Massaging Visualizations with Graph Spectral Filtering

P. Valdivia and L.G. Nonato

In this paper we propose a methodology to adapt the theory of Graph Signal Processing 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. 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.

1 Introduction

Signal processing has long been a fundamental tool in fields such as image processing [Opp99], computer vision [GK13], and computer graphics [ZVKD10], leveraging the development of filtering mechanisms designed to tackle problems such as denoising [BCM05, LDC16], anomaly detection [KZL*17], detail enhancement [GO12, VL08], object registration [RC96, LRBB17], among others. In visualization, signal processing techniques have also been playing its role, mainly to support scientific visualization applications [BMWM06, ES05]. However, the use of signal processing and filtering techniques are rarely used in the area of information visualization, in particular for unstructured data, which we take to be high-dimensional points without intrinsic connectivity.

In fact, the intrinsic unstructured nature of data involved in many information visualization applications hampers the direct use of standard methodologies such as spectral and multi-scale filters [Goo16] to reduce noise and emphasize structures hidden in the data. Such difficulty has also been faced in other areas, such as graphics and computer vision, where it fostered the development of new theoretical and computational methodologies [KZL*17,GO12,LRBB17]. However, the same has not been true for information visualization, where not even well known filtering schemes have been properly explored.

The present paper is a first step towards filling the gap mentioned above, proposing a novel methodology to enable the use of signal processing tools to assist high-dimensional data visualization. The proposed methodology builds upon spectral filtering mechanisms derived from the *Graph Signal Processing* (GSP) theory as a resource to reduce noise and highlight important structures present in

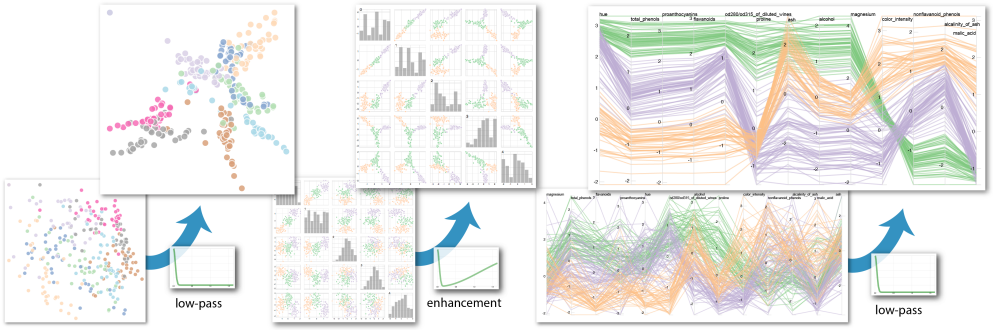


Figure 1: Graph spectral filtering can reduce noise and enphasize structures present in the data. From left to right, multidimensional projection, scatter plot matrix, and parallel coordinates visualizations before and after data filtering.

the data, supporting the generation of cleaner and more informative visualizations. As illustrated in Figure (**??**), we show that combining GPS-based spectral filtering with high-dimensional data visualization tools such as parallel coordinates, scatter plots, and multidimensional projection renders the visualizations more clean and informative. Moreover, we employ well established metrics to quantify the impact of the proposed methodology in the quality of visualizations. The provided experiments reveal that even simple filtering schemes can result in remarkably improved visual results. The effectiveness of the proposed methodology is shown through a set of experiments involving datasets with varying degrees of complexity and different dimensions.

In summary, the main contributions of this work are:

- the first study of using filtering techniques as an integral part of high-dimensional data visualization;
- a novel methodology to filter high-dimensional data aimed at improving visualizations;
- a quantitative study measuring the benefits of the proposed methodology for visualization purposes.

2 Related Work

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 [AA08], summarization [LKS13], and data pruning/selection [Weg03] 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 [BMWM06, SKLE03] and to filter aliasing effects introduced during the pertur-

bation of opacity maps [KKSS13]. In fact, the need for data filtering resource has long been acknowledged in context of volume rendering [SSG95, VKG03], being investigated until the present time [SBS*17]. Fourier transform is also the basis of methods designed to identify and visualize patterns in vector fields [ES05, RHS15]. Filters designed to operate in spatial domains has also been employed to remove low-frequency components from spot-noise based vector field visualizations [dLVW95].

