Icon and Geometric Data Visualization with a Self-Organizing Map Grid

Alessandra Marli M. Morais[1], Marcos Gonçalves Quiles[2], and Rafael D. C. Santos[1]

[1] National Institute for Space Research
Av dos Astronautas. 1758, CEP 12227-010, São José dos Campos, Brazil
[2] UNIFESP São José dos Campos
Rua Talim, 330, CEP 12231-280, São José dos Campos, Brazil
{alessandra.marli,quiles}@unifesp.br
rafael.santos@inpe.br

Abstract. Data Visualization is an important tool for tasks related to Knowledge Discovery in Databases (KDD). Often the data to be visualized is complex, have multiple dimensions or features and consists of many individual data points, making visualization with traditional icon- and pixel-based and geometric techniques difficult. In this paper we propose a combination of icon-based and geometric-based visualization techniques backed up by a Self-Organizing Map, which allows dimensionality reduction and topology preservation. The technique is applied to some datasets of simple and intermediate complexity, and the results shows that it is possible to reduce clutter and facilitate identification of associations, clusters and outliers.

Keywords: Visualization, Kohonen Self-organizing Maps

1 Introduction

Recent technological advances in information technology allows the collection and storage of large amounts of data almost effortlessly [1]. Data is collected as much as possible, without much thought at the collecting criteria because it is expected that it may be a potential source of valuable information after analysis [2]. However, finding valuable information from collected datasets (a process known as Knowledge Discovery in Databases or KDD) is often a non-trivial task.

Data visualization techniques has been used in KDD [3] as an attempt to make human beings an integral part of the data analysis process [1], nevertheless, creation of good visual representations which help and maximize data comprehension is not easily achievable. A subjectively good visual representation of a dataset depends largely on the data itself and is still a largely intuitive and ad-hoc process. Many of the well-established visualization techniques found in basic visualization tools (e.g. charts) do not provide adequate support for some

large, multidimensional and/or complex datasets [1], being adequate only for summarization of the data.

Two of the problems inherent to visualization of large datasets are clutter, which happens due to the need to display many data points at once; and projection/representation, which are techniques required for the visualization of multiple-dimensional data by reducing the number of dimensions for visual representation.

The Self-Organizing Maps (SOM), a neural network algorithm proposed by Teuvo Kohonen [4], is a technique that can be used to attempt to solve these two problems. The SOM is composed of a matrix or lattice of cells, each representing a prototypical data point that can be considered a centroid of a cluster of the original data. The data quantization capabilities of the SOM allows the creation of subsets of the data (reducing clutter). The SOM also creates a view of the data corresponding to a reprojection of the original dimensions into a smaller number of dimensions, usually two, allowing the exploration of the relation between cells, for similarity and dissimilarity identification.

Each cell in a SOM lattice can be used as a basis for well-established visualization techniques or new techniques more adequate to the problem and data. Integration of the SOM features with the particular visualization techniques can improve the identification of patterns in datasets when compared with the traditional visualization techniques alone.

This paper presents techniques for multidimensional and/or large datasets that uses the Kohonen Self-organizing Map as basis to cluster and reorganize data vectors. The paper is organized as follows: Section 2 introduces the main concepts about visualization. Section 3 describes the Self-Organizing Maps algorithm and its use in visualization tasks. Section 4 presents some examples of the technique. Finally, Section 5 presents conclusions and directions for future work.

2 Visualization

Visualization is usually described as the mapping of data to a visual representation. The visual representations, if well constructed, can be useful not only to present the information quickly to users, but also to help and maximize data comprehension [5]. Data visualization can be used for Exploratory Analysis, Confirmatory Analysis and Presentation [6].

A good visual representation enables the processing of several items of information at the same time, thanks to the human visual system and brain which is able to process a huge amount of information simultaneously and which is more efficient to extract information from visual representation than from an amount of numbers, text or a combination of these [5].

According to Tufte [7], a good visual representation is one which shows complex ideas with clarity, precision and efficiency. Graphical excellence is what gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space telling the true and avoiding misinterpretation

about data. However, there is no universal process to be done that ensures these requirements.

Defining a mapping that will result in a good visual representation is not a trivial task. The mapping depends largely on the task being considered and it is still a largely intuitive and ad-hoc process. It should feature easy understanding, avoid complex captions and undesirable visualization characteristics like occlusions and line crossings, that might appear as an artifact limiting the usefulness of the visualization [8].

