UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Graph signal processing for visual analysis and data exploration

Paola Tatiana Llerena Valdivia

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Paola Tatiana Llerena Valdivia

Graph signal processing for visual analysis and data exploration

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Luis Gustavo Nonato

USP – São Carlos
July 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

L791g Llerena Valdivia, Paola Tatiana
Graph signal processing for visual analysis and
data exploration / Paola Tatiana Llerena Valdivia;
orientador Luis Gustavo Nonato. -- São Carlos, 2018.
88 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2018.

1. Analise visual. 2. Processamento de sinais em
grafos. 3. Visualização de informação. 4. Filtragem
de dados. 5. Transformada wavelets em grafos. I.
Nonato, Luis Gustavo, orient. II. Título.

Paola Tatiana Llerena Valdivia

**Processamento de sinais em grafos para análise visual e
exploração de dados**

Tese apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Doutora em Ciências – Ciências de Computação e
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientador: Prof. Dr. Luis Gustavo Nonato

USP – São Carlos
Julho de 2018

*A mis primeros maestros,
mis abuelitos Jaime y Norma.*

ACKNOWLEDGEMENTS

Firstly, I want to thank Luis Gustavo Nonato, my mentor, for challenging myself during my PhD and MSc periods. Thanks for helping and guiding me during these years. I also want to thank my colleagues of the Visual and Geometry Processing Group, specially Fábio and Alcebiades for the important contributions to this work.

To my beloved family: Jaime (mi abuelito), Norma (mi abuelita), Tatiana (mi mami), Neg (mi sis), and Ross (mi hermanita). I also want to thank my dad Jaime.

To my friends in São Carlos, specially to: Neg, Jorge, Alfredo, Edna, Aurea, Oscar and Erick.

To my mentor from AVIZ, Jean-Daniel Fekete; and to my friends from AVIZ: Lonni and Christoph.

An special thanks to Fundação de Amparo à pesquisa do Estado de São Paulo - FAPESP (Process#2013/14089-3, Process#2016/10532-8) for funding my research during my PhD and my internship abroad.

I also want to thank to all professors and faculty members from ICMS-USP who, directly or indirectly contributed to this work. I am sure there are many names missing here, to all of them, thanks!

ABSTRACT

VALDIVIA, P. **Graph signal processing for visual analysis and data exploration**. 2018. 88 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Signal processing is used in a wide variety of applications, ranging from digital image processing to biomedicine. Recently, some tools from signal processing have been extended to the context of graphs, allowing its use on irregular domains. Among others, the Fourier Transform and the Wavelet Transform have been adapted to such context. Graph signal processing (GSP) is a new field with many potential applications on data exploration. In this dissertation we show how tools from graph signal processing can be used for visual analysis. Specifically, we proposed a data filtering method, based on spectral graph filtering, that led to high quality visualizations which were attested qualitatively and quantitatively. On the other hand, we relied on the graph wavelet transform to enable the visual analysis of massive time-varying data revealing interesting phenomena and events. The proposed applications of GSP to visually analyze data are a first step towards incorporating the use of this theory into information visualization methods. Many possibilities from GSP can be explored by improving the understanding of static and time-varying phenomena that are yet to be uncovered.

Keywords: Visual analysis, Graph signal processing, Information visualization, Graph wavelet transform.

RESUMO

VALDIVIA, P. **Processamento de sinais em grafos para análise visual e exploração de dados**. 2018. 88 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

O processamento de sinais é usado em uma ampla variedade de aplicações, desde o processamento digital de imagens até a biomedicina. Recentemente, algumas ferramentas do processamento de sinais foram estendidas ao contexto de grafos, permitindo seu uso em domínios irregulares. Entre outros, a Transformada de Fourier e a Transformada Wavelet foram adaptadas nesse contexto. O Processamento de Sinais em Grafos (PSG) é um novo campo com muitos aplicativos potenciais na exploração de dados. Nesta dissertação mostramos como ferramentas de processamento de sinal gráfico podem ser usadas para análise visual. Especificamente, o método de filtragem de dados porposto, baseado na filtragem de grafos espectrais, levou a visualizações de alta qualidade que foram atestadas qualitativa e quantitativamente. Por outro lado, usamos a transformada de wavelet em grafos para permitir a análise visual de dados massivos variantes no tempo, revelando fenômenos e eventos interessantes. As aplicações propostas do PSG para analisar visualmente os dados são um primeiro passo para incorporar o uso desta teoria nos métodos de visualização da informação. Muitas possibilidades do PSG podem ser exploradas melhorando a compreensão de fenômenos estáticos e variantes no tempo que ainda não foram descobertos.

Palavras-chave: Análise visual, Processamento de sinais em grafos, Visualização de informação, Transformada wavelets em grafos.

LIST OF FIGURES

Figure 1 – A few examples of graphs that describe relational structure of data.	21
Figure 2 – Signals defined on an path graph with 64 nodes and its respective GFT. . . .	23
Figure 3 – Signals defined on an irregular graph and its respective GFT.	26
Figure 4 – Low-pass and enhancement filtering of a noisy step function.	28
Figure 5 – Graph spectral filtering towards enhancing the quality of visualizations. . . .	29
Figure 6 – Low-pass filter to the signals corresponding to the x and y coordinates of three bivariate Gaussians.	33
Figure 7 – Graph attributes represented as a one-dimensional signal.	34
Figure 8 – Scatter Plot Matrix of the synthetic dataset: original data.	35
Figure 9 – Scatter Plot Matrix of the synthetic dataset: low-pass filtered data.	36
Figure 10 – Scatter Plot Matrix of the synthetic dataset: enhancement filtered data. . . .	37
Figure 11 – Average quality measure of the synthetic dataset scatter plots. The larger the averages the better.	38
Figure 12 – Parallel Coordinates view of the wine dataset with over-plotting optimal axes arrangement.	39
Figure 13 – Parallel Coordinates view of the wine dataset with parallelism optimal axes arrangement.	40
Figure 14 – Parallel coordinates quality measures for the original (green), low-pass fil- tered (purple), and enhancement filtered (orange) data.	41
Figure 15 – Lamp projection of the Ecoli dataset. Left: original data. Right: enhanced filtered data.	41
Figure 16 – a) Lamp projection of the Eggs dataset using the original (left) and low-pass filtered (right) data. b) PCA projection of the Eggs dataset using the original (left) and low-pass filtered (right) data.	42
Figure 17 – Quality measure of projection layouts using the original (green), low-pass (purple), and enhancement filtered (orange) data.	44
Figure 18 – Spectral Graph Wavelet coefficients of nodes for a signal defined on a path graph.	49
Figure 19 – Filter design: Graph spectral filters for the SGW and SAGW dictionaries. . .	50
Figure 20 – Spectral Graph Wavelet and Spectral Adapted Graph Wavelet coefficients of nodes for a signal defined on a cometh graph.	51
Figure 21 – Time series integrated views in 2D.	54
Figure 22 – Time series integrated views in 3D.	56

Figure 23 – Time series multiple views.	57
Figure 24 – Cartesian product $G \times H$. Source: (VALDIVIA <i>et al.</i> , 2015).	62
Figure 25 – Dimensional feature vectors characterizing four frequency classes.	63
Figure 26 – A function f in a specific time instant.	64
Figure 27 – Stacked graph illustrating how each of the five classes evolve over time. Source: (VALDIVIA <i>et al.</i> , 2015).	65
Figure 28 – Screenshot our prototype interface.	66
Figure 29 – Coefficients and raw data for four distinct time slices.	68
Figure 30 – Time series associated to a node in the bottom blue region in Figure 29b. Source: (VALDIVIA <i>et al.</i> , 2015).	68
Figure 31 – <i>Citi Bike NY</i> station map. Stations from downtown Manhattan.	69
Figure 32 – Time-slices of the density maps of bikes in downtown Manhattan in October 28, 2013.	71
Figure 33 – High frequency variations tend to occur on the bike lanes, particularly on their intersections. Source: (VALDIVIA <i>et al.</i> , 2015).	72
Figure 34 – Stacked graph showing one week of bike active from October 15th to 31st, 2013.	72
Figure 35 – Stacked graph visualization of one week of taxi pickups from August 12th to 18th.	73
Figure 36 – Four different time slices for the taxi activity, corresponding to August 12, 2013.	74
Figure 37 – Time series associated to the node representing the intersection of Liberty street and South End avenue.	74
Figure 38 – Time series associated to the vertex representing the node highlighted in the fourth column of Figure 36. Source: (VALDIVIA <i>et al.</i> , 2015).	76

CONTENTS

1	INTRODUCTION	17
1.1	Outline	19
2	GRAPH SIGNAL PROCESSING	21
2.1	Spectral graph theory	22
2.2	Graph Fourier transform	24
2.3	Spectral Filtering	27
3	GRAPH SPECTRAL FILTERING TOWARDS ENHANCING THE QUALITY OF VISUALIZATIONS	29
3.1	Signal processing in visualization	30
3.2	Graph Spectral Filtering in Multidimensional Data	31
3.3	Results	34
3.3.1	<i>Data Sets</i>	34
3.3.2	<i>Scatter Plot Matrix</i>	34
3.3.2.1	<i>Parallel Coordinates</i>	39
3.3.2.2	<i>Multidimensional Projections</i>	42
3.4	Discussion and Limitations	43
3.5	Final Considerations	44
4	GRAPH WAVELET TRANSFORM	47
4.1	Graph Wavelets review	47
4.2	Spectral Graph Wavelet Transform	48
5	SPATIOTEMPORAL ANALYSIS AND VISUALIZATION ON GRAPHS	53
5.1	Visualization of spatial time-series	53
5.1.1	<i>Integrated views</i>	53
5.1.2	<i>Multiple views</i>	56
5.2	Visual analytics of spatiotemporal data	58
5.2.1	<i>Mobility patterns</i>	59
6	GRAPH WAVELETS ANALYSIS ON TIME-VARYING DATA . . .	61
6.1	Node classification and stacked graphs	62
6.2	Computational times	65

6.3	Visual elements and linked views	65
6.4	Visual analytics with graph Wavelets	66
6.5	Case studies	69
6.5.1	<i>Citi Bike</i>	69
6.5.2	<i>NY Taxi</i>	72
6.6	Discussion and limitations	76
6.7	Final considerations	77
7	CONCLUSIONS	79
	BIBLIOGRAPHY	81

INTRODUCTION

Signal processing has long been a fundamental tool in fields such as image processing (OPPENHEIM, 1999), computer vision (GRANLUND; KNUTSSON, 2013), and computer graphics (ZHANG; Van Kaick; DYER, 2010), leveraging the development of filtering mechanisms designed to tackle problems such as denoising (BUADES; COLL; MOREL, 2005; LU; DENG; CHEN, 2016), anomaly detection (KANG *et al.*, 2017), detail enhancement (GASTAL; OLIVEIRA, 2012; VALLET; LÉVY, 2008), object registration (REDDY; CHATTERJI, 1996; LITANY *et al.*, 2017), among others.

Recently, the theory of graph signal processing has emerged as a main alternative to analyze signals defined on the vertices of graphs (SHUMAN *et al.*, 2013). It focuses on adapting tools and operators, such as Fourier and Wavelet transforms, to the context of graphs, opening a multitude of possibilities to the exploration and understanding of static and time-varying phenomena in irregular domains.

Similarly to the case of regular domains, most of the analysis derived from Graph Fourier (GFT) and Graph Wavelet Transforms (GWT) (MOHAN *et al.*, 2014; VALDIVIA *et al.*, 2015; TREMBLAY; BORGNAT, 2014; Dal Col *et al.*, 2017b), relies on the processing of coefficients associated with certain basis functions defined on the vertices of a graph (HAMMOND; VANDERGHEYNST; GRIBONVAL, 2011). However, the irregular nature of graph domains makes the coefficients sensitive not only to the properties of the signal under analysis, but also to the topology of the graph (SHUMAN *et al.*, 2015).

The sensitivity of the GFT and GWT to signal variation across irregular domains is an evident and noteworthy characteristic of these transforms, which can be used to understand phenomena related to daily life. For example, the street map of a city can be represented as a graph and the way the data under analysis varies across streets, from corner to corner, may reveal important aspects of city's dynamics.

Another interesting scenario is the case of unstructured data, which we take to be high-

dimensional points without intrinsic connectivity. It is well known that data acquisition can lead to noisy information, thus spectral graph filtering schemes can be useful to reduce noise and emphasize structures hidden in the data. Regardless of the possibilities, not many efforts have been devoted to incorporating graph signal processing tools into visual analysis for data exploration.

In this dissertation, we address the gap mentioned above to support effective visual data analysis that relies on graph signal processing tools. Specifically, we address two problems, firstly we show how to assist high-dimensional data visualization using signal processing tools. The second addressed problem is how to leverage GWT to enable the visual analysis of urban data directing users to interesting patterns and changes. We also discuss some aspects on how to understand the interplay between GFT and GWT coefficients and how to interpret this coefficients so they can be used to uncover patterns and interesting phenomena hidden in the data.

In summary, this dissertation gathers techniques and results developed and published during the doctorate term. Specifically, it puts together the results from the following works:

1. “Wavelet-based visual analysis for data exploration” Col, A. D., **Valdivia, P.**, Petronetto, F., Dias, F., Silva, C. T., and Nonato, L. G. (2017a). *Computing in Science Engineering*, 19(5):85–91 ([Dal Col et al., 2017a](#)).
 - a discussion on some aspects of the GFT and GWT, providing examples of how the signal and the topology of the graph impact the computation of coefficients;
 - a set of controlled experiments, using real and synthetic data, to illustrate how the topology and signal can affect the result of GFT and GWT.
2. “Wavelet-based visualization of time-varying data on graphs” **Valdivia, P.**, Dias, F., Petronetto, F., Silva, C. T., and Nonato, L. G. (2015). *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1 ([VALDIVIA et al., 2015](#)).

The contributions of this work are:

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

Additional works developed in the doctorate period:

1. “Wavelet-based visual analysis of dynamic networks”. Col, A. D., **Valdivia, P.**, Petronetto, F., Dias, F., Silva, C. T., and Nonato, L. G. (2017b). *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1 ([Dal Col et al., 2017b](#)).
2. “Watersheds on Hypergraphs for Data Clustering”. Dias, F., Mansour, M. R., **Valdivia, P.**, Cousty, J., and Najman, L. (2017), *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. PP 211–221 ([DIAS et al., 2017](#)).
3. “Analyzing Dynamic Hypergraphs with Parallel Aggregated Ordered Hypergraph Visualization”. **Valdivia, P.**, Buono, P., Plaisant C., and Fekete, J.-D. (2018).
Submitted to IEEE InfoVis 2018. (IEEE Information Visualization (InfoVis) Conference).

1.1 Outline

This dissertation is organized in five main parts: The first part (Chapter 2), which serves as background for the methodologies developed in the second part, covers specific concepts on graph signal processing and spectral graph filtering. The second part (Chapter 3), which is part of the contributions presented in this dissertation, presents in detail the proposed methodology to apply graph spectral filtering for enhancing the quality of high-dimensional data visualizations. The third part (Chapter 4) presents concepts on the Graph Wavelet Transform, concepts that support our methodology presented in the fourth part; it also presents a discussion on how the graph topology and properties of the signal affect the wavelet coefficients. The fourth part (Chapters 5 and 6) shows a methodology, developed during the doctorate, on how Wavelet coefficients can enable visual analysis in urban data to identify patterns and outliers. Finally, the fifth part (Chapter 7) present the conclusions and future work.

GRAPH SIGNAL PROCESSING

Graphs naturally describe the relational structure of data in many scenarios, including data on social, brain, and transportation networks (Figure 1). Each edge of the graph can have an associated value, usually called weight, representing, for instance, geographical proximity or similarity between nodes connected by such edge. Such graphs are denominated *weighted graphs*. Moreover, depending on whether edges have or not a direction, graphs are referred as directed or undirected graphs respectively. In this work, we only consider undirected weighted graphs.