Classical signal processing also plays a role in information visualization. A good example is the edge bundling technique proposed by Lhuillier et al. [LHT17], 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 [NH06]. Low-pass filtering has bee applied to smooth out density maps in illustrative parallel coordinates [MM08].

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 at al. [VDP*15] 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. [DCVP*17b] 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. [HGW*16], 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 paper comes to shed light on this issue, proposing a methodology that enables the use of graph filtering mechanisms to leverage more informative visualizations.

3 Graph Fourier Transform

Before presenting the proposed high-dimensional data filtering methodology we describe the mathematical foundations of the so-called Graph Fourier Transform (GFT) and Graph Filtering, which are the basis of our approach.

3.1 Graph Fourier Transform

We denote by $G = (V, E, w)$ a *graph* made up of a node set $V = \{\tau_1, \tau_2, \ldots, \tau_n\}$, an edge set $E = \{(\tau_i, \tau_j), \tau_i, \tau_j \in V, i \neq j\}$, and a weight function $w : E \to \mathbb{R}$ that associates a non-negative scalar to each edge of G. In our context, G is assumed to be *connected*, that is, for every pair of nodes there is a sequence of adjacent edges connecting the nodes.

The *weighted adjacency matrix* of G, denoted as $A = (a_{ij})$, is the matrix satisfying $a_{ij} = w(\tau_i, \tau_j)$ if $(\tau_i, \tau_j) \in E$ and $a_{ij} = 0$ otherwise. This matrix is used to define the (non-normalized) *graph Laplacian*, which is given by $L = D - A$, where $D = \text{diag}(d_1, d_2, \ldots, d_n)$ is a diagonal matrix with entries $d_i = \sum_j a_{ij}$ and n is the number of nodes in V. The graph Laplacian is a real, symmetric, and semi-positive definite matrix, what ensures a complete set of orthonormal real eigenvectors u_ℓ, with corresponding non-negative real eigenvalues λ_ℓ, $\ell = 1, 2, \ldots, n$. Moreover, zero is

always an eigenvalue of L whose corresponding eigenvector is a constant vector.

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. Moreover, eigenvectors associated to small eigenvalues tend to have a less oscillatory behavior than eigenvectors associated to large eigenvalues. A more detailed discussion about the relation between the spectrum of Laplacian matrices and Fourier theory can be found in the work by Shuman et al. [SRV16] and Dal Col et al. [DCVP*17a].

A signal defined on the nodes of G is a function $f : V \to \mathbb{R}$ that associates a scalar $f(\tau_i)$ to each node $\tau_i \in V$. Denoting the eigenvalues and corresponding eigenvectors of the Laplacian matrix of G by λ_ℓ and u_ℓ, $\ell = 1,\ldots,n$ respectively, and assuming the eigenvalues are sorted in non-decreasing order, $\lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$ (the first strict inequality is due to the assumption that G is connected), the *Graph Fourier Transform* (GFT) of a signal f, denoted $\hat{f} : \Lambda \to \mathbb{R}$, where Λ is the spectral domain (set of eigenvalues), is defined as:

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

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 $(\lambda_\ell, \hat{f}(\lambda_\ell))$, $i = 1,2,\ldots,n$, using a bar-like plot as illustrated in Figure (2).

Given the GFT $\hat{f}$, the original signal f can be recovered via the *inverse Graph Fourier Transform* (iGFT), which is defined as:

$$f = \sum_{\ell=1}^{n} \hat{f}(\lambda_\ell) u_\ell \qquad (2)$$

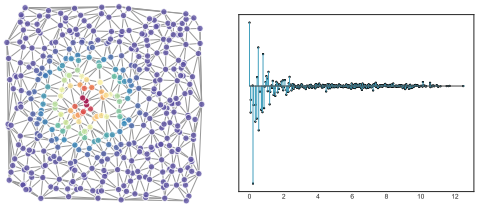


Figure 2: Left: a Gaussian function centered in a given node. Right: GFT of the Gaussian function. Since the Gaussian is a smooth function, the largest coefficients of the GFT are concentrated in the low frequency region of the spectrum (smaller eigenvalues).

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

$$\boxed{\text{GFT}} \quad \hat{f} = U^\top f \qquad\qquad \boxed{\text{iGFT}} \quad f = U\hat{f} \qquad (3)$$

3.2 Spectral Filtering

A *graph spectral filter* $\hat{h} : \Lambda \to \mathbb{R}$ 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 $\hat{f} : \Lambda \to \mathbb{R}$ can be seen as a particular instance of a graph spectral filter.