The challenges and issues of a good mapping are significant when the data to be visually represented is large, highly dimensional or complex. The concept of dimensionality and size refers respectively to the amount of characteristics (attributes) which describe a data item of the data set and the total amount of items which comprise this set. There is no precise rule on what characterizes a high-dimensional and large data set [3]. However, despite of the powerful combinations of human brain and visual perception ability and the technological advances, there are some limitations on our capacity to deal with dimensions and the amount of data items that can be displayed.

The human vision is easily able to deal with two and three dimensions, so methods to visualize data with dimensionality greater than three often involves projecting the information into fewer dimensions. This needs to be done with care since some approaches to do it may not maintain the existing topological relationships between data in the original feature space, which is an important feature in the analysis process [9]. For visualization of large datasets, one must also consider a hardware-imposed limitation on the amount of data items that can be displayed on paper or on screen [1].

Even though visualization has an important role in KDD, the major problem that hampers the use of data visualization in KDD is that many of the well-established visualization techniques are not effective when applied to complex data sets [3]. Daniel Keim [10] describes some techniques for visualizing large amounts of multidimensional data. According to the taxonomy described by Keim and Kriegel [10], examples of visualization techniques are: Geometric Techniques (Parallel Coordinates, Scatterplot Matrices and others), Icon Techniques (Chernoff faces, Color Icons and others), Pixel-Oriented Techniques (Recursive Patterns, Pixel Bar Charts and others), Hierarchical (n-Vision, dimensional stacking and treemaps), Graph (Hy+, Margritte and SeeNet), and hybrid approaches.

Oliveira and Levkowitz [3] describe some characteristics considering strengths and weaknesses of those visualization techniques regarding the dimension and size of a given data set, but as mentioned these techniques may not be effective when applied to large datasets or datasets with multiple dimensions. For these cases, techniques that allow dimensionality reduction and grouping to remove clutter may be useful.

3 The Self-Organizing Map

The Self-Organizing Maps (SOM) [4] is a neural network algorithm based on unsupervised learning. It implements an orderly mapping of high-dimensional data into a regular low-dimensional grid, compressing information while preserving the most important topological and metric relationships of the data item on the display [11].

The SOM consist of M units located on a regular low-dimensional grid which represent the map. The grid is usually one- or two-dimensional, particularly when the objective is to use the SOM for data visualization. Each unit j has a prototype vector $m_j = [m_{j1}, ..., m_{jd}]$ in a location r_j, where d represent the dimension of a data item. The map adjusts to the data by adapting the values of its prototype vectors during the training phase. At each training step t a sample data vector $x_i = [x_{i1}, ..., x_{id}]$ is chosen and the distances, usually Euclidean distance, between x_i and all the prototype vectors are calculated to obtain the best-matching unit (BMU), c_i:

$$c_i = argmin_j\{||x_i - m_j(t)||\} \tag{1}$$

Once the closest prototype vector is found its values and the values of its neighborhood prototype vectors are updated, moving them toward x_i. The update rule is:

$$m_j(t+1) = m_j(t) + \alpha(t)h_{c_j}j[x_i - m_j(t)] \tag{2}$$

where t is the training step index, $\alpha(t)$ is the learning rate at moment t and $h_{c_j}j$ is a neighborhood centered winner unit (the best-matching unit). The winner unit is updated with a rate while its neighborhood is also updated with a smaller rate, determined by the distance on the map grid $||r_{c_i} - r_j||$, for example, by a Gaussian:

$$h_{c_j}j(t) = e^{-\frac{||rc_j - r_j||^2}{2\sigma^2(t)}} \tag{3}$$

where r_{c_j} and r_j are positions of units c_i and j on the SOM grid and $\sigma(t)$ is the neighborhood radius.

The algorithm produces two interesting characteristics to complex data set visualization: Quantization and Projection. The quantization result in a tentative to find a set of prototype vectors which reproduce the original data set as well as possible, while the projection try to find low dimensional coordinates that preserve the distances (or the order of distances) between the originally high-dimensional data [12]. The SOM algorithm has proved to be especially good at maintain the topology of the original dataset, meaning that if two data samples are close to each other in the grid, they are likely to be close in the original high-dimensional space data [12].

3.1 Self-Organizing Maps as Visualization Tool – General Ideas

The methods for visualizing data using SOM at literature can be grouped in three categories based on the goal of visualization [12]: methods that get an idea of the overall data shape and detect possible cluster structure, methods that analyze the prototype vectors and methods for analysis of new data samples for classification and novelty detection purposes.