Formally, a graph $G = \{V, E\}$ is defined by a set of nodes $V = \{\tau_1, \tau_2, \dots, \tau_n\}$ and a set of edges E . In an undirected graph, an edge is an unordered pair of nodes. If there exists an edge connecting nodes τ_i and τ_j , i.e. $(\tau_i, \tau_j) \in E$, those nodes are said to be adjacent or neighbors. We can represent a graph G using its adjacency matrix $\mathbf{A}$. The entries a_{ij} of this matrix are equal to one if and only if $(\tau_i, \tau_j) \in E$.

Weighted graphs can be represented by a triplet $G^w = \{V, E, w\}$, where in addition to V and E , there is a weight function $w : E \rightarrow \mathbb{R}$ that associates a non-negative scalar to each edge of G . The *weighted adjacency matrix* of G , denoted as $W = (w_{ij})$, is the matrix satisfying

Figure 1 – A few examples of graphs that describe relational structure of data.

$w_{ij} = w(\tau_i, \tau_j)$ if τ_i is adjacent to τ_j and $w_{ij} = 0$ otherwise. These entries represent the weight of the edges between nodes. By construction, W is a symmetric square matrix of order n . In our context, besides of assuming an undirected weighted graph, G is assumed to be *connected*, that is, for every pair of nodes there is a sequence of adjacent edges connecting the nodes.

A signal defined on the nodes of G , $f : V \rightarrow \mathbb{R}$, is a function that associates a real value $f(\tau_i)$ to each node $\tau_i \in V$. For example, left column in Figure 2 shows some signals defined in a path graph with 64 nodes, in which each node, except the two extreme ones, is adjacent to exactly two other nodes (top in Figure 2). In Figure 2 the amplitude of the signal is represented using a double encoding: the height of the vertical blue lines and the color of the node.

2.1 Spectral graph theory

Spectral graph theory is the study of graph properties by means of algebraic computations from a graph matrix. Such matrix could be the adjacency or the Laplacian matrix, among others. The graph spectrum and its applications have been extensively studied, and are well documented in several surveys and books (CVETKOVIĆ; ROWLINSON; SIMIC, 2009; CHUNG, 1997; BIYIKOGLU; LEYDOLD; STADLER, 2007).

In this work, we are specially interested in the graph Laplacian spectrum because it provides with the means of naturally extending the Fourier transform to the context of graphs, allowing the definition of signal processing operations on graphs, including the Wavelets transform.

For a real valued function $f : V \rightarrow \mathbb{R}$, the graph Laplace operator for graphs is defined as $(\Delta f)(i) = \sum_{(\tau_i, \tau_j) \in E} w_{ij} [f(i) - f(j)]$. This is an analog of the continuous Laplace operator, which clearly can be seen in a grid graph, where applying the operator to any non-boundary node leads to:

$$\begin{aligned} -(\Delta f)(i) &= [f(x_0 + 1, y_0) - f(x_0, y_0)] - [f(x_0, y_0) - f(x_0 - 1, y_0)] \\ &\quad + [f(x_0, y_0 + 1) - f(x_0, y_0)] - [f(x_0, y_0) - f(x_0, y_0 - 1)] \\ &\approx \frac{\partial^2 f}{\partial x^2}(x_0, y_0) + \frac{\partial^2 f}{\partial y^2}(x_0, y_0) = (\Delta f)(x_0, y_0), \end{aligned} \quad (2.1)$$

which is the Laplacian finite difference approximation called *five-point stencil*.

Using matrix notation the Laplacian can be defined as:

$$L := D - W, \quad (2.2)$$

where $D = \text{diag}(d_0, d_2, \dots, d_{n-1})$ is a diagonal matrix with entries $d_i = \sum_j w_{ij}$ and n is the number of nodes in V . The graph Laplacian is a real, symmetric, and positive-semidefinite matrix, what ensures a complete set of orthonormal real eigenvectors $\{u_\ell\}_{\ell=1, \dots, n-1}$, with corresponding non-negative real eigenvalues $\Lambda = \{\lambda_\ell\}_{\ell=1, \dots, n-1}$. Assuming a connected graph, eigenvalues can

Figure 2 – **Signal:** eigenvectors u_1 , u_2 , u_3 , u_{10} , u_{32} , u_{63} , and f_0 signals defined on a path graph with 64 nodes. Signal f_0 is a combination of the low (u_3) and high frequency (u_{63}) signals. Circles represent the graph nodes, and the height of the vertical blue lines and the color of the nodes correspond to the amplitude of the signal. **GFT:** The GFT are indicated by arrows. The GFT corresponding to the signal f_0 indicates the presence of low and high frequencies.

be ordered $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n := \lambda_{\max}$. Moreover, zero is always an eigenvalue of L whose corresponding eigenvector is a constant vector.

The graph Laplacian eigenvectors $u_\ell : V \rightarrow \mathbb{R}$ can be interpreted as signals on the graph G . The left column in Figure 2, from top to bottom, shows the eigenvectors $u_1, u_2, u_3, u_{10}, u_{32}, u_{63}$ of the graph Laplacian for a path graph with 64 nodes. It can be seen that eigenvectors associated to small eigenvalues tend to have a less oscillatory behavior than eigenvectors associated to large eigenvalues. Therefore, the eigenvalues and eigenvectors of the graph Laplacian play a similar role as frequencies and basis functions in the classical Fourier theory. More specifically, small eigenvalues, that is, the ones closer to the eigenvalue zero, correspond to low frequencies, while large eigenvalues correspond to high frequencies. A more detailed discussion about the relation between the spectrum of Laplacian matrices and Fourier theory can be found in the work by [Shuman, Ricaud and Vandergheynst \(2016\)](#).

The *normalized Laplacian matrix* is defined as:

$$\mathcal{L} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}}. \quad (2.3)$$

Such matrix also has n real non-negative eigenvalues $\{\tilde{\lambda}_\ell\}_{\ell=1,\dots,n}$ and a complete set of orthonormal eigenvectors $\{\tilde{u}_\ell\}_{\ell=1,\dots,n}$. Moreover, on a connected graph the eigenvalues of $\mathcal{L}$ are bounded: $0 = \tilde{\lambda}_1 < \tilde{\lambda}_2 \leq \dots \leq \tilde{\lambda}_n := \tilde{\lambda}_{\max} \leq 2$, with $\tilde{\lambda}_{\max} = 2$ if and only if the graph is bipartite.

Non-normalized and normalized Laplacian matrices, along with the adjacency matrix, can be used to extend the classical Fourier theory to graphs. In this work we will use both the non-normalized and normalized Laplacian matrices. More details on using the adjacency matrix to extend the Fourier theory can be found in the work by [Sandryhaila and Moura \(2014\)](#).

2.2 Graph Fourier transform

The classic Fourier transform is defined as the expansion of a function f in terms of complex exponentials:

$$\hat{f}(\xi) = \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{-2\pi i \xi t} dt, \quad (2.4)$$

where ξ stands for the frequency and t for the time. Such complex exponentials are eigenfunctions of the Laplacian operator, in one dimension:

$$-\Delta(e^{2\pi i \xi t}) = -\frac{\partial^2}{\partial t^2} e^{2\pi i \xi t} = (2\pi \xi)^2 e^{2\pi i \xi t}. \quad (2.5)$$

By analogy to the continuous Laplacian operator, where eigenfunctions and eigenvalues are the Fourier modes and frequencies respectively, the u_ℓ and λ_ℓ , $\ell = 0, \dots, n-1$ are considered as graph Fourier modes and frequencies. Therefore, small values of λ_ℓ correspond to low

frequency modes while large values indicate high frequency modes. The *graph Fourier transform* of $\mathbf{f}$ at the frequency λ_ℓ is defined as (HAMMOND; VANDERGHEYNST; GRIBONVAL, 2011):

$$\hat{f}(\lambda_\ell) = \langle u_\ell, f \rangle = \sum_{j=1}^n u_\ell(\tau_j) f(\tau_j), \quad (2.6)$$

where λ_ℓ is an eigenvalue of L , u_ℓ is its corresponding eigenvector and $u_\ell(\tau_j)$ is the value of the eigenvector at the position j . Since the GFT assigns a scalar value to each eigenvalue (frequency) $\lambda_\ell \in \Lambda$, one can visualize the result of a GFT by plotting the pairs $(\lambda_\ell, \hat{f}(\lambda_\ell))$, $i = 0, 1, \dots, n-1$, using a bar-like plot as illustrated on the right of Figure 2.

Similarly to the classic Fourier transform, the GFT reveals the frequencies present in a signal. High frequency coefficients $\hat{f}(\lambda_\ell)$ (when λ_ℓ is in the rightmost part of the spectrum) indicate that a signal varies abruptly in some regions of the graph, while low frequency coefficients suggest smooth signal variation. In order to illustrate this concept and show the relation with classic Fourier transform, we consider the path graph with 64 nodes of Figure 2. We consider seven eigenvectors as signals, four of low frequencies (u_1, u_2, u_3, u_{10}), one mid-frequency (u_{32}) and one of high frequency (u_{63} , as illustrated in Figure 2. The GFT of those signals is also depicted on the right of Figure 2, corresponding to the spikes on the corresponding frequencies, the same behavior as in the classic Fourier transform.

The GFT provides a global description of the signal behavior on the graph. For example, let f_0 be a signal defined as a combination of one low and high frequency eigenvectors (u_3 and u_{63} respectively), as illustrated on the bottom left of Figure 2. The GFT of f_0 , shown on the bottom right of Figure 2, indicates the presence of low and high frequencies in the signal, suggesting that the signal has both smooth and abrupt variation.

In Figure 3 we illustrate the GFT of some signals defined on an irregular graph with 400 nodes, colors indicate the values of the signal in each node. The first two rows of Figure 3 show the GFT of six eigenvectors as signals on the graph: three low frequencies (u_1, u_2, u_{11}), one mid-frequencies (u_{211}) and two high frequencies (u_{380} and u_{399}). f_0 (bottom left of Figure 3) shows the GFT of a Gaussian function on the same irregular graph. As depicted on the bottom left of Figure 3, the largest coefficients of the GFT of f_0 are concentrated in the low frequency region of the spectrum (smaller eigenvalues). In other words, the smooth global variation of this function is captured by the GFT. On the bottom of Figure 3, we show other signals, f_1 and f_2 to illustrate the behavior of the GFT in different scenarios. Most regions of f_2 have adjacent nodes with the same value, except regions on the “border” of the “V”, which is reflected in larger values on the low frequencies. On the other hand, f_1 has only some nodes with different value from its neighbors, which can be seen in the presence of all kind of frequencies in the spectrum domain.

Under certain conditions, the classic inverse Fourier transform is given by:

$$f(t) = \int_{\mathbb{R}} \hat{f}(\xi) e^{2\pi i \xi t} d\xi, \quad (2.7)$$

Figure 3 – Signals defined on an irregular graph and its respective GFT. In every case, the signal is encoded using the color of each node and its respective GFT is shown on bottom of each graph. The two first rows show eigenvectors $u_1, u_2, u_{11}, u_{211}, u_{380}, u_{399}$ signals defined on an irregular graph with 400 nodes. f_0 is a signal defined as a Gaussian function centered in a given node. f_1 and f_2 are signals defined to show the behavior of the GFT in other scenarios.

and its corresponding *inverse Graph Fourier Transform* (iGFT) is defined as:

$$f = \sum_{\ell=1}^n \hat{f}(\lambda_\ell) u_\ell \quad (2.8)$$

If we denote by U the (orthogonal) matrix with columns given by the eigenvectors u_ℓ , the GFT and iGFT can be obtained by matrix multiplication as follows:

$$\begin{array}{cc} \boxed{\text{GFT}} & \boxed{\text{iGFT}} \\ \hat{f} = U^\top f & f = U \hat{f} \end{array} \quad (2.9)$$

2.3 Spectral Filtering

A *graph spectral filter*, or kernel, $\hat{h} : \Lambda \rightarrow \mathbb{R}$ is a function defined in the spectral domain that associates a scalar value $\hat{h}(\lambda_\ell)$ to each eigenvalue $\lambda_\ell \in \Lambda$. The GFT $\hat{f} : \Lambda \rightarrow \mathbb{R}$ can be seen as a particular instance of a graph spectral filter.

A *graph spectral filtering* of a signal f is defined as:

$$\tilde{f} = \hat{h} \hat{f} \quad (2.10)$$

where $\hat{f}$ is the GFT of f and $\hat{h}$ a graph spectral filter. The filtered version $\tilde{f}$ of f in the graph domain is

$$\tilde{f}(\tau_j) = \sum_{\ell=1}^n \hat{h}(\lambda_\ell) \hat{f}(\lambda_\ell) u_\ell(\tau_j) \quad (2.11)$$

Using the matrix notation defined in Equation 2.9 after some algebraic manipulation one can obtain :

$$\tilde{f} = U H U^\top f \quad (2.12)$$

where H is a diagonal matrix with entries $\hat{h}(\lambda_1), \dots, \hat{h}(\lambda_n)$.

The design of a proper filter $\hat{h}$ is application dependent, some examples can be found in the work by [Shuman et al. \(2013\)](#). In this work we will use low-pass $\hat{h}_l$, high-pass $\hat{h}_h$, and enhancement $\hat{h}_e$ filters, which are defined, among many possibilities, as:

$$\hat{h}_l(\lambda) = \exp\left(\frac{(-\alpha \lambda)^2}{(2 \lambda_{\max})^2}\right), \quad \alpha > 0 \quad (2.13)$$

$$\hat{h}_h(\lambda) = 1 - \hat{h}_l(\lambda) \quad (2.14)$$

$$\hat{h}_e(\lambda) = \beta f_l(x) + (1 - \beta) f_h(x), \quad \beta \in [0, 1] \quad (2.15)$$

where $\lambda_{\max}$ is the largest eigenvalue and α, β are user defined parameters.

Figure 4 illustrates the effect of a low-pass filter and an enhancement when applied in a noisy step function.

Figure 4 – Top: Step function (left) and its noisy version (right). Middle: Low-pass (left) and Enhancement (right) filters. Bottom: Noisy step function filtered with the low-pass (left) and enhancement (right) filters.

GRAPH SPECTRAL FILTERING TOWARDS ENHANCING THE QUALITY OF VISUALIZATIONS

We propose a methodology to adapt the theory of Graph Signal Processing (GSP) to filter high-dimensional data towards enhancing the quality of visualizations. The proposed filtering mechanisms result in less cluttered visualizations while highlighting important structures contained in the data, making the visual layouts cleaner and more informative, as illustrated in Figure 5. The effectiveness of the proposed methodology is shown through a set of quantitative measures that gauge the impact of the filtering process to visualizations. The performed experiments reveal that even simple band-pass filters can remarkably improve visualizations such as parallel coordinates, scatter plots and multidimensional projections, attesting the importance of incorporating filtering mechanisms as integral part of high-dimensional data visualization process.

Figure 5 – Graph spectral filtering can reduce noise and emphasize structures present in the data. From left to right, multidimensional projection, scatter plot matrix, and parallel coordinates visualizations before and after data filtering.

3.1 Signal processing in visualization

In this section we provide an overview of techniques that rely on signal processing apparatus as underlying resource to support visualization. Indirect filtering such as aggregation (ANDRIENKO; ANDRIENKO, 2008), summarization (LINS; KLOSOWSKI; SCHEIDEGGER, 2013), and data pruning/selection (WEGMAN, 2003) are not in our scope and will not be considered. The goal in this section is not to provide a comprehensive survey about methods that somehow make use of signal processing concepts to leverage visualizations. Rather, we focus on the diversity of use of signal processing tools in the context of visualization, emphasizing the broad scope of the classical theory and the potential of its recent graph-based variant.