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

$$\hat{\tilde{f}} = \hat{f}\hat{h} \qquad (4)$$

where $\hat{f}$ is the GFT of f and $\hat{h}$ a graph spectral filter. Using the matrix notation defined in Eq. (3) and some algebraic manipulation one can obtain the filtered version $\tilde{f}$ of f in the graph domain as:

$$\tilde{f} = UHU^\top f \qquad (5)$$

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

The design a proper filter $\hat{h}$ is application dependent. In this work we will play with two

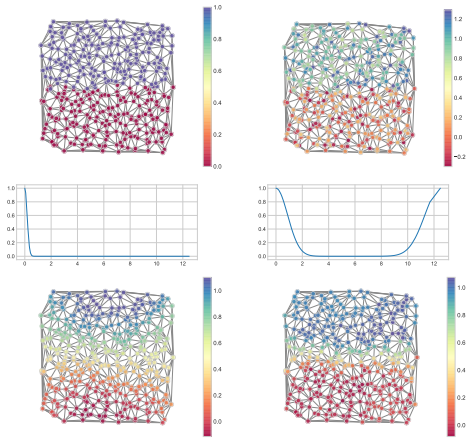


Figure 3: Top: Step function (left) and its noisy version (right). Middle: Low (left) and Enhancement (right) filters. Bottom: Noisy step function filtered with the low (left) and enhancement (right) filters.

different filters, low-pass $\hat{h}_l$, high-pass $\hat{h}_h$, and enhancement $\hat{h}_e$, which are define as:

$$\hat{h}_l(\lambda) = \exp\left(\frac{(-\alpha\lambda)^2}{(2\lambda_{\max})^2}\right), \quad \alpha \in [0,\infty) \tag{6}$$

$$\hat{h}_h(\lambda) = 1 - \hat{h}_l(\lambda) \tag{7}$$

$$\hat{h}_e(\lambda) = \beta f_l(x) + (1-\beta) f_h(x), \quad \beta \in [0,\infty) \tag{8}$$

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

Figure (3) illustrate the effect of a low-pass filter and an enhancement when applied in a noisy step function.

4 Spectral Filtering in Multidimensional Data

The GSP theory described in Section (3) 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 empha-

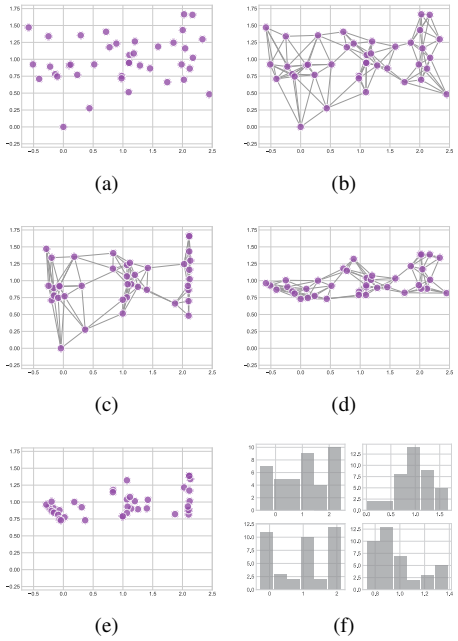


Figure 4: a): points drawn from three bivariate Gaussians centered at $(0,1)$, $(1,1)$, $(2,1)$ and constant diagonal covariance matrix; b): k-nearest neighbor graph with $k = 5$; c) and d): smoothing out the x and y coordinates respectively by a graph spectral low-pass filtering; e): configuration of 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).

size structures towards producing better visualizations.

The definition of a graph structure is the first step to adapt GSP to operate in high-dimensional data. In this work we adopt the simple approach of defining the graph G as the k-nearest graph (KNN-graph) of the data. As discussed in Section (6), 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.

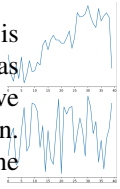
Let $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$ be a set of points in an m-dimensional space, $\tau_i \in \mathbb{R}^m$, $i = 1, \ldots, n$ and G be the KNN-graph derived from $\mathcal{T}$. From now on we will not make any distinction between a data instance τ_i and its corresponding node in 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) $\tau^j : V \to \mathbb{R}$ 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 Eq. (5). 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 (4) 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 (4(a)) 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 (4(b)) shows the k-nearest neighbor graph of the points with $k = 5$. Figures (4(c)) and (4(d)) 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 (4(c)), 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 (4(d)), where the y coordinate is smoothed out. Figure (4(e)) 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 (4(e)), which is not the case in original layout (Figure (4(a))).