At the first category some projection techniques as Principal Component Analysis (PCA) and Sammon's projection are used to visualize the shape of the SOM in the input space. These approaches plot each prototype vector in a two or three dimension as dots, for example, colored according to its position on the SOM grid and connected with its neighbors by lines, giving an informative picture of the global shape [13] and may exposing the clusters' structures. A more efficient technique to show cluster structure is the distance matrix, being the Unified Distance Matrix (U-Matrix) one of the most used [14]. The U-Matrix is a visual representation of the SOM to reveal cluster structure of the data set. The approach colors a grid according to the distance from each vector prototype and its neighbors: dark colors are chosen to represent large distances while light colors correspond to proximity in the input space and thus represent clusters.

Analysis of prototype vectors is a wide field whose approach is to visualize data attributes using the prototype vectors in order to get some insights about the spreading of values, cluster properties and correlations between attributes. According to Vesanto [12] the component plane plays the key role in the analysis of the prototype vectors. The component plane consist of a SOM grid where each map unit represent one prototype vector attribute colored according to the value of this attribute. The analysis can be done by plotting one component plane or all of them side by side. Other uses are the plotting of vector prototype attributes using some simple visualization technique like scatter plot matrix and by plotting of vector prototype attributes on its grid position.

Methods from the third group can give some insights about which prototype vector corresponds to a given new data vector, how accurate it is and if the new data vector really belongs to the abstract model created by the SOM. In order to detect this the response of the prototype set to the data sample has to be quantified and visualized. Some visual representations based on histograms are described by Vesanto [12].

4 Visualization using the SOM and Icon and Geometric Techniques

The basic principle of using the SOM grid for visualization is to create a graphical representation associated with each prototype (neuron), then displaying these graphical representations as a matrix. Each component of the matrix can be based on a traditional e.g. icon-, pixel- or geometric-based representations, or specific visualization techniques.

It is expected that the general appearance of the composition of the graphical componentes on the SOM matrix will yield interesting visual clues to the nature

of the data and reduce clutter, and at the same time the organization of the components will allow comparison between sets of similar data (in one component) to data that is somehow similar but to a lesser extent (in neighbor components). Due to the SOM properties the visual representation obtained will obey some of the principles posed by Tufte [7], particularly, show the data, induce the viewer to think about the substance rather than about the methodology, avoid distorting what data has to say, present many data in a small space, make large data sets coherent, encourage the eye to compare different pieces of data and reveal the data at several levels of detail.

In order to demonstrate the technique we've developed an API (Application Programming Interface, or a set of classes) using the Java language that allows the creation of a SOM based on input data and the creation of graphical components to represent the data visually.

A very simple example that demonstrate the approach can be seen in Figure 1. In this example, a synthetic dataset with four Gaussian blobs in a two-dimensional dataset is processed by the SOM. In order to visualize the prototype vectors (the neurons) we developed a simple component that displays at the same time the original dataset points (using small dark gray dots); the prototype vector (neuron) as a light, large blue dot, on the position determined by its values. The data points that can be considered as assigned or belonging to that neuron are also displayed as medium blue dots.

Each component displays also the number of data points assigned to that neuron. The visualization components are displayed in a grid, corresponding to the SOM architecture, in this case a 4×4 rectangular lattice.

The plot shown in Figure 1 illustrate the main concepts of using the SOM for visualization: an overview of the grid shows clusters and how the neurons' prototypes approximates the centers of the clusters. A more detailed view shows possible groupings of clusters, e.g. for the large blob in the center of the distribution, which is represented by several neurons which are contiguous due to the topology-preservation capability of the SOM.

It is important to point that this first example is deliberately simple, with two-dimensional data being mapped to two dimensions, but the technique could easily deal with more data and/or more dimensions, provided that the component used for visualization could represent multiple dimensions. That will be demonstrated by other examples in this section.

4.1 Example: Geometric and Icon Techniques applied to the display of the Iris dataset

Although there is no single rule to develop a good visual representation, some well-established visualization techniques may give a first insight about the data and works as a way to construct a better visual representation. For visual representation of data processed by a SOM, geometric and icon techniques can be good starting points.