Classical Signal Processing in Visualization

The classical signal processing machinery has been employed to support a variety of visualization methods. In the context of scientific visualization, for example, volume rendering techniques have built upon Fourier transform to properly sample volume rendering integrals (BERGNER *et al.*, 2006; SCHULZE *et al.*, 2003) and to filter aliasing effects introduced during the perturbation of opacity maps (KHLEBNIKOV *et al.*, 2013). In fact, the need for data filtering resource has long been acknowledged in context of volume rendering (SAKAS; SCHREYER; GRIMM, 1995; VIOLA; KANITSAR; GROLLER, 2003), being investigated until the present time (SOLTESZOVA *et al.*, 2017). Fourier transform is also the basis of methods designed to identify and visualize patterns in vector fields (EBLING; SCHEUERMANN, 2005; REICH; HLAWITSCHKA; SCHEUERMANN, 2015). Filters designed to operate in spatial domains has also been employed to remove low-frequency components from spot-noise based vector field visualizations (de Leeuw; Van Wijk, 1995).

Classical signal processing also plays a role in information visualization. A good example is the edge bundling technique proposed by Lhuillier, Hurter and Telea (2017), which maps the bundling problem to the spectral domain in order to speed up computation through the Fourier convolution property. Image-based filtering has been applied to detect outliers and trends in 2D binned data associated to pairs of adjacent axes in parallel coordinates visualizations, enabling output-oriented visualizations (NOVOTNY; HAUSER, 2006). Low-pass filtering has been applied to smooth out density maps in illustrative parallel coordinates (MCDONNELL; MUELLER, 2008).

Graph Signal Processing in Visualization

The more recent framework of graph signal processing has fostered some developments in visualization. For instance, the method proposed by Valdivia *et al.* (2015), discussed later in this dissertation on Chapter 6, relies on graph wavelet transform to identify low and high frequency patterns in graphs where time varying data is associated to the nodes. A similar

approach has been proposed by Dal Col *et al.* (2017b) in the context of dynamic networks, building upon graph wavelet coefficients to visually identify time slices where the network assumes “low” and “high-frequency” configurations. Graph Fourier transform is also the basis of the method proposed by Huang *et al.* (2015), which decomposes brain signals into pieces that correspond to smooth or rapid signal variations, enabling the visual analysis of different brain activities.

From the discussion above, it is clear the importance of signal processing in the context of visualization. However, the use of filtering mechanism as a resource to reduce noise and emphasize important structures present in the data has been little explored, mainly in information visualization applications. The present dissertation comes to shed light on this issue, proposing a methodology that enables the use of graph filtering mechanisms to leverage more informative visualizations.

3.2 Graph Spectral Filtering in Multidimensional Data

The GSP theory described in Chapter 2 is designed to operate in scalar signals defined on the vertices of a graph. Our goal, though, is to employ spectral filtering to sift high-dimensional data, cleaning noise and emphasize structures towards producing better visualizations.

The definition of a graph structure is the first step to adapt GSP to operate in high-dimensional data. We adopt the simple approach of defining the graph G as the k -nearest graph (KNN-graph) of the data. As discussed in Section 3.4, more sophisticated alternatives can be employed to build a graph from high-dimensional data, however, our methodology already presents quite satisfactory results with the KNN-graph.

Let $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$ be a set of points in an m -dimensional space, $\tau_i \in \mathbb{R}^m, i = 1, \dots, n$ and G be the KNN-graph derived from $\mathcal{T}$. From now on we will not make any distinction between a data instance τ_i and its corresponding node in the KNN-graph, using the same symbol for both.

We denote by τ^j the n -dimensional array containing the j -th attribute (coordinate) of all instances in $\mathcal{T}$. The array τ^j can be seen as a scalar function (signal) $\tau^j : V \rightarrow \mathbb{R}$ that assigns to each node τ_i the value of its j -th attribute τ_i^j . Since τ^j is a signal defined on the nodes of G , we can apply graph spectral filtering to process τ^j . In other words, given a spectral filter $\hat{h}$, we can compute the filtered version $\tilde{\tau}^j$ of the j -th attribute of the data by replacing f to τ^j in Equation (2.12). Repeating the process to all attributes, we can filter the whole data set. In summary, our approach is made up of three main steps: 1) build the KNN-graph from $\mathcal{T}$, 2) convert each attribute to a signal defined on the nodes of the KNN-graph, 3) filter the signals (attributes) based on a given spectral filter $\hat{h}$.

Figure 6 illustrates the proposed methodology and provides some intuition about the

behavior of the filtering process when applied to data attributes. The point cloud depicted in Figure 6a was generated by drawn 40 points from three bivariate Gaussians centered at the points $(0, 1)$, $(1, 1)$, $(2, 1)$, all with covariance matrix given by σI , where I is the 2×2 identity matrix and $\sigma = 0.1$. Figure 6b shows the k -nearest neighbor graph of the points with $k = 5$. Figures 6c and 6d show the distribution of points after applying a low-pass filter to the signals corresponding to the x and y coordinates, respectively. The low-pass filter smooths out the signals, making the points more tightly grouped. Moreover, the filter operates based on the graph topology, smoothing more stringently the coordinates of the points that are more densely connected. This fact can be noticed in Figure 6c, where the low-pass filter squished horizontally each of the three groups that are more densely connected, making them better delineated. A similar behaviour is observed in Figure 6d, where the y coordinate is smoothed out. Figure 6e shows the configuration of the points after filtering both the $\tilde{x}$ and $\tilde{y}$ coordinates. The presence of three distinct groups of points is clearly discernible in Figure 6e, which is not the case in original layout (Figure 6a).

Another interesting outcome of the filtering scheme is the better understanding of how each attribute is distributed. Figure 6f depicts histograms of the x (left column) and y (right column) coordinates before (top row) and after (bottom row) the graph-based low-pass filtering. Comparing the histograms of the x coordinate before (top left) and after (bottom left) the filtering process one can clearly see that the x coordinates become mostly concentrated in three specific values after filtering (bottom left), indicating the presence of three groups of points. The three groups are not so discernible in the original data histogram (top left).

It is clear from the construction above that the proposed methodology handles each attribute as an independent signal. Since each attribute is seen as a signal, why not to resort the classical signal processing theory rather than apply to the more intricate and costly framework proposed above? The answer for this question is simple: to handle an attribute as a “conventional” signal we have to set a domain for it to live in. There is no natural way to define a domain for an attribute. Suppose for example we want to handle an attribute τ^j as an one-dimensional signal. In this case we have to define the pairs (x_i, τ_i^j) in order to process τ^j as a function. However, fixing a sequence $\{x_1, \dots, x_n\}$ means to impose an order to τ_i^j that in fact doesn't exist.

The top image in Figure 7 depicts the x coordinates of the points in Figure 6a as an one-dimensional signal were the domain is given by the index of the corresponding points. If we chose another order for the nodes, the shape of the signal changes completely, as illustrated at the bottom image in Figure 7. Therefore, unless one can come up with a good strategy to define a domain for the attribute signals, it doesn't make sense to filter them. In contrast, the KNN-graph used in the proposed formulation comprises a natural domain where the attribute signals can be defined.

In the following we analyze the impact of the proposed filtering technique in different visualization scenarios.

Figure 6 – a) Points drawn from three bivariate Gaussians centered at $(0, 1)$, $(1, 1)$, $(2, 1)$ and constant diagonal covariance matrix; b) k -nearest neighbor graph with $k = 5$; c) and d): smoothing out the x and y coordinates respectively by a graph spectral low-pass filtering; e): configuration of points after filtering both the x and y coordinates; f): distribution of the x and y coordinates (left and right columns respectively) before and after the filtering process (top and bottom rows respectively).

Figure 7 – Graph attributes represented as a one-dimensional signal.

3.3 Results

In this section we describe a set of experiments that shows the usefulness of the proposed filtering scheme to improve the quality of visualizations. More specifically, we apply our methodology to high-dimensional data visualization in three different scenarios: scatter plot, parallel coordinates and multidimensional projection. In order to assess the impact of the graph filtering scheme in the visualizations we employ quantitative measures designed to assess the quality of scatter plot, parallel coordinates and multidimensional projection layouts. Before describing and discussing the results, we present the datasets used in our experiments.

3.3.1 Data Sets

Four different data sets are used in the experiments, a synthetic dataset (Blobs), the Ecoli and Wine data sets from the UCI Repository ([LICHMAN, 2013](#)), and a proprietary data set (Eggs) containing instances with 5000 attributes which correspond to the output layer of an Encoder-Decoder Neural Network ([VINCENT *et al.*, 2010](#)) applied to images of helminth eggs. All datasets have labeled instances and their number of instances, classes, and dimensionality are detailed in Table 1. The synthetic dataset was generated using the scikit-learn ([PEDREGOSA *et al.*, 2011](#)) method `make_blobs` with three classes and standard deviation equal to 2.3. Only scalar attributes from the dataset Ecoli are used (categorical attributes are disregarded).

3.3.2 Scatter Plot Matrix

Figure 8 shows the scatter plot matrix generated from the synthetic dataset. Figures 9 and 10 are the scatter plot matrices of filtered versions of the synthetic dataset with a low-pass

Table 1 – Data sets used in the experiments.

Name	Size	Dim	# of classes	Source
Blobs	200	5	3	synthetic
Ecoli	336	5	8	(LICHMAN, 2013)
Wine	178	13	3	(LICHMAN, 2013)
Eggs	320	5000	8	proprietary

Figure 8 – Scatter Plot Matrix of the synthetic dataset: original data.

Figure 9 – Scatter Plot Matrix of the synthetic dataset: low-pass filtered data.

and an enhancement graph filter, respectively, applied to the data.

Consider the upper left 2×2 sub-matrices corresponding to the first two attributes. Analyzing the histograms of the filtered sub-matrices (Figures 9 and 10) one can easily see that the distribution of the first and second attributes are concentrated in three intervals, indicating the existence of three well defined groups. The identification of the three groups is not so evident in the histograms of the original data (top image). Moreover, the correlation between the two attributes is more clearly revealed in the filtered scatter plot and based on the histograms users could infer the presence of three groups in the correlated attributes. Regarding the lower right 3×3 sub-matrix formed by the three last attributes, the histograms do not indicate those attributes are segregating the data into groups. However, the three groups are well separated when filtering

Figure 10 – Scatter Plot Matrix of the synthetic dataset: enhancement filtered data.

Figure 11 – Average quality measure of the synthetic dataset scatter plots depicted in Figures 8 to 10. The larger the averages the better.

is used to generate the scatter plots, mainly in scatter plots corresponding to the enhancement filter. The three groups are noway obvious in the scatter plots of raw data.

Quantitative evaluation

In order to quantitatively attest the better quality of the visualizations generated from the filtered data we compare the layouts using four different metrics widely employed to gauge the quality of scatter plots, namely, silhouette, homogeneity, completeness and adjusted rand index. The silhouette metric (ROUSSEEUV, 1987) accounts for both the cohesion and separation of grouped instances. Homogeneity, completeness (ROSENBERG; HIRSCHBERG, 2007) and adjusted rand index (HUBERT; ARABIE, 1985) compare the ground truth labels against the ones obtained by clustering the projected points in the 2D layout. More specifically, completeness gauges if members of a given class are contained in only one cluster. Homogeneity measures if all clusters contain only data points which are members of a single class. Adjusted rand index (adjusted_rand) measures the similarity between the ground truth labels and the clusterized groups. The larger the value of the metrics, the better the quality of the layout. The affinity propagation method (FREY; DUECK, 2007) was used as clustering method as it does not require the number of clusters as input parameter.

The result of applying the metrics described above to the scatter plots depicted in Figures 8 to 10 is presented in Figure 11. The values correspond to the average score of each metric over all layout of the scatter plots matrix. Notice that scatter plots built from filtered data are better for all the metrics and for two of them (completeness and adjusted_rand) the improvement is considerable.

(a) Original dataset

(b) Low-pass filtered data

Figure 12 – Parallel Coordinates view of the wine dataset with over-plotting optimal axes arrangement.

3.3.2.1 Parallel Coordinates

The second set experiments analyze the impact of graph filtering in parallel coordinates visualization. The top images in Figures 12 and 13 show parallel coordinates visualizations of the wine dataset using *over-plotting* and *parallelism* metrics (see details below) to find the optimal order of the axes. The bottom images in Figure 12 and Figure 13 are parallel coordinates visualizations of low-pass and enhancement filtered versions of the data also using over-plotting and parallelism metrics, respectively, optimize the order of the axes. It is clear the visual improvement of the layouts when filtered data is used to build the visualizations. In this particular application, low-pass filter leads to pleasant layouts with are very well defined groups.

(a) Original dataset

(b) Enhancement filtered data

Figure 13 – Parallel Coordinates view of the wine dataset with parallelism optimal axes arrangement.

Quantitative evaluation

We rely on the methodology proposed by [Dasgupta and Kosara \(2010\)](#) to measure the quality of parallel coordinates layouts. In short words, the reasoning is to define a metric to evaluate the quality between every pair of axes in the layout and apply a branch-and-bound optimization to rearrange the axes so as to optimize the metric. Once an optimal arrangement is reached, the metric can once again be applied to quantify the quality of the layout. In our experiments we assess the layout with four distinct quality metrics, line crossings, over-plotting, median angle, and parallelism. Line crossings ($l_crossing$) measures the number of intersections between line segments in the layout, median angle accounts for the median crossing angle of the line segments, over-plotting indicates the information loss due to limited number of screen pixels, and parallelism (p_norm) measures how parallel the line segments are. Except for the

Figure 14 – Parallel coordinates quality measures for the original (green), low-pass filtered (purple), and enhancement filtered (orange) data. The smaller the values of $l_crossing$, median angle, and over-plotting the better. The larger the p_norm the better. Each dot corresponds to the quality between a pair of axes in the layout.

Figure 15 – Lamp projection of the Ecoli dataset. Left: original data. Right: enhanced filtered data.

parallelism metric, smaller values are better. More details about the mathematical definitions and computational aspects of the metrics can be found in the work by [Dasgupta and Kosara \(2010\)](#).

Figure 14 depicts the result of applying the metrics to the “optimum” parallel coordinates layouts between each pair of axes. Notice that for $l_crossing$, over-plotting, and median_angles, the metric’s values (each dot is a quality measurement made between a pair of axes) corresponding to the layouts produced from filtered data are more concentrated in the lower part of the plot while for the parallelism metric the values are on the higher part, indicating the superior quality of the layouts generated from filtered data.

Figure 16 – a) Lamp projection of the Eggs dataset using the original (left) and low-pass filtered (right) data. b) PCA projection of the Eggs dataset using the original (left) and low-pass filtered (right) data.

3.3.2.2 Multidimensional Projections

The last set of experiments assess the impact of the proposed filtering methodology in layouts produced by multidimensional projections. We employ the LAMP technique (JOIA *et al.*, 2011) and PCA (HOTELLING, 1933) as projection methods. We choose LAMP and PCA because they are well known projection methods widely used in visualization applications, but any other projection technique could be chosen. Two datasets are used in the experiments, the Ecoli and the Eggs datasets (see Table 1).

Figure 15 shows the result of projecting the original Ecoli data (left) and its enhancement filtered version (right) using LAMP. Comparing both projections is easy to see that groups are better defined when filtered data is used as input for the projection. In particular, notice the good separability of the pink and gray classes in the filtered projection. Even the blue class that is completely entangled with the purple class in the original plot becomes better concentrated in

the layout with filtered data.

Figure 16 shows the result of visualizing the Eggs data set using LAMP and PCA. Eggs is a complex data set with instances embedded in a space with 5000 dimensions, what makes projections based on distance information, as is the case of LAMP, more prone to distortions (a consequence of the curse of dimensionality). Large dimensional data is typically handled via PCA. Notice how better defined are the groups in both LAMP and PCA layouts when filtered data is used as input for the projections. In particular, with exception of the dark blue and green groups, the low-pass filter has untangled all the other groups pretty well, mainly in the LAMP layout (Figure 16a left), an impressive result when we compare to the original filtered data layout.

Quantitative evaluation

The quality of the multidimensional projection layouts are assessed using the same metrics employed to evaluate scatter plot layouts (Figure 16). The results are summarized in Figure 17. Once again, layouts produced from filtered data present better quality for all the metrics.

3.4 Discussion and Limitations

Results presented in Section 3.3 attest the benefits of the proposed filtering schemes when used in combination with visualization methods. The provided qualitative and quantitative results shows that filtering mechanisms can be primordial in the context of visualization.