Another interesting outcome of the filtering scheme is the better understanding of how each attribute is distributed. Figure (4(f)) 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 $\{x_1, \ldots, x_n\}$ means to impose an order to τ_i^j that in fact doesn't exist. The top image in the inset on the right depicts the x coordinates of the points in Figure (4(a)) as an one-dimensional signal were the domain is given by the index of the corresponding points. If we chose another other for the nodes, the shape of the signal changes completly, as illustrated at the bottom image in the inset. 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.

5 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 of quality of scatter plot, parallel coordinates and multidimensional projection layouts. Before describing and discussing the results, we present the datasets used in our experiments.

Table 1: Data sets used in the experiments.

Name	Size	Dim	# of classes	Source
Blobs	200	5	3	synthetic
Ecoli	336	5	8	[Lic13]
Wine	178	13	3	[Lic13]
Eggs	320	5000	8	proprietary

5.1 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 [Lic13], and a proprietary data set (Eggs) containing instances with 5000 attributes which correspond to the output layer of an Encoder-Decoder Neural Network [VLL*10] 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 [PVG*11] method `make_blobs` with three classes and standard deviation equal to 2.3. Only scalar attributes from the dataset Ecoli are used (categoriacal attributes are desregarded).

5.2 Scatter Plot Matrix

The top image in Figure 5 shows the scatter plot matrix generated from the synthetic dataset. Figures 5 middle and bottom are the scatter plot matrices of filtered versions of the synthetic dataset with a low-pass (middle) and an enhancement (bottom) graph filter 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-

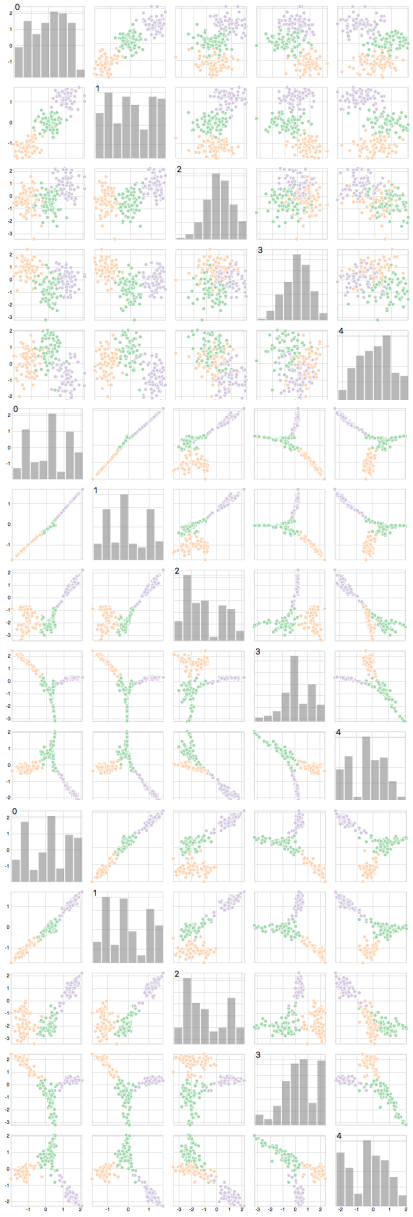
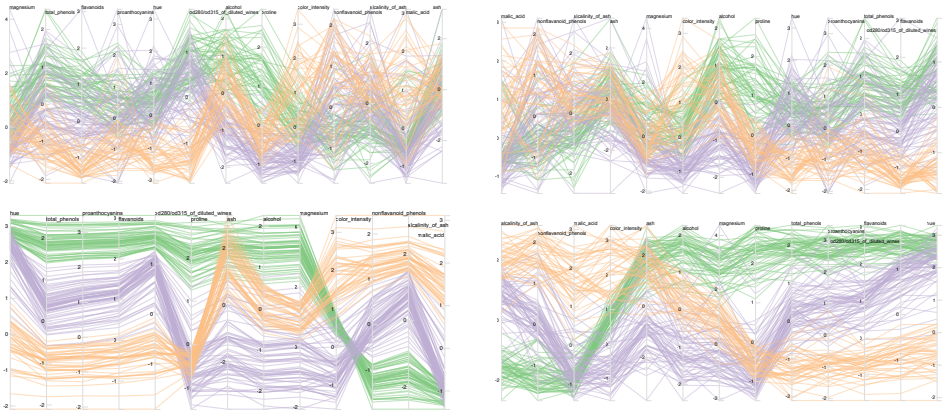