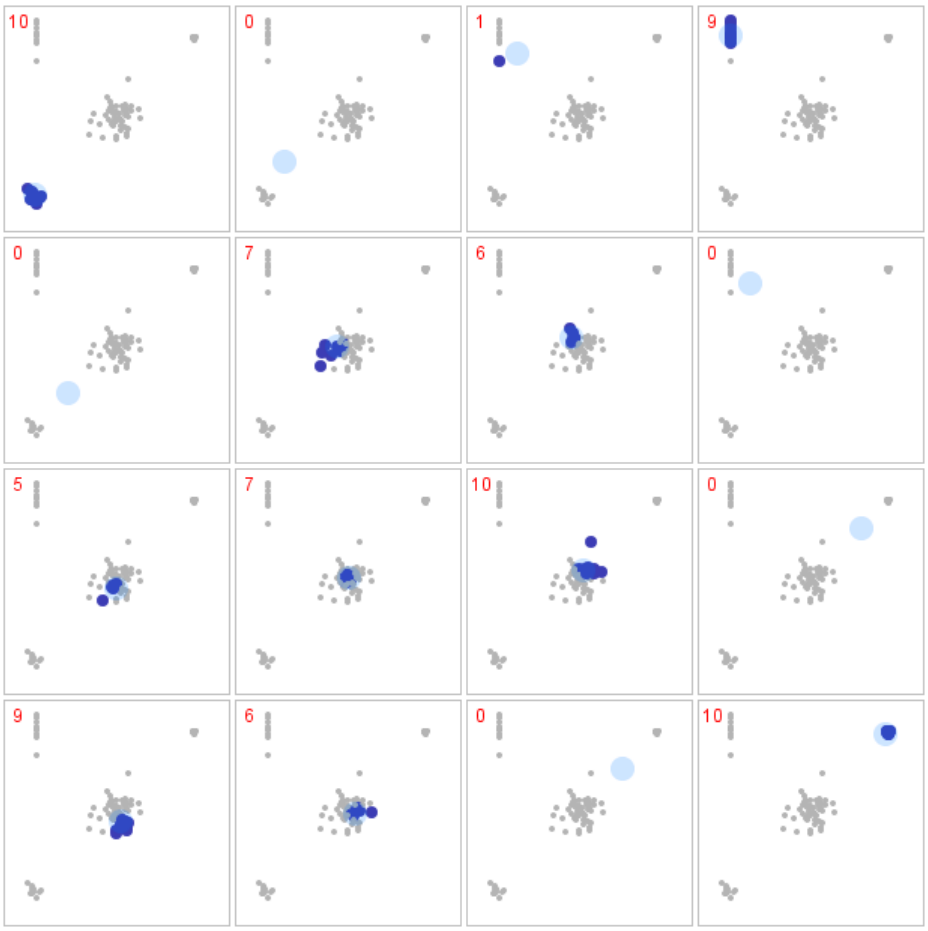

Fig. 1. A visual representation of synthetic data using a 4x4 SOM grid

Geometric techniques consist of mapping the data of the attributes on a geometric space [5], usually by finding an interesting projection of data set dimension [10].

Inselberg's parallel coordinates [15] is an example which maps k-dimensional data to two-dimensional surface by using k equidistant parallel axes where each axes correspond to one dimension and may be scaled according to the minimum and maximum value of this dimension; one data item is presented as a polygonal line, intersecting each axes at the point which corresponds to the value of considered dimension. The parallel coordinates technique is effective for detecting outliers and correlation amongst different dimensions [3] but may suffer from overlap and occlusions of lines when used to display large datasets.

To give an example of the capabilities of the proposed API we will explore some visualization techniques applied to the popular Iris dataset [16]. This

dataset is formed by 50 samples of each Iris plant species: Iris Setosa, Iris Virginica and Iris Versicolor. Each sample contains four numerical values: sepal length, sepal width, petal length and petal width. This set is known to have two easily separable clusters, being one formed by the samples of Iris Setosa and the other formed by the samples of Iris Virginica and Iris Versicolor. It can not be considered as a complex data set, nevertheless, it is sufficient to demonstrate the concepts.

Figure 2 shows the Iris dataset plotted using a plain Parallel Coordinates plot. Each polygonal line crosses the four vertical axes, each representing a dimension on the dataset.