It is important to make clear we are not advocating that filtered data should replace original data in all visualizations. Rather, we encourage the concomitant use of “raw” and filtered data in visualizations. A good example is the synthetic dataset visualization depicted in Figures 8 to 10, where the plots from raw and filtered data bring complementary information. In fact, scatter plots of the original data discriminate the groups quite well as to the first two attributes, while the histograms from filtered data shows precisely where those attributes are more concentrated.

We consider the proposed methodology just a first step towards incorporating signal processing tools into information visualization methods, as a multitude of possibilities are still to be explored. For instance, how to design optimal filters for a particular application? Which filter is more appropriate for a given visualization method? Is it possible to learn optimal filters from existing visualizations? Answering those questions is not straightforward, demanding a substantial amount of investigation as future work.

There are, though, some pitfalls of our methodology, as for example the proper definition of the graph structure from which the whole graph signal analysis is derived. As we discussed in Section 3.2, the topology of the graph impacts directly in the result of the spectral filtering

(a) Ecoli

(b) Eggs

Figure 17 – Quality measure of projection layouts using the original (green), low-pass (purple), and enhancement filtered (orange) data. a) Metrics applied to LAMP (left) and PCA (right) projections of the Ecoli dataset; b) Metrics applied to LAMP (left) and PCA (right) projections of the Eggs dataset.

process. A graph with a dense set of edges connecting unrelated nodes can lead to unsatisfactory results. This problem raises the issue of how to generate graphs that are more appropriate than the KNN-graph used in our methodology. An option could be to adapt graph learning methods (DONG *et al.*, 2016) to the context of GSP-assisted data visualization.

Another import aspect in the present context is that GSP is not the only alternative to handle data towards improving visualizations. Other schemes such as spatial and statistical filtering, widely used in graphics and computer vision, could also be adapted to visualization purposes.

3.5 Final Considerations

In this chapter we introduced a new filtering methodology based on graph signal processing theory, which turns out to be useful to improve the quality of visualizations. The provided qualitative and quantitative results attest the positive impact of the proposed filtering method

to generate high quality visualizations. We believe that present work is just a first step towards making filtering mechanisms a basic tool in the context of visualizations, opening a new avenue for further developments.

GRAPH WAVELET TRANSFORM

4.1 Graph Wavelets review

Classic Wavelets theory was extended to graphs, allowing to perform localized analysis of graph functions. [Shuman *et al.* \(2013\)](#) classifies graph Wavelets methods into: *node domain* and *graph spectral* designs.

Node domain designs

[Crovella and Kolaczyk \(2003\)](#) defined Wavelet like functions $\psi_{m,\tau_i} : V \rightarrow \mathbb{R}$, which are localized on node τ_i and scale m , as dilations and translations of a mother function ψ . The restriction $\sum_{\tau_i \in V} \psi_{m,\tau_i} = 0$ is imposed on such functions, so that the resulting Wavelets have zero mean. Thus, ψ_{m,τ_i} is defined as:

$$\psi_{m,i}(\tau_j) = \frac{\alpha_{\tau_i,k}}{|\partial \mathcal{N}(\tau_i, k)|}, \quad \forall \tau_j \in \partial \mathcal{N}(\tau_i, k),$$

where $\sum_{k=0}^s \alpha_{\tau_i,k} = 0$ and $\partial \mathcal{N}(\tau_i, k)$ is the subset of nodes which are k nodes away from τ_i .

Constants $\alpha_{\tau_i,k}$ are computed from a continuous function $\psi(\cdot)$, supported on $[0, 1]$, by taking $\alpha_{\tau_i,k}$ to be the average of $\psi(\cdot)$ on the interval $I_{m,k} = [\frac{k}{m+1}, \frac{k+1}{m+1}]$. The localized functions ψ_{m,τ_i} provide a multi-scale vision of the graph function. However, they are not invertible, which is a desirable property in many signal processing applications.

Variations of the lifting scheme, used in traditional wavelets theory, were proposed for graphs ([NARANG; ORTEGA, 2009](#); [WAGNER; DELOUILLE; BARANIUK, 2006](#); [SHEN; ORTEGA, 2010](#)). In this approach, the node set is divided into two disjoint subsets $V = V_I \cup V_J$. At each lifting step, nodes from V_I use their own values and values from their neighbors in V_J to *predict* their coefficients. Then, using this predicted values along with its own values, nodes from

V_J update their coefficients. The subset V_J is the input for the next lifting step. Jansen, Nason and Silverman (2009) proposed an iterative lifting scheme computing one coefficient at a time.

Graph spectral domain designs

Coifman and Maggioni (2006) introduced a method for building a set of orthonormal scaling functions and wavelets associated to them (*diffusion Wavelets*). The scaling basis is constructed as an approximation of a given diffusion operator that models the graph topology at a specified scale. For each scale, a wavelet function is built by finding the orthogonal complement to the corresponding scale space. The orthogonalization process is done using the Gram-Schmidt procedure which is expensive and also obscures the relation between the diffusion operator and Wavelets functions.

Hammond, Vandergheynst and Gribonval (2011) proposed to use kernel filters (dictionaries) on the graph spectral domain to define the Graph Wavelets Transform. They introduced the Spectral Graph Wavelets (SGW) based on dilations of a polynomial function. Later, other dictionaries, kernel filters in the spectral domain, to perform the Wavelet Transform on graphs were proposed (NARANG; ORTEGA, 2012; LEONARDI; Van De Ville, 2011; LEONARDI; Van De Ville, 2013; SHUMAN *et al.*, 2015). Narang and Ortega (2012) proposed two-channel filter banks on bipartite graphs. The resulting Wavelets functions are orthogonal and critically sampled allowing for perfect reconstruction. However, this transform is only possible for bipartite graphs. Shuman *et al.* (2015) proposed Spectrum-Adapted Graph Wavelets (SAGW) which is a dictionary defined as warped versions of uniformly translated kernels. SGW and SAGW are further discussed in Section 4.2.

4.2 Spectral Graph Wavelet Transform

A dictionary $\{\hat{g}_m\}_{m=1,2,\dots,M}$ is defined as a set of graph spectral filters $\hat{g}_m : \Lambda \rightarrow \mathbb{R}$, $m = 1, 2, \dots, M$, where M is the number of kernels in the dictionary. Recalling that the eigenvalues λ_j correspond to frequencies in the graph Fourier domain, we can define filters by properly handling the eigenvalues λ_j according to scales m . The Spectral Graph Wavelet Transform, from now on referred as *Graph Wavelet Transform* (GWT), (Equation (4.1)) relies on a dictionary of one scaling kernel $\{\hat{g}_1\}$ and $M - 1$ wavelet kernels $\{\hat{g}_m\}_{m=2,\dots,M}$ defined in the spectral domain. Given a signal f and a dictionary $\{\hat{g}_m\}_{m=1,2,\dots,M}$, the GWT $W_f : \{1, \dots, M\} \times V \rightarrow \mathbb{R}$ is defined as:

$$W_f(m, \tau_j) = \sum_{\ell=1}^n \hat{g}_m(\lambda_\ell) \hat{f}(\lambda_\ell) u_\ell(\tau_j), \quad (4.1)$$

which is the graph filtering operator defined in Section 2.3.

By varying m while keeping τ_j fixed, we obtain M coefficients associated to τ_j : one scaling coefficient and $M - 1$ wavelet coefficients. Moreover, the product $\hat{g}_m(\lambda_\ell) \hat{f}(\lambda_\ell)$, on the

Figure 18 – **Signal:** f_0 signal defined on a path graph with 64 nodes. Signal f_0 is a combination of low and high frequency signals. The height of the vertical blue lines and the color of the nodes correspond to the amplitude of the signal. **SGW coefficients:** Wavelet coefficients of all nodes for signal f_0 . Each column represents a node, while each row represents a scale of the wavelets coefficients from low (top of the matrix) to high (bottom of the matrix) frequencies. Coefficient values are encoded using color. The order of the columns is the same as the order of the nodes from the path graph shown above.

right of Equation 4.1, shows that each coefficient $W_f(m, \tau_j)$ is obtained by modulating $\hat{f}(\lambda_\ell)$ by the kernel $\hat{g}_m(\lambda_\ell)$. The rationale behind Equation (4.1) is that for large values of m , large scales, the filter g stretches out favoring high frequency modes essential to good localization. Small values of m (small scales) compress the function around low frequency modes to encode coarser description of a local neighborhood.

To illustrate this fact, consider the same signal f_0 used in Chapter 2, defined as a combination of low and high frequency eigenvectors. The heatmap on bottom of Figure 18 depicts the wavelets coefficients of all nodes, where each column is associated to a node and each row represents a scale m . Rows on the heatmap are ordered from low to high frequencies and columns have the same order as in the path graph shown above. The coefficient corresponding to the first low frequency for node n_4 is larger than any other coefficients, reflecting the fact that the signal in node n_4 is smooth. In contrast, larger coefficients appear in high frequencies, indicating that the signal oscillates more abruptly around node n_{42} . While node n_{31} , where the signal is close to smooth and abrupt oscillations, has coefficients of low value in all scales.

Many ways to define dictionaries for the GWT have been proposed in the literature (LEONARDI; Van De Ville, 2011; LEONARDI; Van De Ville, 2013; SHUMAN *et al.*, 2015). The *Spectral Graph Wavelets (SGW)*, proposed by Hammond, Vandergheynst and Gribonval (2011), are defined as a dictionary composed of a low-pass filter (scaling kernel) and dilations of a band-pass filter (wavelet kernels). In mathematical terms, the wavelet kernels are defined using the mother kernel:

$$\hat{g}(x) = \begin{cases} x_1^{-\alpha} x^\alpha & \text{for } x < x_1 \\ p(x) & \text{for } x_1 \leq x \leq x_2 \\ x_2^\beta x^{-\beta} & \text{for } x > x_2 \end{cases} \quad (4.2)$$

Figure 19 – Filter design: Graph spectral filters for the SGW and SAGW dictionaries. The SAGW filters adapt to regions with larger spectral density. The x marks on the horizontal axis represent the graph Laplacian spectrum of a path graph (left) and a comet graph (right).

where α, β, x_1, x_2 are parameters of the filter that can be tuned to change the behavior of $\hat{g}$. Function $p(\cdot)$ is a cubic polynomial satisfying $p(x_1) = p(x_2) = 1$, $p'(x_1) = \alpha/x_1$, and $p'(x_2) = -\beta/x_2$. The $M - 1$ wavelet kernels are obtained by scaling the mother kernel: $\hat{g}_m(\lambda_\ell)_{m=2,\dots,M} = \hat{g}(m\lambda_\ell)$. The scales are logarithmically sampled between $m_{\min} = m_2, \dots, m_M = m_{\max}$, where $m_{\min} = x_2/\lambda_{\max}$, $\lambda_{\max}$ is the largest frequency in the spectrum, and $m_{\max} = 20x_2/\lambda_{\max}$.

The scaling kernel is defined as $\hat{g}_1(x) = \gamma \left(-\exp\left(\frac{x}{0.6\lambda_{\min}}\right)^4 \right)$ where the parameter γ is chosen such that $\hat{g}_1(0)$ is equal to the maximum value of g . Left column of Figure 19 illustrates the SGW with 6 distinct scales for the wavelet kernels and the scaling kernel, computed as described above.

In the example in Figure 18 the wavelet coefficients were obtained using the SGW for a path graph. In this case the graph defining the signal domain is essentially the same as a discretization of the real line, that is, the graph topology is fairly regular. Thus, the strategy employed by SGW of evenly cover the spectral domain considering only $\lambda_{\max}$ will result in a dictionary that represents well the graph frequencies, as shown in left of Figure 19.

However, in scenarios where the graph is irregular the eigenvalues are not evenly distributed. For instance, the x marks on the horizontal axis on the plots on the right of Figure 19 represent the spectrum for a comet graph (like the one shown in Figure 20). Using the SGW will result in a dictionary with unequal number of eigenvalues in each frequency band (top-right of Figure 19). Shuman *et al.* (2015) addressed this problem by proposing the *Spectrum-Adapted Graph Wavelets* (SAGW). The bottom-right of Figure 19 shows the SAGW dictionary defined in the spectral domain of a comet graph. As one can clearly see, when eigenvalues are not evenly distributed in the spectral domain, as is the case of the comet graph, the SAGW dictionary better fits the frequency bands.

The SAGW wavelet kernels are defined as: $\hat{g}_m(\lambda_\ell)_{m=2,\dots,M} = \hat{g}(\omega(\lambda_\ell))$, where ω is a warping function that approximates the cumulative spectral density function. And the scaling

Figure 20 – **Top:** Second largest graph Laplacian eigenvector (u_{62}) defined on a comet graph with 64 nodes. The color associated to the graph nodes encodes the signal. **SGW coefficients:** Spectral Graph Wavelet coefficients of all nodes for signal u_{62} . Each column represents a node, while each row represents a scale of the wavelets coefficients from low (top of the matrix) to high frequencies (bottom of the matrix). Coefficient values are encoded using color. **SAGW coefficients:** Spectral Adapted Graph Wavelet coefficients of all nodes for signal u_{62} .

kernel: $\widehat{g}_1(x) = \sqrt{Ra_0^2 + R/2 \sum_{k=1}^K - \sum_{m=2}^M |\widehat{g}_m(\ell)|^2}$, where R , M and K are any integers such that $2 < R \leq M$ and $K < R/2$, and $a_k \in \mathbb{R}$ are coefficients satisfying $\sum_{k=0}^K (-1)^k a_k = 0$. The SAGW dictionary forms a tight frame that allows interpreting the squared magnitudes of the coefficients as an energy density function (SHUMAN *et al.*, 2015), making the understanding of wavelets coefficients an easier task.

In order to further illustrate this point, we created a comet graph with 64 nodes with the star extremity containing 30 nodes. The signal is defined as the second largest graph Laplacian eigenvector (Figure 20 top). The wavelet coefficients, computed via SGW and SAGW, are also illustrated at the bottom of Figure 20. The SGW dictionary of node n_{47} led to wavelet coefficients with dominant frequencies in the central scales, not properly characterizing the high frequency nature of the signal under analysis. In contrast, SAGW dictionary resulted in wavelet coefficients with larger intensity in the scales related to high frequencies, better capturing the pattern of variation of the signal.

SPATIOTEMPORAL ANALYSIS AND VISUALIZATION ON GRAPHS

Visual analytics stands as a cooperative data analysis paradigm that aims to find effective ways of combining analytical power from humans and computers (KEIM *et al.*, 2008).

When data include information about location and time, it is usually called *spatiotemporal data*. Spatiotemporal data may be represented as *spatial time series*. Analysis of spatial time series is not always straightforward. For instance, identifying regions and time intervals of abrupt and/or smooth variation on spatial time series, while visually tracking the evolution of those patterns over time, is not an easy task. A sudden temporal variation can indicate the appearance of an unexpected event or reveal a change in the behavior of some phenomenon. Furthermore, analyzing variations over time helps in understanding its temporal evolution. Therefore, providing users with effective tools for analyzing spatial time series is a challenge.

The literature about time-varying data visualization is extensive and comprehensive surveys can be found in the works by (AIGNER *et al.*, 2007; AIGNER *et al.*, 2011), which develop a systematic review on time-varying data visualization. Another review presented by (BACH *et al.*, 2014) classifies time-varying data visualizations based on space-time cube operations. The related problem of movement visualization was also extensively reviewed (DEMSAR *et al.*, 2015; ANDRIENKO; ANDRIENKO, 2012).