Figure 5: Scatter Plot Matrix of the synthetic dataset. Top: original data; Middle: low-pass filtered data; Bottom: enhancement filtered data.

matrices (middle and bottom) 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 are 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, histograms does not indicate those attributes are segregating the data into groups. However, the three groups are well separated when filtering 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 [Rou87] accounts for both the cohesion and separation of grouped instances. Homogeneity, completeness [RH07] and adjusted rand index [HA85] 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. Homegeneity 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 propaga-



(a) Overploting-based axes ordering

(b) Parallelism-based axes ordering

Figure 6: Parallel Coordinates view of the wine dataset with a) Over-plotting optimal axes arrangement and b) Parallelism optimal axes arrangement; a) and b) top: Original dataset; a) and b) bottom: low-pass and enhancement filtered data, respectively.

tion method [FD07] was used as clustering method as it does not require the number of clusters as input parameter.

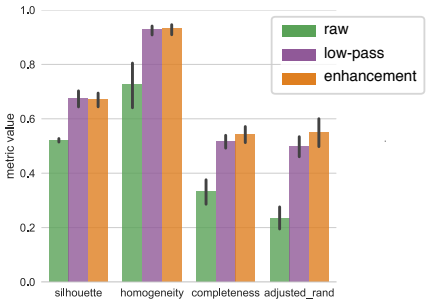


Figure 7: Average quality measure of the synthetic dataset scatter plots depicted in Figure (5). The larger the averages the better.

The result of applying the metrics described above to the scatter plots depicted in Figures 5 is presented in Figure 7. 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.

5.3 Parallel Coordinates

The second set experiments analyze the impact of graph filtering in parallel coordinates visualization. The top images in Figures (6(a)) and (6(b)) show parallel coordinates visualizations of the wine dataset using *over-ploting* and *parallelism* metrics (see details below) to find the optimal order of the axes. The bottom images in Figures (6(a)) and (6(b)) are parallel coordinates visualizations of low-pass and enhancement filtered versions of the data also using over-ploting 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.

Quantitative evaluation We rely on the methodology proposed by Dasgupta and Kosara [DK10] to measure the quality of parallel coordinates layouts. In short words, the

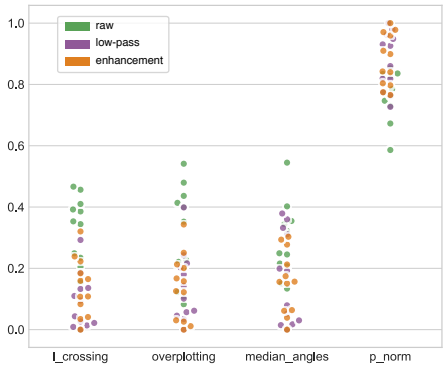


Figure 8: Parallel coordinates quality measures for the original (green), low-pass filtered (purple), and enhancement filtered (orange) data. The smaller the values of l_crossing, median angle, and over-plotting the better. The larger the p_norm the better. Each dot corresponds to the quality between a pair of axes in the layout.

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 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 [DK10].

Figure 8 depicts the result of applying the metrics to the "optimum" parallel coordinates layouts between each pair of axes. Notice that for l_crossing, over-ploting, 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.

5.4 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 [JPC*11] and PCA [Hot33] 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 (9) shows the result of projecting the original Ecoli data (top) and its enhancement filtered version (bottom) 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 (10) 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 informa-