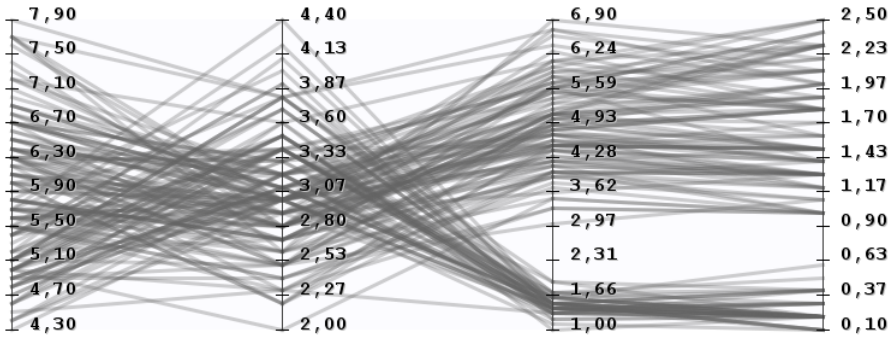

Fig. 2. Iris data set visualized by Parallel Coordinates

Figure 2 represents the 150 data points on the dataset – not a very large amount of data, but even so it is possible to see correlation between some attributes and some grouping, even with minor occlusion problems. Groupings can be made more evident and the occlusion problems can be reduced by preprocessing the data with a SOM and plotting it on a grid, as shown in Figure 3.

In Figure 3 the prototype vector is displayed by dark polygons while the light ones represent the data items mapped by it. The number on top of each grid unit show the amount of data represented by the neuron at this grid SOM position. It is possible to observe a grouping of data with similar attributes in four elements in the lower right corner (corresponding to the Iris Serosa samples), and that in general that clutter was reduced. It is also important to point out that the component that displays a single parallel coordinate within the SOM was designed to be more concise, i.e. without labels and other information that could clutter the display composition itself.

Icon-based visualization techniques are based on the association of attributes with features in a geometric figure so the features' values will determine the general appearance of the geometric figure as a whole [5]. Icon-based visualization techniques can be used to allow the instinctive comparison of similarities and differences.

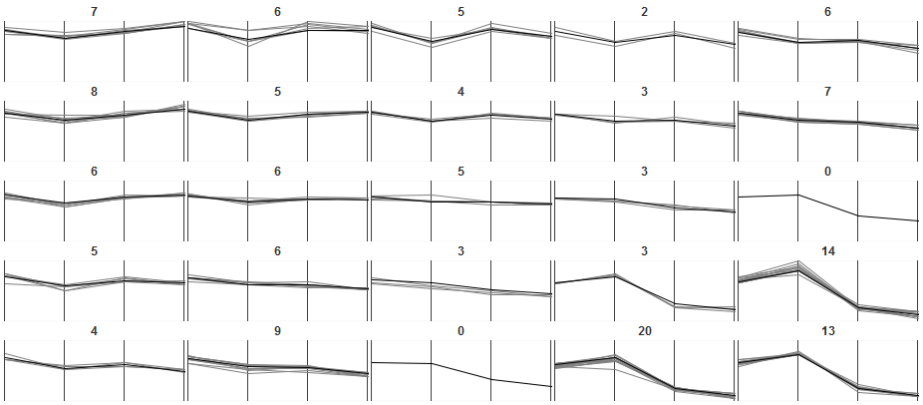

Fig. 3. Iris data set visualized by Parallel Coordinate over a SOM grid

Chernoff faces are one example of this technique: in this approach each dimension is mapped to the properties of a face icon, i.e, shape of face, nose, mouth and eyes. This technique can deal with multidimensional data, but its interpretation require training and each dimension may be treated differently by the human visual perception. The combined use with SOM is not able to minimize its weaknesses, but cluster and outliers detection may be improved.

Figure 4 shows the same Iris dataset preprocessed by a SOM and plotted over a grid, with each element on the grid being a Chernoff face representation of the prototype vector (neuron) and data associated to that vector. The prototype vector is displayed int dark lines while the light ones represent the data items mapped by it. The number on top of each grid unit show the number of data vectors represented by the neuron at this SOM grid's position.

In the example shown in Figure 4 the length and width of the sepal are mapped to the shape of mouth and nose and the length and width of petal are mapped to the shape of face and eyes. The cluster (group of neurons) that represents the samples of Iris Setosa can easily be seen in the bottom right corner of the SOM grid. Another interesting feature that can be visualized with the clustering of Chernoff faces is the difference between the prototype vector and data associated with this vector, shown in the neurons in the top left corner, meaning that the data is clustered in those neurons but with some variance on the features' values.