5.1 Visualization of spatial time-series

5.1.1 Integrated views

Integrated views attempt to reduce cognitive load on users by presenting the information in a single view. When visualizing spatial time series, time series glyphs (FUCHS *et al.*, 2013) can be plotted on top of a spatial map. To avoid clutter, such glyphs must be capable of encoding

Figure 21 – Time series integrated views in 2D. **a** Two-tone pseudo coloring technique (SAITO *et al.*, 2005). **b** Value flow maps (ANDRIENKO; ANDRIENKO, 2004). **c** Time-oriented polygons (SHANBHAG; RHEINGANS; DESJARDINS, 2005). **d** Growth Ring Maps (BAK *et al.*, 2009; ANDRIENKO *et al.*, 2011). **e** Storygraphs (SHRESTHA *et al.*, 2013).

large time series in a space-efficient way (WICKHAM *et al.*, 2012). For instance, Saito *et al.* (2005) encode temperature time series using the two-tone pseudo coloring technique, resulting in a compact glyph which is later plotted over a geographic map, as depicted in Figure 21a. In a similar manner, *Value Flow Maps* (ANDRIENKO; ANDRIENKO, 2004), representing the temporal behavior of an area, were plotted over its corresponding area on a cartographic map (Figure 21b).

Another possibility is to use the entire spatial area associated to the time series to display a time graph. Such approach was called *time-oriented polygons* by (SHANBHAG; RHEINGANS; DESJARDINS, 2005). An example of the representation of an increasing and decreasing time series is shown in Figure 21c.

Pixel oriented graphs (KEIM, 2000) can also be plotted over a map to display spatial time series. (BAK *et al.*, 2009) propose to represent every spatiotemporal event by one pixel placed in an orbital way. Colors can be used to represent categories or periods of time. Such representation is called *Growth Rings* and has been used by (ANDRIENKO *et al.*, 2011) to display the number of photos taken at different locations (Figure 21d). The advantage of pixel oriented graphs over glyphs and time-oriented polygons is that they are highly scalable for large time series. However, when dealing with spatially dense time series, spatial clutter is a drawback for both approaches.

A quite different approach, illustrated in Figure 21e, is proposed by (SHRESTHA *et al.*, 2013). There, a graph with two vertical axes is used to represent spatial dimensions, e.g. latitude and longitude, while time is represented in the horizontal axis. So, each latitude-longitude coordinate is represented by a line connecting its values at each respective axis. Temporal events occurring at certain coordinate are plotted through its corresponding line resulting in the so called *Storygraphs*. Those graphs can be augmented with frequency information reflected as color intensity (LI; GOEBEL; SJOBERGH, 2014). While this approach can display large amounts of spatiotemporal data, spatial proximity of events can go unnoticed.

Displaying temporal information over a spatial map has also been approached using 3D views. The *space-time cube metaphor* considers time as an additional dimension for plotting spatiotemporal data. Including time as a third dimension was first proposed by (HÄGER-STRAND, 2005). This approach was then extensively used to visualize spatiotemporal data, such as earthquakes (GATALSKY; ANDRIENKO; ANDRIENKO, 2004), precipitating clouds (TURDUKULOV; KRAAK; BLOK, 2007), space-time paths (LENNTORP, 1976; KRAAK, 2003), stock movements (DWYER; GALLAGHER, 2004), among others.

(TOMINSKI; SCHULZE-WOLLGAST; SCHUMANN, 2005) display time series as 3D icons on a map (Figure 22a). To avoid cluttering, they allow users to hide information by filtering attributes on each time slice. Colors are used to display the maximum value at each time, which they call maxima-events. Time series are also represented as 3D columns called *Data Vases* (BEBIS *et al.*, ; THAKUR; HANSON, 2010), where time slices are considered to have a fixed height while the column width represents some attribute or statistical property. This

Figure 22 – Integrated views in 3D. **a** 3D icons on a map (TOMINSKI; SCHULZE-WOLLGAST; SCHUMANN, 2005). **b** Data Vases (BEBIS *et al.*,). **c** Isosurfaces, constrained isosurfaces and 3D wall maps, respectively (CHENG *et al.*, 2013).

approach is illustrated in Figure 22b. Manipulation of the 3D view and interactive techniques, such as brushing and filtering, are allowed. Besides, aggregation is used as a data transformation method to alleviate cluttering.

From data obtained at traffic sensors, (CHENG *et al.*, 2013) used the time-space cube metaphor to visualize traffic congestion. Isosurfaces, isosurfaces constrained to the road network and 3D wall maps are constructed from spatial time series. An example of each of these appears in Figure 22c. In addition to visual elements, no further interaction is addressed in such work.

Data occlusion and spatial cluttering, along with 3D interaction difficulties are some of the common drawbacks of techniques using the space-time cube metaphor. Besides, these techniques usually need to be combined with more specialized data analysis methods to allow users to make sense of data.

5.1.2 Multiple views

Each time slice of spatiotemporal data can be represented by a spatial graph, and several of such spatial graphs can be displayed together in a *Small Multiples* fashion (TUFTE, 1990).

Figure 23 – Multiple views. **a** Small multiples (PARK; QUEALY, 2012). **b** Coordinated multiple views: Taxis (FERREIRA *et al.*, 2013) and GeoAnalytics (JERN; FRANZEN, 2006), respectively.

Small multiples have been used to visualize annual drought in the U.S. (PARK; QUEALY, 2012), monthly drought in the state of California (KYLE; LAUDER, 2015) or approved harvests per year (PETERSON, 2011). This approach is exemplified in Figure 23a.

Presenting multiple spatial graphs at the same time allows for identifying the global evolution of temporal data and for detecting patterns. However, it may be difficult to make local comparisons. Moreover, only a short amount of graphs can be displayed at the same time, which makes this approach unsuitable for representing large spatial time series.

Coordinated multiple views (ROBERTS, 2007; ANDRIENKO; ANDRIENKO, 2007) have also been used for spatial time series visualizations. The majority of methods using this approach, e.g., (JERN; FRANZEN, 2006; MACIEJEWSKI *et al.*, 2010; FERREIRA *et al.*, 2013), present a spatial graph as the main view, as illustrated in Figure 23b. Such spatial graph is usually a heat map or a density heat map. The main view is linked with other views, such as time series graphs, parallel coordinates, histograms, among others.

Frameworks using coordinated multiple views can benefit from the properties of each

linked view while avoiding clutter. Nevertheless, having more views reduces the computational performance and usually implies a higher cognitive load for users.

5.2 Visual analytics of spatiotemporal data

[Demsar, Fotheringham and Charlton \(2008\)](#) propose to use geographically weighted regression to identify spatiotemporal patterns in spatial time series. They build a dictionary consisting of different spatiotemporal configurations and then represent each element on the spatial time series using this dictionary. That is, each element is associated to a series of coefficients encoding specific spatiotemporal behaviors. Such coefficients are later grouped using a *Self Organizing Map* (SOM) ([KOHONEN, 1995](#)), which also serves as a mean of visually exploring the results. The SOM view is linked with a spatial view and a time series graph to allow the exploration of the data. Nevertheless, the spatiotemporal patterns detected are limited by the configuration of the dictionary.

SOMs have also been used by [Andrienko et al. \(2010\)](#) to cluster spatial configurations at different time slices (spatial-in-time), and temporal profiles at different locations (time-in-space). When displaying the resulting SOM matrix, similar cells have similar color. That view is linked with other views allowing the discovery of periodic events, anomalous situations and other patterns. However, since temporal or spatial information is treated globally, discovering local variations, temporal or spatial, is impossible.

[Köthur et al. \(2014\)](#) group time slices using a hierarchical clustering approach. The resulting hierarchy is visualized and used to explore the data, along with other linked views. The resulting groups only account for global affinity since the similarity measure used was the sum of squared errors between time slices.

A popular domain for time-varying information defined on graphs is in the study of urban data. [Ferreira et al. \(2013\)](#) use the start and end information of taxi trips to explore urban mobility and patterns, introducing an efficient storage manager to deal with large volume of data. The method is capable of handling different queries such as trips that begin on downtown Manhattan and end at the airports. However, the exploration of changes in the information is not fully developed, providing only a plot of total events and the juxtaposition of geographic renderings for different time slices. [Doraiswamy et al. \(2014\)](#) consider the same data as one of the case studies, proposing a method to support event-guided exploration of large urban data. The problem is approached through topological tools and is able to detect events of different scales and shapes. Detected events can be used as a parameter for querying, however, the method does not provide details on the temporal evolution of the information not detected as an event.

5.2.1 Mobility patterns

Wang *et al.* (2013) consider taxi tracking information that is filtered and matched to road networks. This information is used to calculate trajectories and create propagation graphs of the traffic information. The propagation graphs are used to show concise information about detected events, specifically how the traffic jams propagate. In contrast to our approach, their pixel-based layout does not provide an efficient way to observe patterns of change in the information. Zeng *et al.* (2013) study the visualization of interchange patterns between defined locations, introducing the concept of *interchange circos diagrams*. The method uses juxtaposition of diagrams to explore temporal changes. However, since the method is based on multi-scale information aggregation, users have to interactively explore the information throughout different scales, what can hinder the detection of small anomalies. Such remarks can also be made for the work proposed by Andrienko and Andrienko (2008), where data aggregation methods are used to explore trajectory-oriented and traffic-oriented views of movement data, going beyond the usual spatiotemporal aggregation schemes by considering direction and route information. Since they overlay the pixel-based traffic information into the map, only the coarse geographical information is kept during visualization.

Pu *et al.* (2014) present a system for visualizing mobility patterns based on phone call information. The visualization is based on a Voronoi diagram computed from cellphone tower location, allowing the analysis of people migration between different towers. The migration flow is depicted on the edges of the diagram. A similar approach was used by Sun *et al.* (2014) to analyze temporal information using roads of a map as a graph. The time-varying information is associated with the edges of the graph rather than the nodes. The main advantage of Pu and Sun methods is the ability to correlate spatial and temporal information. Such methods are not suitable, though, for large graphs. Handling levels of detail is not also a straightforward task.

A quite different approach is proposed by Chu *et al.* (2014), where geographic coordinates are transformed into street names, therefore the trajectory of a particular taxi can be expressed as a document and the dataset itself as a document corpora, allowing to process the data through nature language processing tools such as *Latent Dirichlet Allocation*. Data variations are expressed through changes in the “taxi topics” and depicted using a set of alternative plots. However, the visualization is focused on the taxi topic information, thus raw information such as the concentration of taxis in specific position and time can not be directly visualized.

Guo *et al.* (2011) introduce a visualization system called TripVista, aiming the exploration of microscopic traffic patterns and abnormal behaviors, such as traffic on a particular street junction. The method involves, along with other visualization techniques, a timeline visualization derived from Theme-River Havre, Hetzler and Nowell (2000), but enriched with directional information. While this approach is capable of identifying underlying patterns, outliers, and abrupt changes in the data, it is not directly scalable to macroscopic patterns and can not discriminate changes according to the frequency of the signal.

GRAPH WAVELETS ANALYSIS ON TIME-VARYING DATA

In this chapter we will show how visual analytic methodologies for the analysis of spatial time series can take advantage of the recent developments of graph Wavelet theory. In contrast to other methods that also make use of graph wavelets, our approach will be endowed with visual resources that will make the analysis of time-varying data an easy task, even for users not experienced with Wavelet theory.

The graph wavelet formulation presented in Chapter 4 was designed to process real functions with no temporal variation. A natural way to extend the theory to process time-varying data is through the so called Cartesian product graph.

Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two graphs and τ_i and ι_j be nodes in V_G and V_H , respectively. The Cartesian product between G and H is the graph $G \times H$ with node set $V_G \times V_H$ and edges connecting two nodes (τ_i, ι_j) and (τ_k, ι_l) if and only if either $\tau_i = \tau_k$ and ι_j is adjacent to ι_l in H , or $\iota_j = \iota_l$ and τ_i is adjacent to τ_k in G .

The particular case where H is a path graph is of special interest in our context, as the Cartesian product $G \times H$ can be seen as copies of G stacked according to the nodes of H , as illustrated in Figure 24.

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

The main advantage of extending a time-varying function as described above is that $f_{G \times H}$ is a “steady” function on the nodes of $G \times H$, therefore, the graph wavelet theory presented

Figure 24 – Cartesian product $G \times H$. Source: (VALDIVIA *et al.*, 2015).

in section 4.2 can be directly employed to analyze $f_{G \times H}$. However, the number of nodes in $G \times H$ increases as nr , thus, even for moderate values of n and r , the size of $G \times H$ and the corresponding Laplacian matrix can be considerable, hampering the computation of eigenvalues and eigenvectors needed for wavelets analysis. Nevertheless, the spectrum of Cartesian product graphs has the particular property of being derived from the spectrum of G and H , making unnecessary the construction of the Laplacian matrix associated to $G \times H$. Precisely, let u_j, λ_j be the eigenvectors and eigenvalues of G and $\mathbf{v}_k, \mu_k$ be the eigenvectors and eigenvalues of H , then $\lambda_j + \mu_k$ is an eigenvalue of $G \times H$ and $\mathbf{w}_{jk} = \mathbf{u}_j \otimes \mathbf{v}_k$ the corresponding eigenvector, where $\otimes$ is the Kronecker product Büyükoğlu, Leydold and Stadler (2007). Computing the spectrum of $G \times H$ from the eigenvalues and eigenvectors of G and H makes the use of graph wavelet theory feasible for handling large time-varying data.

6.1 Node classification and stacked graphs

Let $f_{G \times H} : V_G \times V_H \rightarrow \mathbb{R}$ be the extension of a time-varying function to a Cartesian product graph $G \times H$, as described above. We denote by $\omega_{f_{G \times H}}(s, i, j)$ the wavelet coefficient of $f_{G \times H}$ at scale s , and location (τ_i, ι_j) , as described in equation (4.2). If scales are discretized as $s_{min} = s_1, s_2, \dots, s_m = s_{max}$, then each node (τ_i, ι_j) of $G \times H$ can be associated to an m -dimensional feature vector with attributes given by the wavelet coefficients $w_{ij} = (\omega_{f_{G \times H}}(s_m, i, j), \dots, \omega_{f_{G \times H}}(s_1, i, j))$, where large scales are arranged on the left and small scales on the right of w_{ij} .

For small scales, wavelet coefficients tend to be high in regions of the graph where f varies abruptly. On the other hand, for large scales, wavelet coefficients with high magnitude tend to show up in regions where f has a “smoother” behavior. Therefore, each node of $G \times H$ can be characterized according to the distribution of wavelet coefficients in its corresponding feature vector. More precisely, if wavelets coefficients of a node (τ_i, ι_j) assume larger values in small scales (on the right part of w_{ij}) then that node is located in a region of abrupt variation of $f_{G \times H}$, and we will call (τ_i, ι_j) a *high frequency node*. If the coefficients are more concentrated in

Figure 25 – Four dimensional feature vectors characterizing: from top to bottom high, mid-high, mid-low, low and indeterminate frequency classes. Each row depicts ten examples of feature vectors in each training class and the bars correspond to the magnitude of each attribute. Source: (VALDIVIA *et al.*, 2015).

large scales (on the left part of w_{ij}) then the node is located in a region of smoother variation and it will be called *low frequency node*. In our experiments we have noticed that four scales are enough to reveal high and low frequency phenomena, being the number of scales employed in all results and case studies presented in this work.

Node classification As discussed above, each node (τ_i, ι_j) can be classified according to the distribution coefficients in its corresponding feature vector w_{ij} . Classifying each node according to the pattern of coefficients in w_{ij} makes easier the identification of high and low frequency regions in the graph while allowing the visualization of how those regions evolve over time.

Classification can be performed through pattern recognition mechanisms commonly employed by the machine learning community. In this work we choose the pattern recognition neural network (PRNN) classifier, proposed in Møller (1993), due to its reduced computational times and satisfactory performance in terms of classification. PRNN is a supervised classification method, thus demanding a training set in order to fit the classification model. Assuming that the feature vectors are four dimensional, that is, each w_{ij} is built from four scale levels, we generate a training set made up of five classes, as illustrated in Figure 25. Feature vectors in the low frequency class has $w_{ij}(1)$ as the dominant attribute (Figure 25, dark blue bars); $w_{ij}(2)$ is

Figure 26 – A function f in a specific time instant (a) and feature classification (b) characterizing regions of low to high and indeterminate frequencies. Source: (VALDIVIA *et al.*, 2015).

the dominant attribute in the mid-low frequency class (Figure 25, light blue bars); $w_{ij}(3)$ and $w_{ij}(4)$ are the dominant attributes in the mid-high (Figure 25, orange bars) and high frequency (Figure 25, red bars) classes, respectively. Feature vectors with no dominant attribute characterize the indeterminate class (Figure 25, yellow bars). We simulate feature vector patterns in each class to serve as training data for the classification method. The training set was generated with one thousand samples per class, thus comprising a data set with five thousand elements. Figure 25 illustrates ten samples from each class generated by our training data simulator. The simulator was designed to mimic coefficient patterns we observed in real and synthetic data sets.