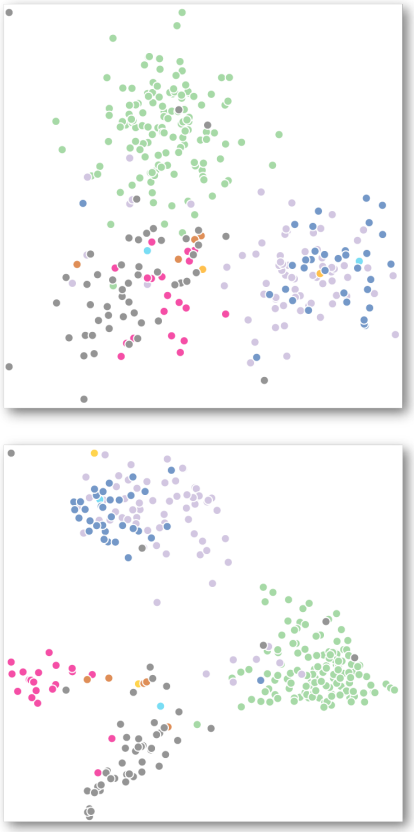


Figure 9: Lamp projection of the Ecoli dataset. Top: original data. Bottom: enhanced filtered data.

tion, 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 (10(a)) 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 (Section 5.2). The results are summarized in Figure 11. Once again, layouts produced from filtered data present better quality for all the metrics.

6 Discussion and Limitations

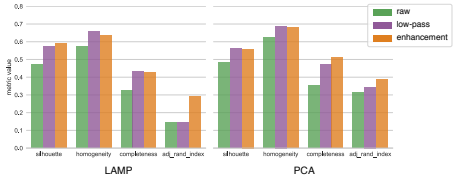
Results presented in Section (5) attest the benefits of the proposed filtering schemes when used in combination with visualization methods. The provided qualitative and quantitative results shows that filtering mechanisms can be primordial in the context of visualization.

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 Figure (5), 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.

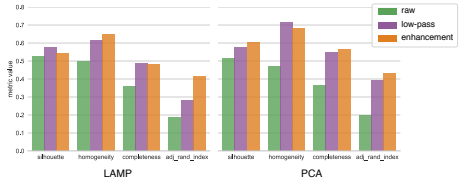
We consider the proposed methodology just a first step towards incorporating signal

(a) LAMP (b) PCA

Figure 10: 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.



(a) Ecoli



(b) Eggs

Figure 11: 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.

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 are 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 (4), the topology of the graph impacts directly in the result of the spectral filtering 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 [DTFV16] 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 visu-

alization purposes.

7 Conclusions

In this work 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.

References

[AA08] ANDRIENKO G., ANDRIENKO N.: Spatio-temporal aggregation for visual analysis of movements. In *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on* (2008), pp. 51–58.

[BCM05] BUADES A., COLL B., MOREL J.-M.: A non-local algorithm for image denoising. In *IEEE CVPR* (2005), vol. 2, pp. 60–65.

[BMWM06] 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 Trans. Vis. Comp. Graph. 12*, 5 (2006).

[DCVP*17a] 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 19*, 5 (2017), 85–91.

[DCVP*17b] DAL COL A., VALDIVIA P., PETRONETTO F., DIAS F., SILVA C. T., NONATO L. G.: Wavelet-based visual analysis of dynamic networks. *IEEE Trans. Vis. Comp. Graph. (preprint)* (2017).

[DK10] DASGUPTA A., KOSARA R.: Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1017–1026.

[dLVW95] DE LEEUW W. C., VAN WIJK J. J.: Enhanced spot noise for vector field visualization. In *IEEE Visualization* (1995), pp. 233–240.

[DTFV16] DONG X., THANOU D., FROSSARD P., VANDERGHEYNST P.: Learning laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing 64*, 23 (2016), 6160–6173.

[ES05] EBLING J., SCHEUERMANN G.: Clifford fourier transform on vector fields. *IEEE Trans. Vis. Comp. Graph. 11*, 4 (2005), 469–479.

[FD07] FREY B. J., DUECK D.: Clustering by passing messages between data points. *science 315*, 5814 (2007), 972–976.

[GK13] GRANLUND G. H., KNUTSSON H.: *Signal processing for computer vision*. Springer Science & Business Media, 2013.

[GO12] GASTAL E. S., OLIVEIRA M. M.: Adaptive manifolds for real-time high-dimensional filtering. *ACM Transactions on Graphics 31*, 4 (2012), 33.

[Goo16] GOODMAN R. W.: *Discrete Fourier and wavelet transforms: an introduction through linear algebra with applications to signal processing*. World Scientific Publishing Co Inc, 2016.

[HA85] HUBERT L., ARABIE P.: Comparing partitions. *Journal of classification 2*, 1 (1985), 193–218.

[HGW*16] HUANG W., GOLDSBERRY L., WYMBS N. F., GRAFTON S. T., BASSETT D. S., RIBEIRO A.: Graph frequency analysis of brain signals. *IEEE Journal of Selected Topics in Signal Processing 10*, 7 (2016), 1189–1203.