4.2 Example: Geometric Techniques applied to display of a medium-sized time series dataset

As a second example of our approach, we've developed a tool to visualize a collection of time-series representing the accumulated precipitation measured by a sensor network in São Paulo state, Brazil [17]. Each entry on this dataset is composed by a time series with two years of monthly average precipitation as

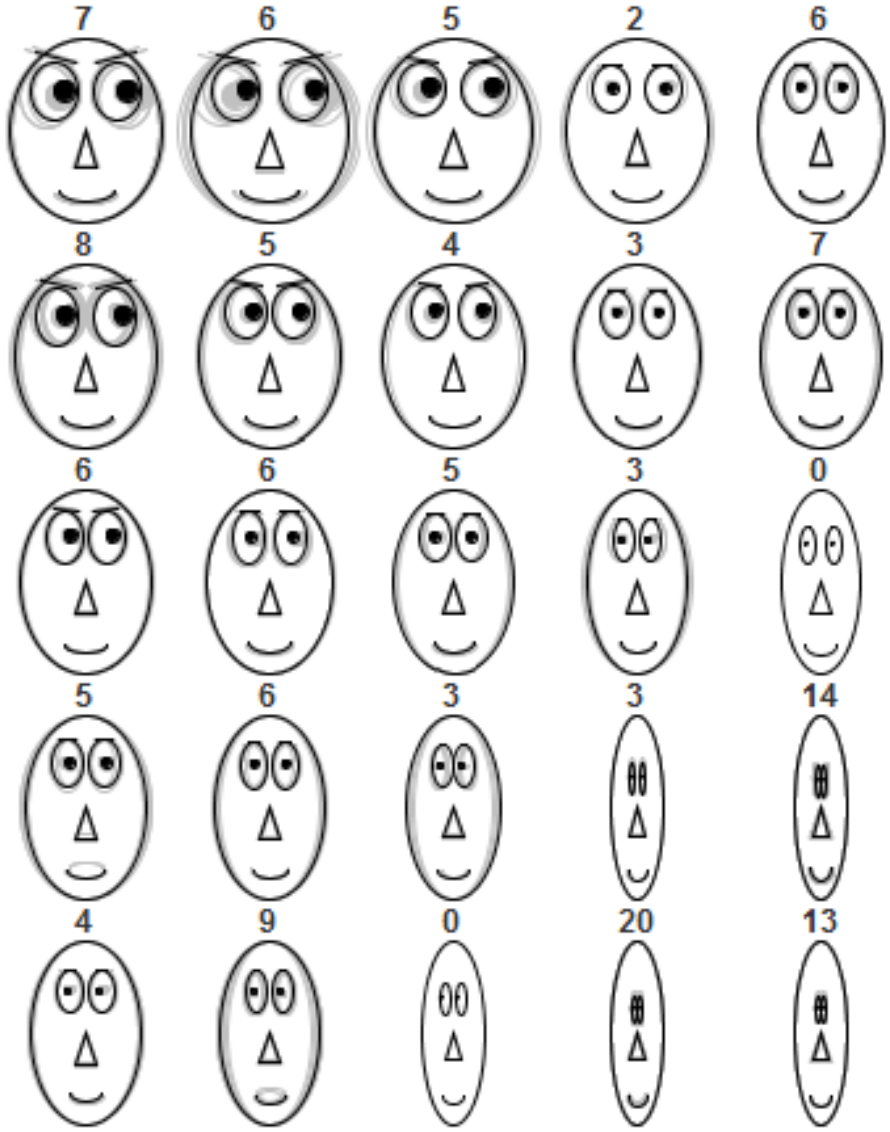

Fig. 4. Iris data set visualized by Chernoff faces over a SOM grid

measured by each data collection platform in the sensor network, in a total of 1340 time series.

Figure 5 shows a parallel coordinates plot of the 1340 time series. In that figure it is possible to identify some seasonal patterns, roughly corresponding to

dry and wet seasons. It is also possible to identify several outliers and months with more and less variance on the average precipitation.