Figure 26 shows a specific time-frame of a synthetic time-varying data (left) and regions of low, mid-low, mid-high, high and indeterminate frequencies (right) classified with the PRNN after a training step with simulated training data.

Stacked graph For each time t_i , we can count the number of nodes that belong to a specific class in that time. More specifically, let C be one of the five classes defined above and $(\tau_i, \iota_j)_C$ be a node in C . For each time-instant t_k , the set of nodes $C_{t_k} = \{(\tau_i, \iota_j)_C \mid j = k\}$, corresponds to the nodes in $V_G \times \iota_k$ that belongs to C . Therefore, we can apply a stacked graph metaphor to visualize the number of nodes in each class in the time t_i , $i = 1, \dots, m$, which allows to visualize how classes evolve over time. Figure 27 illustrates the proposed visualization. Each horizontal strip represents a class, colors discriminate the class types, and the thickness of each strip accounts for the number of nodes in each class in each time t_i . The highlighted instant in Figure 27 corresponds to the time slice depicted in Figure 26b.

As we show in the following sections, the stacked graph metaphor allows for easily identifying time intervals where a function f varies abruptly or presents low levels of variation, assisting in the analysis of f over time.

Figure 27 – Stacked graph illustrating how each of the five classes evolve over time. Source: (VALDIVIA *et al.*, 2015).

6.2 Computational times

The bottleneck of our algorithm is the calculation of wavelets coefficients (Eq. 4.2). The coefficients are computed from eigenvectors and eigenvalues of the graphs G and H . We use LAPACK to compute the spectrum of both graphs, where the routine used has overall complexity $O(n^3)$. Since the spectrum of $G \times H$ can be derived from the spectrum of graphs G and H using the Kronecker product and sum of eigenvalues, asymptotic complexity to compute all eigenvectors and eigenvalues of $G \times H$ is $O(n^2r^2 + n^3 + r^3)$, where n and r are the number of nodes in G and H respectively.

6.3 Visual elements and linked views

Figure 28 shows the interface of our prototype applied to a synthetic data set. The stacked graph-based visualization is depicted on the top left of the interface. From the stacked graph it is possible to notice that the amount of variation in the data increases initially, remains stable for a while, then decreases. The data does not present time variation in the middle of the time domain, but there are nodes with non-zero coefficients due to spatial variation. By changing the slider under the stacked graph we can control the minimum value of the coefficients to be considered during visualization, hiding noises or events of smaller magnitude.

Nodes with the same classification are grouped and depicted with a specific color that encodes their frequency class. The average pattern of coefficients in each class is shown in the bottom left area of the interface. From the stacked graph one can clearly see that the variation on the data is predominantly of low-frequency, happening in more than 50% of the nodes of the Cartesian product graph. High and mid-high frequencies take place in about 20% of the nodes. From the small amount of vertices classified as high-frequency, we can infer that high-frequency events happen along the time, but they are spatially small. In this specific data set, detailed in Section 6.4, the time-varying information was designed to alternate between zero and one in

Figure 28 – Screenshot our prototype interface which comprises a stacked graph visualization (top left), threshold slide bar for wavelets coefficients (middle left), average feature vector in each class (bottom left), graph map depicting node labels and function intensity in specific time slice (top right), and time series of selected node (bottom right). Source: (VALDIVIA *et al.*, 2015).

sepecific regions and time intervals of the graph, but the region where the change happens is large and the underlying function remains constant for enough time to be expressed using lower frequencies, making high frequency coefficients to show up only in a small number of nodes.

The stacked graph visualization is linked to a time slice visualization depicting function intensities and corresponding nodes classification, as shown on top right in Figure 28. A time slice is selected from the stacked graph by placing the mouse on a specific time. Each node in that time interval with coefficients higher than a threshold (set in the slide bar) is colored according to its classification. By clicking on a node one can access the corresponding feature vector and function value for that node. Moreover, the time series associated with the selected node is presented on the bottom right area of the interface, allowing the users to visualize how the data varies in that node over the whole period of time.

6.4 Visual analytics with graph Wavelets

In this section we present some basic visual analysis enabled by the proposed methodology. To properly illustrate particular aspects of our method, we have designed synthetic time-varying data containing specific properties we deem important to be presented and discussed, postponing the use of real data to Section 6.5. The time-varying function domain is a graph with 1200 nodes and 50 time slices, resulting in a Cartesian product graph with 60,000

nodes. The system prototype was implemented in JavaScript and the data has been pre-processed in a regular desktop computer, equipped with an i7 Intel processor and 8Gb of RAM, in order to compute wavelets coefficients in each node of the Cartesian product graph. The pre-processing step for this particular synthetic data set took approximately 20 seconds.

In our context, an event is any portion of the domain where one can notice a variation in the data under analysis. This definition is quite general and can be adapted to many different contexts. Our dataset contains six distinct types of events, which appear in different sizes and time intervals with distinct duration over time. In the following, we show the usefulness of the proposed combination of graph wavelets, pattern classification, and stacked graph when analyzing the given synthetic data.

Figure 29 shows two sets of time slices from our synthetic data set. More specifically, Figures 29a and 29b correspond to the classification of nodes in two adjacent time slices t_5 and t_6 , while Figures 29c and 29d show node classification in non-consecutive time slices t_i and t_j . Figures 29e, 29f, 29g, 29h depict the function in the same time slices as in Figures 29a, 29b, 29c, and 29d, respectively.

Figure 29a shows the nodes classified according to their wavelet coefficients in time slice t_5 . Notice that even though the function has no value in that time slice (all nodes are gray in Figure 29e) wavelets coefficients are non zero, indicating that some temporal function variation should take place from adjacent time slices. Moreover, nodes are mostly in orange class, meaning that a mid-high frequency variation is about to happen in that region. In summary, Figures 29a and 29e point out that the data should present an abrupt temporal change from time slice t_5 to the adjacent slices, what is attested when analyzing Figures 29b and 29f referring to time t_6 . Notice from Figure 29b that several nodes in the middle of the event become low-frequency (blue nodes), which are surrounded by yellow and orange nodes, indicating a spatial and temporal stability in the center of the event from time slice t_6 to the following slices. The nodes on the border of the event are mostly medium-high frequency, with some high frequency nodes. Those high, mid-high frequency nodes are mainly due to spatial variation of the data. Nodes classified as indeterminate indicate that both low (mid-low) as well as high (mid-high) frequencies are present in those nodes due to the simultaneous temporal and spatial variability.

It is also easy to see that a new region with light-blue nodes shows up in Figure 29b, indicating a variation from t_6 to the following time slices in that region. However, those nodes belong to the mid-low frequency class, meaning that either the function variation is not so “severe” in that region or the data variation will not take place exactly in t_7 but in the following time slices. By clicking in one of the nodes in the new blue region we can visualize its corresponding time series, as depicted in Figure 30. From the time series it is easy to see a function variation from time slice t_8 .

Figures 29c and 29g show another time slice of the synthetic data. Coefficients in the middle of the largest events are dominated by low frequency nodes, while nodes in the border

Figure 29 – Coefficients and raw data for four distinct time slices. Top row: Coefficients, bottom row: respective raw data. Source: (VALDIVIA *et al.*, 2015).

Figure 30 – Time series associated to a node in the bottom blue region in Figure 29b. Source: (VALDIVIA *et al.*, 2015).

Table 2 – Real data sets used in the Case Studies. Manhattan’s street graph G contains 4,694 nodes.

Data	Time slices in H	Nodes in $G \times H$	Pre-processing times (aprox.)
Bike - Oct 28	96	450,624	1h
Bike - Oct 25–31	168	788,592	2h
Taxi - Aug 11–18	336	1,577,184	10h

tend to be mid-high frequency nodes. Some nodes that are never part of an event are also classified as mid-low frequency, indicating their spatial proximity to other events. However, not all nearby vertices have coefficients above the chosen threshold, so they remain hidden. The time slice just after the end of the largest event is presented in Figures 29d and 29h. Similarly to what happened in Figure 29a, mid-high frequency nodes indicate a new temporal change related to the vanishment of event.

6.5 Case studies

In this section, we present two case studies, using real data, where our proposed methodology is used to explore urban data. Both studies consider downtown Manhattan - NY, which is expressed as a graph. The first data set contains information about bike movements, the second contains information about taxi pickups. Table 2 briefly presents some details of the data sets, along with the computational times as to pre-processing step to compute graph wavelets coefficients and node classification.

6.5.1 Citi Bike

Figure 31 – *Citi Bike NY* station map. Stations from downtown Manhattan (pink markers) are considered in this study.

This data set is composed of information about trips of bikes that can be rented at stations in New York City [CitiBikeNY \(2015\)](#). Only stations from downtown Manhattan - NY are considered in this study, depicted as pink markers in Figure 31. Each data instance contains

Trip Duration (seconds), Start Time and Date, Stop Time and Date, Start Station Name, End Station Name, Station ID, Station Lat/Long, Bike ID, User Type (24-hour, 7-day pass, or Annual Member), Gender (unknown, male, female), and Year of Birth.

Since the data provides the start and end geographic location for each bike, we approximate routes by computing the shortest path between start and end location using the Manhattan's street map as background graph. Assuming the bikes travel at constant speed, the trip duration can be used to estimate the position of each bike in its approximate shortest route. Given a node in the Manhattan's street graph and a time interval, we use bike positions to identify the bikes that are closest to that node in the given time interval. The number of closest bikes is aggregated to provide the density of bikes in each node. Top images in Figure 32 shows the density maps as to October 28th at 8:30am (left), and at 5:30pm (right), 2013. Bottom images in Figure 32 show how the density function varies in the corresponding regions of the map in the given time intervals. The day was discretized in intervals of fifteen minutes and downtown Manhattan's graph has 4,694 nodes, resulting in a Cartesian product graph with 450,624 nodes.

Figure 33a depicts four highlighted regions that correspond to high or mid-high frequency classes, that is, in those regions the density of bikes changes quite abruptly. Comparing Figure 33b with Figure 33a we notice that the high frequency regions labeled as 1, 2, and 3 correspond to areas with a high concentration of bike lanes, indicating that the density of bikes has more variation near bike lanes, in particular at the points where bike lanes intersect each other. Region 4 also presents a high level of data variation, however, it does not correspond to a bike lane region. In fact, it is next to the main bike lines in the east part of Manhattan. After careful analysis, we concluded that the discrepancy is due to our naive algorithm for approximating routes, which only computes the shortest path between start and stop stations without taking into account the bike lanes. In contrast to what happen in the west part of Manhattan, in the east area the bike stations are not concentrated along the bike lanes, thus making shortest path routes a bad approximation to the true routes.

Figure 34 shows the stacked graph of one week of bike activities from October 25th to 31st, 2013. In this case, each day was discretized hourly, leading to a Cartesian graph with 788,592 nodes. The stacked graph clearly reveals several interesting information. For instance, it is pretty clear that there is a different pattern between working days and weekend. In working days, there are two peaks of activities, one in the morning, between 8am and 9am, and another in late afternoon, between 5pm and 6pm. The peak in the morning is smaller than in the afternoon, showing that more activity happens in afternoons. The amount of high and low frequency variation is almost the same in the morning. However, regions of low frequency are predominant in the afternoon, meaning that the density of bikes varies more smoothly in this period. During the weekend, there is only one peak of bike activity, which is more concentrated in the afternoon.

It is also possible to see from the stacked graph in Figure 34 that something different happened on Thursday 31st, as morning and afternoon peaks are almost of the same size. October

Figure 32 – Time-slices of the density maps of bikes in downtown Manhattan in October 28, 2013, about 8:30am (left) and about 5:30pm (right). Wavelet coefficients classification are shown below their respective time intervals showing variation regions of the density function. Colors correspond to classes of low (dark blue), mid-low (light blue), mid-high (orange), high (red) and indetermined (yellow) frequency variation. Source: (VALDIVIA *et al.*, 2015).

Figure 33 – High frequency variations tend to occur on the bike lanes, particularly on their intersections.
Source: (VALDIVIA *et al.*, 2015).

Figure 34 – Stacked graph showing one week of bike active from October 15th to 31st, 2013. The activity patterns in working days and weekend are clearly different. Source: (VALDIVIA *et al.*, 2015).

31st is Halloween day, so the change in activity is possibly caused by the special date.

6.5.2 NY Taxi

This data set contains information regarding taxi trips in downtown Manhattan. The original data was provided by the NYC Taxi and Limousine Commission and contains information about the taxis themselves, the drivers, the amount of passengers, the pickup and drop-off times, the duration and distance of the trips, and geographical information for the pickups and drop-offs. While this data set also contains the geographical start and end of each trip, we did not estimate trajectories, but consider only the start location of each trip.

Figure 35 depicts the stacked graph of one week of taxi activities, from August 12th to 18th, 2013. Each day was discretized and the data aggregated in periods of thirty minutes,

Figure 35 – Stacked graph visualization of one week of taxi pickups from August 12th to 18th. A large number of taxi trips happen at night. Source: (VALDIVIA *et al.*, 2015).

leading to a Cartesian graph with 1,577,184 nodes. Similarly to the bike dataset, the pattern of activity of each day is clearly represented, but, in this case, weekdays and weekends have a more similar shape, with variable intensity. The most significant difference is between night and day periods, since during the night the number of taxi trips increases considerably.

Comparison between day and night patterns reveals a decrease between 4pm and 5pm, but the amount varies for each day of the week, from a slight reduction on Monday to a very accentuated valley on Friday. Taxi activity by the end of the night also varies, going from 11pm on Monday to 4:30am on Friday and, curiously, 4am on Saturdays. Although the night activity lasts longer on Friday, our linked view reveals that the amount of nodes with significant activity at the same period decreases, thus keeping the total of affected nodes roughly the same, as one can clearly see by comparing the shape of Friday against Thursday or Wednesday. Sunday has less trips, evenly distributed among the whole active period.

Considering the classification of nodes, we see a predominance of low frequency, even at 5pm, where we have a clear temporal change. Therefore, we can assume that this temporal variation occurs as a large, uniform event, that can be characterized as low frequency. In other words, while the number of taxi trips rapidly increases at 5pm, it does so in a large part of the Manhattan graph domain, without many peaks of activity. However, there are also nodes classified as high or mid-high frequency, pointing some abrupt variation in specific locations of the map. However, the number of such nodes is small compared to the number of low frequency nodes.

Figure 36 illustrates the classification of vertices and the density pickup function for four different time slices. The first column represents the time slice from 7:30am to 8am, August 12, 2013. There are taxi pickups in most of the considered region, but the method identified regions of particular changes around Penn Station, the Port Authority bus terminal on 42th street, and at the corner of Liberty street and South End Avenue, highlighted on bottom left of the map. While

Figure 36 – Four different time slices for the taxi activity, corresponding to August 12, 2013, 7:30am to 8am (left); August 13, 2013, 10:30pm to 11pm (mid-left); August 16, 2013, 7:30pm to 8pm (mid-right); and August 17, 2013, 1:30am to 2am (right). Source: (VALDIVIA *et al.*, 2015).

Figure 37 – Time series associated to the node representing the intersection of Liberty street and South End avenue, highlighted in the first column of Figure 36. Source: (VALDIVIA *et al.*, 2015).

the large amount of taxi pickups around two of the most busy public traffic hubs of New York City are expected, and clearly visible in the raw data, the third identified location is curious. The density value associated to that node is lower than in many nodes in the map, notably near Grand Central Station (top mid-east in the map). The time series associated with the highlighted node is illustrated in Figure 37. There are visible peaks of activity during the morning in weekdays. While the value of the function at such peaks is not very high, corresponding to less than a trip per minute, they represent a clear change in the local dynamics of this particular region.