[Hot33] HOTELLING H.: Analysis of a complex of statistical variables into principal components. *J. Educat. Psych. 24*, 6 (1933), 417.

[JPC*11] JOIA P., PAULOVICH F., COIMBRA D., CUMINATO J., NONATO L.: Local affine multidimensional projection. *IEEE Trans. Vis. Comp. Graph. 17*, 12 (2011), 2563–2571.

[KKSS13] 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 19*, 12 (2013), 2926–2935.

[KZL*17] KANG X., ZHANG X., LI S., LI K., LI J., BENEDIKTSSON J. A.: Hyperspectral anomaly detection with attribute and edge-preserving filters. *IEEE Trans. Geosc. and Rem. Sens. 55*, 10 (2017), 5600–5611.

[LDC16] LU X., DENG Z., CHEN W.: A robust scheme for feature-preserving mesh denoising. *IEEE Trans. Vis. Comp. Graph. 22*, 3 (2016), 1181–1194.

[LHT17] LHUILLIER A., HURTER C., TELEA A.: Ffteb: Edge bundling of huge graphs by the fast fourier transform. In *PacificVis* (2017).

[Lic13] LICHMAN M.: UCI machine learning repository, 2013. URL: `http://archive.ics.uci.edu/ml`.

[LKS13] LINS L., KLOSOWSKI J. T., SCHEIDEGGER C.: Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Trans. Vis. Comp. Graph. 19*, 12 (2013), 2456–2465.

[LRBB17] LITANY O., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M.: Fully spectral partial shape matching. *Comp. Graph. Forum 36*, 2 (2017), 247–258.

[MM08] MCDONNELL K. T., MUELLER K.: Illustrative parallel coordinates. *Comp. Graph. Forum 27*, 3 (2008), 1031–1038.

[NH06] NOVOTNY M., HAUSER H.: Outlier-preserving focus+ context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 893–900.

[Opp99] OPPENHEIM A. V.: *Discrete-time signal processing*. Pearson Education India, 1999.

[PVG*11] 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 12* (2011), 2825–2830.

[RC96] REDDY B. S., CHATTERJI B. N.: An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Trans. Image Proc. 5*, 8 (1996), 1266–1271.

[RH07] ROSENBERG A., HIRSCHBERG J.: V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL* (2007), vol. 7, pp. 410–420.

[RHS15] 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* (2015), pp. 205–219.

[Rou87] ROUSSEEUW P. J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics 20* (1987), 53–65.

[SBS*17] SOLTESZOVA V., BIRKELAND Å., STOPPEL S., VIOLA I., BRUCKNER S.: Output-sensitive filtering of streaming volume data. *Comp. Graph. Forum 36*, 1 (2017), 249–262.

[SKLE03] 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* (2003), pp. 109–118.

[SRV16] SHUMAN D. I., RICAUD B., VANDERGHEYNST P.: Vertex-frequency analysis on graphs. *Applied and Computational Harmonic Analysis 40*, 2 (2016), 260–291.

[SSG95] SAKAS G., SCHREYER L.-A., GRIMM M.: Preprocessing and volume rendering of 3d ultrasonic data. *IEEE Computer Graphics and Applications 15*, 4 (1995), 47–54.

[VDP*15] VALDIVIA P., DIAS F., PETRONETTO F., SILVA C. T., NONATO L. G.: Wavelet-based visualization of time-varying data on graphs. In *IEE VAST* (2015), pp. 1–8.

[VKG03] VIOLA I., KANITSAR A., GROLLER M. E.: Hardware-based nonlinear filtering and segmentation using high-level shading languages. In *IEEE Visualization* (2003), pp. 309–316.

[VL08] VALLET B., LÉVY B.: Spectral geometry processing with manifold harmonics. *Comp. Graph. Forum 27*, 2 (2008), 251–260.

[VLL*10] VINCENT P., LAROCHELLE H., LAJOIE I., BENGIO Y., MANZAGOL P.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research 11* (December 2010), 3371–3408.

[Weg03] WEGMAN E. J.: Visual data mining. *Statistics in medicine 22*, 9 (2003), 1383–1397.

[ZVKD10] ZHANG H., VAN KAICK O., DYER R.: Spectral mesh processing. 1865–1894.