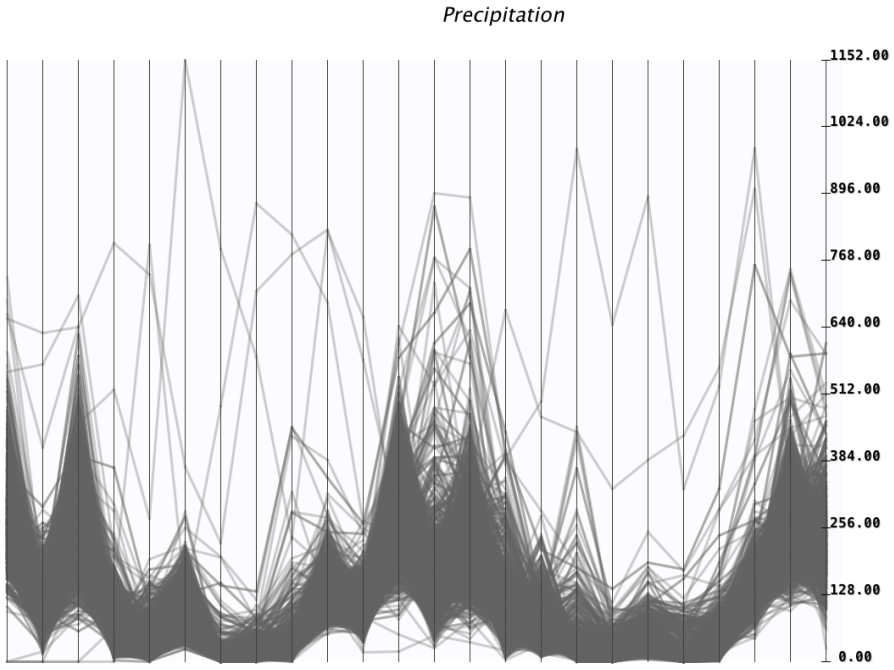

Fig. 5. Visual representation of 1340 time series by a Parallel Coordinates plot

While Figure 5 allows the visualization of some interesting features, it also presents a lot of clutter, making very hard the identification of groups on the dataset. Figure 6 shows the same data, grouped and reprojected into a SOM grid of 7×7 neurons.

In Figure 6 we can see the time series clustered into several loosely-defined groups, with the thick blue line representing the prototype vector (neuron) and with the data associated to that vector represented by thinner gray lines. In this example we don't use dimensionality reduction in the visualization, but the neurons are grouped in the grid accordingly to projections of the original data – in other words, similar time series are supposed to be represented by the same neuron or, if not much similar, by neighbor neurons.

The majority of the time series that follow a similar pattern are loosely grouped in the top left corner of the SOM grid shown in Figure 6. Some neurons clearly grouped time series that can be considered outliers, e.g. the neuron on the sixth row, last column and the one on the last row and first column. Reduction of clutter allowed the visualization of the prototypes and variation of the time