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

The second column in Figure 36 illustrates the period from 10:30pm to 11pm, Tuesday, August 13, 2013. There is a clear increase of taxi activity around Times Square, reflecting the nightlife activity. The highlighted node corresponds to the intersection of 50th street and 8th avenue, near several theaters. Our visual analytics method also identified two locations of high variation, one in Tribeca and another near Nolita (orange nodes in the south). While there is a clear change in these locations at the considered time, we found no obvious explanation for those peaks, deserving a more careful analysis.

The third column in Figure 36 shows the period from 7:30pm to 8pm, Friday, August 16, 2013. While the activity around Penn Station and the Port Authority bus terminal remain significant, there is an increase in the activity in the Turtle Bay area, mostly in its northwest portion, an area with several high-end hotels. Therefore, we believe that such increase is related to the guests of such hotels leaving to enjoy the nightlife on Friday.

The fourth column in Figure 36 illustrates the period from 1:30am to 2am, Saturday, August 17, 2013. We can clearly identify regions with high concentration of bars and restaurants, such as South Chelsea and the Lower East Side. However, we see some high-frequency nodes highlighted in the middle of the map, located in the intersection of 21st street and 6th avenue. The time series corresponding to the highlighted node is shown in Figure 38, from which one can see an increase in taxi activity after midnight, on Fridays, Saturdays and Sundays, with almost two taxi trips per minute starting nearby. We also found no obvious explanation for such abrupt variation during the weekend.

The provided case studies showcase a particular feature of the proposed method, its ability to identify significant changes in the function under analysis, regardless of the magnitude of the associated data. Therefore, it can direct the attention of users to the important changes

Figure 38 – Time series associated to the vertex representing the node highlighted in the fourth column of Figure 36. Source: (VALDIVIA *et al.*, 2015).

without “contaminating” the visualization with higher magnitude data whose behavior is more uniform. In other words, the method is capable of separating interesting changes from the usual evolution of the data, which would otherwise require expert knowledge or a deeper analysis using behavioral models. This property renders our methodology a competing alternative for time-varying data analysis.

6.6 Discussion and limitations

The provided results and case studies clearly show that our approach is able to deal with large amounts of information, while allowing the visualization of spatiotemporal events, regardless of their size. Despite the naive method used to approximate routes in the Citi Bike analysis, the proposed visual analytic tool has turned out effective in revealing how the density of bikes varies over time, pointing out interesting events and their corresponding regions and time intervals. A more elaborated algorithm to approximate routes would certainly enable a more precise analysis of the data.

The proposed classification scheme, based on simulated training data, has turned out satisfactory, enabling informative visualization and data analysis. We have attested the correctness of the classification by manually inspecting the classification results against the corresponding feature vector. However, a more thorough verification method should be implemented, specially for applications demanding a degree of confidence beyond the proof of concept presented in this work.

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

While the stacked graph plot has turned out quite informative, it is not entirely appropriate when the classes are very unbalanced, that is, classes with few elements can be difficult to visualize. However, this minority information can indicate outliers, as pointed out in the case

study involving taxi data. Moreover, when a large temporal interval is considered, the plot can become quite dense, hindering the visualization of smaller variations in the classes.

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

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

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

6.7 Final considerations

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

CONCLUSIONS

In this dissertation, we discussed concepts on graph signal processing and graph wavelets with some didactic examples to help interpreting the Graph Fourier Transform and Graph Wavelets Transform coefficients. We presented how these tools can be used to perform effective visual data analysis. Specifically how graph signal filtering can assist high-dimensional data visualization and how GWT can enable visual analysis in urban data.

In Chapter 3 we showed how the proposed filtering methodology based on graph signal processing theory, improve the quality of visualizations. We provided qualitative and quantitative results showing the positive impact of our method to generate high quality visualizations.

This method is just a first step towards making filtering mechanisms a basic tool in the context of visualizations. There are, indeed, several possibilities that need to be explored. For instance, the construction of the graph plays an important role in our pipeline and we need to further assess the impact of the method used to construct the graph. Another important aspect that needs to be investigated is the design of the filters. We believe that the filter design depends on the application and on the visualization. Also, we are working in evaluating how the data is transformed, for that an statistical study on raw and filtered data needs to be done, and also visual hints could be provided to indicate the data transformation.

The results presented in Chapter 6 show that our approach facilitates the discovery of spatiotemporal events in large amounts of data. The usefulness of the proposed methodology was attested through a set of tests and case studies, rendering it an attractive methodology for many visualization applications. The provided results and case studies clearly show that our approach is able to deal with large amounts of information, while allowing the visualization of spatiotemporal events, regardless of their size.

Some limitations of our approach are considered avenues for future work. Specifically, how to discriminate spatial from temporal changes using the graph wavelet coefficients is challenging. Another relevant limitation to consider is how the topology of the graph affects the

resulting coefficients. This may be solved using the Spectral Adapted Graph Wavelets instead of the Spectral Graph Wavelets, but it wouldn't allow us to use the property of Cartesian product graph for computing the spectrum, making unfeasible to work with large datasets.

Finally, we consider that the visualization field can benefit from the tools derived from graph signal processing (GSP). The proposed applications of GSP to visually analyze data are a first step towards incorporating the use of this theory into information visualization methods. Many possibilities from GSP can be explored by improving the understanding of static and time-varying phenomena that are yet to be uncovered.

BIBLIOGRAPHY

AIGNER, W.; MIKSCH, S.; MÜLLER, W.; SCHUMANN, H.; TOMINSKI, C. Visualizing time-oriented data—a systematic view. **Computers & Graphics**, v. 31, n. 3, p. 401–409, 2007. Citation on page [53](#).

AIGNER, W.; MIKSCH, S.; SCHUMANN, H.; TOMINSKI, C. **Visualization of Time-Oriented Data**. [S.l.]: Springer London, 2011. (Human-Computer Interaction Series). Citation on page [53](#).

ANDRIENKO, G.; ANDRIENKO, N. Coordinated multiple views: a critical view. In: **Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization**. [S.l.]: IEEE, 2007. p. 72–74. Citation on page [57](#).

ANDRIENKO, G.; ANDRIENKO, N. Spatio-temporal aggregation for visual analysis of movements. In: **2008 IEEE Symposium on Visual Analytics Science and Technology (VAST)**. [S.l.]: IEEE, 2008. p. 51–58. Citations on pages [30](#) and [59](#).

ANDRIENKO, G.; ANDRIENKO, N.; BAK, P.; KEIM, D.; KISILEVICH, S.; WROBEL, S. A conceptual framework and taxonomy of techniques for analyzing movement. **Journal of Visual Languages and Computing**, v. 22, n. 3, p. 213–232, 2011. Citations on pages [54](#) and [55](#).

ANDRIENKO, G.; ANDRIENKO, N.; BREMM, S.; SCHRECK, T.; Von Landesberger, T.; BAK, P.; KEIM, D. Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns. **Computer Graphics Forum**, v. 29, n. 3, p. 913–922, 2010. Citation on page [58](#).

ANDRIENKO, N.; ANDRIENKO, G. Interactive visual tools to explore spatio-temporal variation. In: **Proceedings of the Working Conference on Advanced Visual Interfaces**. [S.l.]: ACM Press, 2004. p. 417. Citations on pages [54](#) and [55](#).

ANDRIENKO, N.; ANDRIENKO, G. Visual analytics of movement: An overview of methods, tools and procedures. **Information Visualization**, v. 12, n. 1, p. 3–24, 2012. Citation on page [53](#).

BACH, B.; DRAGICEVIC, P.; ARCHAMBAULT, D.; HURTER, C.; CARPENDALE, S. A review of temporal data visualizations based on space-time cube operations. In: **Eurographics Conference on Visualization**. [S.l.: s.n.], 2014. p. 23–41. Citation on page [53](#).

BAK, P.; MANSMANN, F.; JANETZKO, H.; KEIM, D. A. Spatiotemporal analysis of sensor logs using growth ring maps. **IEEE Transactions on Visualization and Computer Graphics**, v. 15, n. 6, p. 913–920, 2009. Citations on pages [54](#) and [55](#).

BEBIS, G.; BOYLE, R.; PARVIN, B.; KORACIN, D.; KUNO, Y.; WANG, J.; PAJAROLA, R.; LINDSTROM, P.; HINKENJANN, A.; ENCARNACÃO, L.; SILVA, C.; COMING, D. S. **Advances in Visual Computing**. [S.l.]: Springer Science & Business Media. Citations on pages [55](#) and [56](#).

BERGNER, S.; MOLLER, T.; WEISKOPF, D.; MURAKI, D. J. A spectral analysis of function composition and its implications for sampling in direct volume visualization. **IEEE Transactions on Visualization and Computer Graphics**, v. 12, n. 5, 2006. Citation on page 30.

BIYIKOGLU, T.; LEYDOLD, J.; STADLER, P. F. **Laplacian eigenvectors of graphs**. [S.l.]: Springer, 2007. (Lecture notes in mathematics). Citations on pages 22 and 62.

BUADES, A.; COLL, B.; MOREL, J.-M. A non-local algorithm for image denoising. In: **IEEE CVPR**. [S.l.]: IEEE, 2005. p. 60–65. Citation on page 17.

CHENG, T.; TANAKSARANOND, G.; BRUNSDON, C.; HAWORTH, J. Exploratory visualisation of congestion evolutions on urban transport networks. **Transportation Research Part C: Emerging Technologies**, v. 36, p. 296–306, 2013. Citation on page 56.

CHU, D.; SHEETS, D. A.; ZHAO, Y.; WU, Y.; YANG, J.; ZHENG, M.; CHEN, G. Visualizing hidden themes of taxi movement with semantic transformation. In: **2014 IEEE Pacific Visualization Symposium**. [S.l.]: IEEE, 2014. p. 137–144. Citation on page 59.

CHUNG, F. R. **Spectral graph theory**. [S.l.]: American Mathematical Soc., 1997. Citation on page 22.

CitiBikeNY. **Citi Bike trip data**. [S.l.], 2015. Available: <<https://www.citibikenyc.com/system-data>>. Citation on page 69.

COIFMAN, R. R.; MAGGIONI, M. Diffusion wavelets. **Applied and Computational Harmonic Analysis**, v. 21, n. 1, p. 53–94, 2006. Citation on page 48.

CROVELLA, M.; KOLACZYK, E. Graph wavelets for spatial traffic analysis. In: **IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies**. [S.l.]: IEEE, 2003. p. 1848–1857. Citation on page 47.

CVETKOVIĆ, D. M.; ROWLINSON, P.; SIMIC, S. **An Introduction to the Theory of Graph Spectra**. [S.l.]: Cambridge University Press, 2009. Citation on page 22.

Dal Col, A.; VALDIVIA, P.; PETRONETTO, F.; DIAS, F.; SILVA, C. T.; NONATO, L. G. Wavelet-based visual analysis for data exploration. **Computing in Science & Engineering**, v. 19, n. 5, p. 85–91, 2017. Citation on page 18.

Dal Col, A.; VALDIVIA, P.; PETRONETTO, F.; DIAS, F.; SILVA, C. T.; NONATO, L. G. Wavelet-based visual analysis of dynamic networks. **IEEE Transactions on Visualization and Computer Graphics**, 2017. Citations on pages 17, 19, and 31.

DASGUPTA, A.; KOSARA, R. Pargnostics: Screen-space metrics for parallel coordinates. **IEEE Transactions on Visualization and Computer Graphics**, v. 16, n. 6, p. 1017–1026, 2010. Citations on pages 40 and 41.

de Leeuw, W. C.; Van Wijk, J. J. Enhanced spot noise for vector field visualization. In: **IEEE Visualization**. [S.l.: s.n.], 1995. p. 233–240. Citation on page 30.

DEMSAR, U.; BUCHIN, K.; CAGNACCI, F.; SAFI, K.; SPECKMANN, B.; Van de Weghe, N.; WEISKOPF, D.; WEIBEL, R. Analysis and visualisation of movement: an interdisciplinary review. **Movement Ecology**, v. 3, n. 1, p. 19052, 2015. Citation on page 53.

DEMSAR, U.; FOTHERINGHAM, A. S.; CHARLTON, M. Exploring the spatio-temporal dynamics of geographical processes with geographically weighted regression and geovisual analytics. **Information Visualization**, v. 7, n. 3-4, p. 181–197, 2008. Citation on page 58.

DIAS, F.; MANSOUR, M. R.; VALDIVIA, P.; COUSTY, J.; NAJMAN, L. Watersheds on hypergraphs for data clustering. In: ANGULO, J.; VELASCO-FORERO, S.; MEYER, F. (Ed.). **Mathematical Morphology and Its Applications to Signal and Image Processing**. [S.l.]: Springer International Publishing, 2017. p. 211–221. ISBN 978-3-319-57240-6. Citation on page 19.

DONG, X.; THANOU, D.; FROSSARD, P.; VANDERGHEYNST, P. Learning laplacian matrix in smooth graph signal representations. **IEEE Transactions on Signal Processing**, v. 64, n. 23, p. 6160–6173, 2016. Citation on page 44.

DORAISWAMY, H.; FERREIRA, N.; DAMOULAS, T.; FREIRE, J.; SILVA, C. T. Using topological analysis to support event-guided exploration in urban data. **IEEE Transactions on Visualization and Computer Graphics**, v. 20, n. 12, p. 2634–2643, 2014. Citation on page 58.

DWYER, T.; GALLAGHER, D. R. Visualising changes in fund manager holdings in two and a half-dimensions. **Information Visualization**, v. 3, n. 4, p. 227–244, 2004. Citation on page 55.

EBLING, J.; SCHEUERMANN, G. Clifford fourier transform on vector fields. **IEEE Transactions on Visualization and Computer Graphics**, v. 11, n. 4, p. 469–479, 2005. Citation on page 30.

FERREIRA, N.; POCO, J.; VO, H. T.; FREIRE, J.; SILVA, C. T. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. **IEEE Transactions on Visualization and Computer Graphics**, v. 19, n. 12, p. 2149–2158, 2013. Citations on pages 57 and 58.

FREY, B. J.; DUECK, D. Clustering by passing messages between data points. **science**, v. 315, n. 5814, p. 972–976, 2007. Citation on page 38.

FUCHS, J.; FISCHER, F.; MANSMANN, F.; BERTINI, E.; ISENBERG, P. Evaluation of alternative glyph designs for time series data in a small multiple setting. In: **the SIGCHI Conference**. [S.l.]: ACM Press, 2013. p. 3237. Citation on page 53.

GASTAL, E. S. L.; OLIVEIRA, M. M. Adaptive manifolds for real-time high-dimensional filtering. **ACM Trans. Graph.**, 2012. Citation on page 17.

GATALSKY, P.; ANDRIENKO, N.; ANDRIENKO, G. Interactive analysis of event data using space-time cube. In: **2004 Proceedings Eighth International Conference on Information Visualisation**. [S.l.]: IEEE, 2004. p. 145–152. Citation on page 55.

GRANLUND, G. H.; KNUTSSON, H. **Signal processing for computer vision**. [S.l.]: Springer Science & Business Media, 2013. Citation on page 17.

GUO, H.; WANG, Z.; YU, B.; ZHAO, H.; YUAN, X. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In: **2011 IEEE Pacific Visualization Symposium**. [S.l.]: IEEE, 2011. p. 163–170. Citation on page 59.

HÄGERSTRAAND, T. What about people in regional science? **Papers in Regional Science**, v. 24, n. 1, p. 7–24, 2005. Citation on page 55.

HAMMOND, D. K.; VANDERGHEYNST, P.; GRIBONVAL, R. Wavelets on graphs via spectral graph theory. **Applied and Computational Harmonic Analysis**, v. 30, n. 2, p. 129–150, 2011. Citations on pages [17](#), [25](#), [48](#), and [49](#).

HAVRE, S.; HETZLER, B.; NOWELL, L. Themeriver: visualizing theme changes over time. In: **IEEE Visualization 2000**. [S.l.]: IEEE, 2000. p. 115–123. Citation on page [59](#).

