analysis of contagion in networks. *Information Visualization*, 14(2):93–110, 2015.
- [42] J. S. Yi, N. Elmquist, and S. Lee. Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction*, 26(11-12):1031–1051, 2010.
- [43] C. Zhang, D. Florêncio, and C. Loop. Point cloud attribute compression with graph transform. In *IEEE International Conference on Image Processing*, pages 2066–2070. IEEE, 2014.