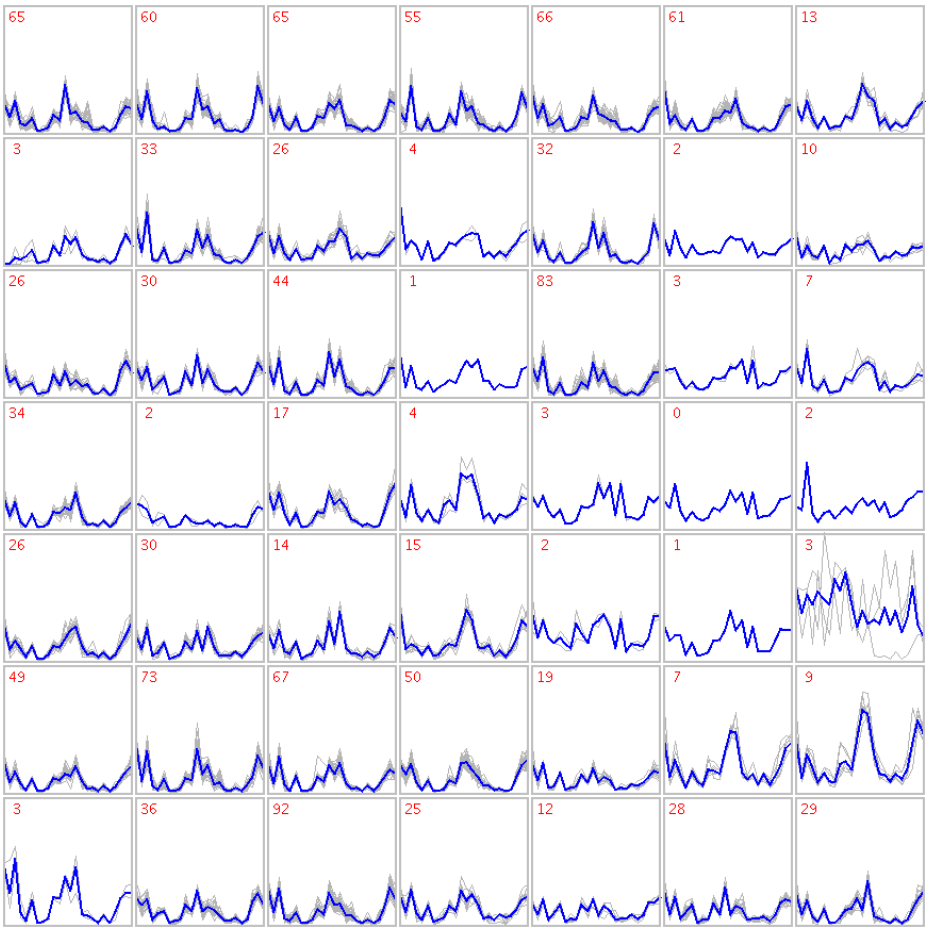

Fig. 6. Monthly rainfall visualized by Parallel Coordinates and SOM grid. The number on top of each grid unit show the amount of data represented by the neuron at this grid SOM position.

series assigned to those prototypes, which would be impossible to see in the original Parallel Coordinates plot.

5 Conclusions and Future Work

In this paper we presented a technique for data visualization comprised of traditional icon- and geometry-based representations organized in a SOM grid or lattice. This technique allows the visualization of features presented and recognizable in groupings of the dataset, making easier the identification of common patterns, outliers and reducing clutter for data plots with several data points.

Future work will be towards a better, more flexible implementation of the underlying SOM algorithm, particularly allowing the use of hexagonal grids and different distance metrics. We are also working on an API (Application Programming Interface) to facilitate the development of visualization tools based on this technique. Our decision on an API instead of a final tool is based on the argument that some visualization techniques are very specific to the data being analyzed, therefore it is better to allow an user the development of a specific component than to force him/her to use an existing one. Nonetheless the API will provide several ready-to-use visualization components.

Acknowledgements Rafael Santos would like to thank FAPESP, the São Paulo Research Foundation, for its support (grant 2014/05453-6). Alessandra M. M. Morais would like to thank MCTI/CNPq for its support (PCI grant 302428/2013-5).

References

1. Keim, D.A., Kriegel, H.P.: Issues in visualizing large databases. Springer (1995)
2. Keim, D.A., Sips, M., Ankerst, M.: Visual data-mining techniques. Bibliothek der Universität Konstanz (2004)
3. De Oliveira, M.C.F., Levkowitz, H.: From visual data exploration to visual data mining: A survey. Visualization and Computer Graphics, IEEE Transactions on **9**(3) (2003) 378–394
4. Kohonen, T.: Self-organizing maps. Volume 30. Springer (2001)
5. Mazza, R.: Introduction to information visualization. Springer (2009)
6. Ankerst, Mihael; Grinstein, G.K.D.A.: Visual data mining: Background, techniques, and drug discovery applications. In: The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2002)
7. Tufte, E.R., Graves-Morris, P.: The visual display of quantitative information. Volume 2. Graphics press Cheshire, CT (1983)
8. Keim, D.A.: Visual exploration of large data sets. Communications of the ACM **44**(8) (2001) 38–44
9. Penn, B.S.: Using self-organizing maps to visualize high-dimensional data. Computers & Geosciences **31**(5) (2005) 531–544
10. Keim, D.A., Kriegel, H.P.: Visualization techniques for mining large databases: A comparison. Knowledge and Data Engineering, IEEE Transactions on **8**(6) (1996) 923–938
11. Kohonen, T.: The self-organizing map. Neurocomputing **21**(1) (1998) 1–6
12. Vesanto, J.: Data exploration process based on the self-organizing map. PhD thesis, Helsinki University of Technology
13. Koua, E.: Using self-organizing maps for information visualization and knowledge discovery in complex geospatial datasets. Proceedings of 21st International Cartographic Renaissance (ICC) (2003) 1694–1702
14. Gorricha, J., Lobo, V.: Improvements on the visualization of clusters in georeferenced data using self-organizing maps. Computers & Geosciences **43** (2012) 177–186
15. Inselberg, A.: Parallel Coordinates – Visual Multidimensional Geometry and Its Applications . Springer (2009)

16. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of eugenics **7**(2) (1936) 179–188
17. Garcia, J., Monteiro, A., Santos, R.: Visual data mining for identification of patterns and outliers in weather stations' data. In Yin, H., Costa, J., Barreto, G., eds.: Intelligent Data Engineering and Automated Learning - IDEAL 2012. Volume 7435 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2012) 245–252 10.1007/978-3-642-32639-4_30.