HOTELLING, H. Analysis of a complex of statistical variables into principal components. **Journal of Educational Psychology**, v. 24, n. 6, p. 417–441, 1933. Citation on page [42](#).

HUANG, W.; GOLDSBERRY, L.; WYMBS, N. F.; GRAFTON, S. T.; BASSETT, D. S.; RIBEIRO, A. Graph frequency analysis of brain signals. **CoRR**, 2015. Citation on page [31](#).

HUBERT, L.; ARABIE, P. Comparing partitions. **Journal of classification**, v. 2, n. 1, p. 193–218, 1985. Citation on page [38](#).

JANSEN, M.; NASON, G. P.; SILVERMAN, B. W. Multiscale methods for data on graphs and irregular multidimensional situations. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, v. 71, n. 1, p. 97–125, 2009. Citation on page [48](#).

JERN, M.; FRANZEN, J. "geoanalytics" - exploring spatio-temporal and multivariate data. In: **Tenth International Conference on Information Visualisation**. [S.l.]: IEEE, 2006. p. 25–31. Citation on page [57](#).

JOIA, P.; PAULOVICH, F.; COIMBRA, D.; CUMINATO, J.; NONATO, L. G. Local affine multidimensional projection. **IEEE Transactions on Visualization and Computer Graphics**, v. 17, n. 12, p. 2563–2571, 2011. Citation on page [42](#).

KANG, X.; ZHANG, X.; LI, S.; LI, K.; LI, J.; BENEDIKTSSON, J. A. Hyperspectral anomaly detection with attribute and edge-preserving filters. **IEEE Transactions on Geoscience and Remote Sensing Letters**, v. 55, n. 10, p. 5600–5611, 2017. Citation on page [17](#).

KEIM, D.; ANDRIENKO, G.; FEKETE, J. D.; GÖRG, C.; KOHLHAMMER, J.; MELANCON, G. Visual analytics: Definition, process, and challenges. In: **Information Visualization**. [S.l.]: Springer Berlin Heidelberg, 2008. p. 154–175. Citation on page [53](#).

KEIM, D. A. Designing pixel-oriented visualization techniques: theory and applications. **IEEE Transactions on Visualization and Computer Graphics**, v. 6, n. 1, p. 59–78, 2000. Citation on page [55](#).

KHLEBNIKOV, R.; KAINZ, B.; STEINBERGER, M.; SCHMALSTIEG, D. Noise-based volume rendering for the visualization of multivariate volumetric data. **IEEE Transactions on Visualization and Computer Graphics**, v. 19, n. 12, p. 2926–2935, 2013. Citation on page [30](#).

KOHONEN, T. **Self-Organizing Maps**. [S.l.]: Springer Science & Business Media, 1995. (Springer Series in Information Sciences, v. 30). Citation on page [58](#).

KÖTHUR, P.; SIPS, M.; UNGER, A.; KUHLMANN, J.; DRANSCH, D. Interactive visual summaries for detection and assessment of spatiotemporal patterns in geospatial time series. **Information Visualization**, v. 13, n. 3, p. 283–298, 2014. Citation on page [58](#).

KRAAK, M.-J. The space-time cube revisited from a geovisualization perspective. In: **Proceedings 21st International Cartographic Conference**. [S.l.: s.n.], 2003. p. 1988–1996. Citation on page 55.

KYLE, K.; LAUDER, T. S. **California drought maps**. [S.l.], 2015. Citation on page 57.

LENNTORP, B. Paths in space-time environments: A time-geographic study of movement possibilities of individuals. **Environment and Planning A**, v. 9, p. 961–972, 1976. Citation on page 55.

LEONARDI, N.; Van De Ville, D. Wavelet frames on graphs defined by fmri functional connectivity. In: **2011 8th IEEE International Symposium on Biomedical Imaging (ISBI 2011)**. [S.l.]: IEEE, 2011. p. 2136–2139. Citations on pages 48 and 49.

LEONARDI, N.; Van De Ville, D. Tight wavelet frames on multislice graphs. **IEEE Transactions on Signal Processing**, v. 61, n. 13, p. 3357–3367, 2013. Citations on pages 48 and 49.

LHUILIER, A.; HURTER, C.; TELEA, A. Ffteb: Edge bundling of huge graphs by the fast fourier transform. In: **2017 IEEE Pacific Visualization Symposium**. [S.l.: s.n.], 2017. Citation on page 30.

LI, X.; GOEBEL, R.; SJOBERGH, J. Towards the identification of consumer trajectories in geo-located search data. In: **2014 18th International Conference on Information Visualisation (IV)**. [S.l.]: IEEE, 2014. p. 202–210. Citation on page 55.

LICHMAN, M. **UCI Machine Learning Repository**. [S.l.], 2013. Citations on pages 34 and 35.

LINS, L.; KLOSOWSKI, J. T.; SCHEIDEGGER, C. Nanocubes for real-time exploration of spatiotemporal datasets. **IEEE Transactions on Visualization and Computer Graphics**, v. 19, n. 12, p. 2456–2465, 2013. Citation on page 30.

LITANY, O.; RODOLÀ, E.; BRONSTEIN, A. M.; BRONSTEIN, M. M. Fully spectral partial shape matching. **Computer Graphics Forum**, v. 36, n. 2, p. 247–258, 2017. Citation on page 17.

LU, X.; DENG, Z.; CHEN, W. A robust scheme for feature-preserving mesh denoising. **IEEE Transactions on Visualization and Computer Graphics**, v. 22, n. 3, p. 1181–1194, 2016. Citation on page 17.

MACIEJEWSKI, R.; RUDOLPH, S.; HAFEN, R.; ABUSALAH, A.; YAKOUT, M.; OUZZANI, M.; CLEVELAND, W. S.; GRANNIS, S. J.; EBERT, D. S. A visual analytics approach to understanding spatiotemporal hotspots. **IEEE Transactions on Visualization and Computer Graphics**, v. 16, n. 2, p. 205–220, 2010. Citation on page 57.

MCDONNELL, K. T.; MUELLER, K. Illustrative parallel coordinates. **Computer Graphics Forum**, v. 27, n. 3, p. 1031–1038, 2008. Citation on page 30.

MOHAN, D. M.; ASIF, M. T.; MITROVIC, N.; DAUWELS, J.; JAILLET, P. Wavelets on graphs with application to transportation networks. In: **IEEE International Conference on Intelligent Transportation Systems**. [S.l.]: IEEE, 2014. p. 1707–1712. Citation on page 17.

MØLLER, M. F. A scaled conjugate gradient algorithm for fast supervised learning. **Neural Networks**, v. 6, n. 4, p. 525–533, 1993. Citation on page 63.

NARANG, S. K.; ORTEGA, A. Lifting based wavelet transforms on graphs. In: **APSIPA ASC 2009 - Asia-Pacific Signal and Information Processing Association 2009 Annual Summit and Conference**. [S.l.]: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, 2009. p. 441–444. Citation on page 47.

NARANG, S. K.; ORTEGA, A. Perfect reconstruction two-channel wavelet filter banks for graph structured data. **IEEE Transactions on Signal Processing**, v. 60, n. 6, p. 2786–2799, 2012. Citation on page 48.

NOVOTNY, M.; HAUSER, H. Outlier-preserving focus+ context visualization in parallel coordinates. **IEEE Transactions on Visualization and Computer Graphics**, v. 12, n. 5, p. 893–900, 2006. Citation on page 30.

NYCBikeMaps. **Manhattan Bike Paths, Bike Lanes Greenways**. [S.l.], 2015. Available: <<http://www.nycbikemaps.com/maps/manhattan-bike-map/>>. Citation on page 72.

OPPENHEIM, A. V. **Discrete-time signal processing**. [S.l.]: Pearson Education India, 1999. Citation on page 17.

PARK, H.; QUEALY, K. **Drought's Footprint**. [S.l.], 2012. Citation on page 57.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citation on page 34.

PETERSON, G. **Approved harvest applications by year**. [S.l.], 2011. Citation on page 57.

PU, J.; LIU, S.; XU, P.; QU, H.; NI, L. M. Mviewer: mobile phone spatiotemporal data viewer. **Frontiers of Computer Science**, v. 8, n. 2, p. 298–315, 2014. Citation on page 59.

REDDY, B. S.; CHATTERJI, B. N. An fft-based technique for translation, rotation, and scale-invariant image registration. **IEEE Transactions on Image Processing**, v. 5, n. 8, p. 1266–1271, 1996. Citation on page 17.

REICH, W.; HLAWITSCHKA, M.; SCHEUERMANN, G. Decomposition of vector fields beyond problems of first order and their applications. In: **Topological Methods in Data Analysis and Visualization**. [S.l.: s.n.], 2015. p. 205–219. Citation on page 30.

ROBERTS, J. C. State of the art: Coordinated & multiple views in exploratory visualization. In: **Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)**. [S.l.]: IEEE, 2007. p. 61–71. Citation on page 57.

ROSENBERG, A.; HIRSCHBERG, J. V-measure: A conditional entropy-based external cluster evaluation measure. In: **EMNLP-CoNLL**. [S.l.: s.n.], 2007. p. 410–420. Citation on page 38.

ROUSSEEUW, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. **Journal of computational and applied mathematics**, v. 20, p. 53–65, 1987. Citation on page 38.

SAITO, T.; MIYAMURA, H. N.; YAMAMOTO, M.; SAITO, H.; HOSHIYA, Y.; KASEDA, T. Two-tone pseudo coloring: compact visualization for one-dimensional data. In: **2005 IEEE Symposium on Information Visualization**. [S.l.]: IEEE, 2005. p. 173–180. Citations on pages [54](#) and [55](#).

SAKAS, G.; SCHREYER, L.-A.; GRIMM, M. Preprocessing and volume rendering of 3d ultrasonic data. **IEEE Computer Graphics and Applications**, 1995. Citation on page [30](#).

SANDRYHAILA, A.; MOURA, J. M. F. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. **IEEE Signal Processing Magazine**, v. 31, n. 5, p. 80–90, 2014. Citation on page [24](#).

SCHULZE, J. P.; KRAUS, M.; LANG, U.; ERTL, T. Integrating pre-integration into the shear-warp algorithm. In: **Eurographics/IEEE TVCG Workshop on Volume graphics**. [S.l.: s.n.], 2003. p. 109–118. Citation on page [30](#).

SHANBHAG, P.; RHEINGANS, P.; DESJARDINS, M. Temporal visualization of planning polygons for efficient partitioning of geo-spatial data. In: **2005 IEEE Symposium on Information Visualization**. [S.l.]: IEEE, 2005. p. 211–218. Citations on pages [54](#) and [55](#).

SHEN, G.; ORTEGA, A. Transform-based distributed data gathering. **IEEE Transactions on Signal Processing**, v. 58, n. 7, p. 3802–3815, 2010. Citation on page [47](#).

SHRESTHA, A.; MILLER, B.; ZHU, Y.; ZHAO, Y. Storygraph. In: **the ACM SIGKDD Workshop**. [S.l.]: ACM Press, 2013. p. 95–103. Citations on pages [54](#) and [55](#).

SHUMAN, D. I.; NARANG, S. K.; FROSSARD, P.; ORTEGA, A.; VANDERGHEYNST, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. **IEEE Signal Processing Magazine**, v. 30, n. 3, p. 83–98, 2013. Citations on pages [17](#), [27](#), and [47](#).

SHUMAN, D. I.; RICAUD, B.; VANDERGHEYNST, P. Vertex-frequency analysis on graphs. **Applied and Computational Harmonic Analysis**, v. 40, n. 2, p. 260–291, 2016. Citation on page [24](#).

SHUMAN, D. I.; WIESMEYR, C.; HOLIGHAUS, N.; VANDERGHEYNST, P. Spectrum-adapted tight graph wavelet and vertex-frequency frames. **IEEE Transactions on Signal Processing**, v. 63, n. 16, p. 4223–4235, 2015. Citations on pages [17](#), [48](#), [49](#), [50](#), and [51](#).

SOLTESZOVA, V.; BIRKELAND, Å.; STOPPEL, S.; VIOLA, I.; BRUCKNER, S. Output-sensitive filtering of streaming volume data. **Computer Graphics Forum**, v. 36, n. 1, p. 249–262, 2017. Citation on page [30](#).

SUN, G.; LIU, Y.; WU, W.; LIANG, R.; QU, H. Embedding temporal display into maps for occlusion-free visualization of spatio-temporal data. In: **2014 IEEE Pacific Visualization Symposium**. [S.l.]: IEEE, 2014. p. 185–192. Citation on page [59](#).

THAKUR, S.; HANSON, A. J. A 3d visualization of multiple time series on maps. In: **2010 14th International Conference Information Visualisation (IV)**. [S.l.]: IEEE, 2010. p. 336–343. Citation on page [55](#).

TOMINSKI, C.; SCHULZE-WOLLGAST, P.; SCHUMANN, H. 3d information visualization for time dependent data on maps. In: **Ninth International Conference on Information Visualisation (IV'05)**. [S.l.]: IEEE, 2005. p. 175–181. Citations on pages [55](#) and [56](#).

TREMBLAY, N.; BORGNAT, P. Graph wavelets for multiscale community mining. **IEEE Transactions on Signal Processing**, v. 62, n. 20, p. 5227–5239, 2014. Citation on page 17.

TUFTE, E. R. **Envisioning information**. [S.l.]: Graphics Press, 1990. (Envisioning Information). Citation on page 56.

TURDUKULOV, U. D.; KRAAK, M. J.; BLOK, C. A. Designing a visual environment for exploration of time series of remote sensing data: In search for convective clouds. **Computers & Graphics**, v. 31, n. 3, p. 370–379, 2007. Citation on page 55.

VALDIVIA, P.; DIAS, F.; PETRONETTO, F.; SILVA, C. T.; NONATO, L. G. Wavelet-based visualization of time-varying data on graphs. In: **2015 IEEE Conference on Visual Analytics Science and Technology (VAST)**. [S.l.]: IEEE, 2015. Citations on pages 14, 17, 18, 30, 62, 63, 64, 65, 66, 68, 71, 72, 73, 74, and 76.

VALLET, B.; LÉVY, B. Spectral geometry processing with manifold harmonics. **Computer Graphics Forum**, v. 27, n. 2, p. 251–260, 2008. Citation on page 17.

VINCENT, P.; LAROCHELLE, H.; LAJOIE, I.; BENGIO, Y.; MANZAGOL, P. A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. **Journal of Machine Learning Research**, v. 11, p. 3371–3408, 2010. Citation on page 34.

VIOLA, I.; KANITSAR, A.; GROLLER, M. E. Hardware-based nonlinear filtering and segmentation using high-level shading languages. In: **IEEE Visualization**. [S.l.]: IEEE, 2003. p. 309–316. Citation on page 30.

WAGNER, R.; DELOUILLE, V.; BARANIUK, R. Distributed wavelet de-noising for sensor networks. In: **Proceedings of the 45th IEEE Conference on Decision and Control**. [S.l.]: IEEE, 2006. p. 373–379. Citation on page 47.

WANG, Z.; LU, M.; YUAN, X.; ZHANG, J.; Van de Wetering, H. Visual traffic jam analysis based on trajectory data. **IEEE Transactions on Visualization and Computer Graphics**, v. 19, n. 12, p. 2159–2168, 2013. Citation on page 59.

WEGMAN, E. J. Visual data mining. **Statistics in medicine**, v. 22, n. 9, p. 1383–1397, 2003. Citation on page 30.

WICKHAM, H.; HOFMANN, H.; WICKHAM, C.; COOK, D. Glyph-maps for visually exploring temporal patterns in climate data and models. **Environmetrics**, v. 23, n. 5, p. 382–393, 2012. Citation on page 55.

ZENG, W.; FU, C.-W.; ARISONA, S. M.; QU, H. Visualizing interchange patterns in massive movement data. **Computer Graphics Forum**, v. 32, n. 3pt3, p. 271–280, 2013. Citation on page 59.

ZHANG, H.; Van Kaick, O.; DYER, R. Spectral mesh processing. v. 29, n. 6, p. 1865–1894, 2010. Citation on page 17.