Outlier-preserving Focus+Context Visualization in Parallel Coordinates

Matej Novotný
Comenius University Bratislava
mnovotny@fmph.uniba.sk

Helwig Hauser
VRVis Research Center, Vienna
Hauser@VRVis.at

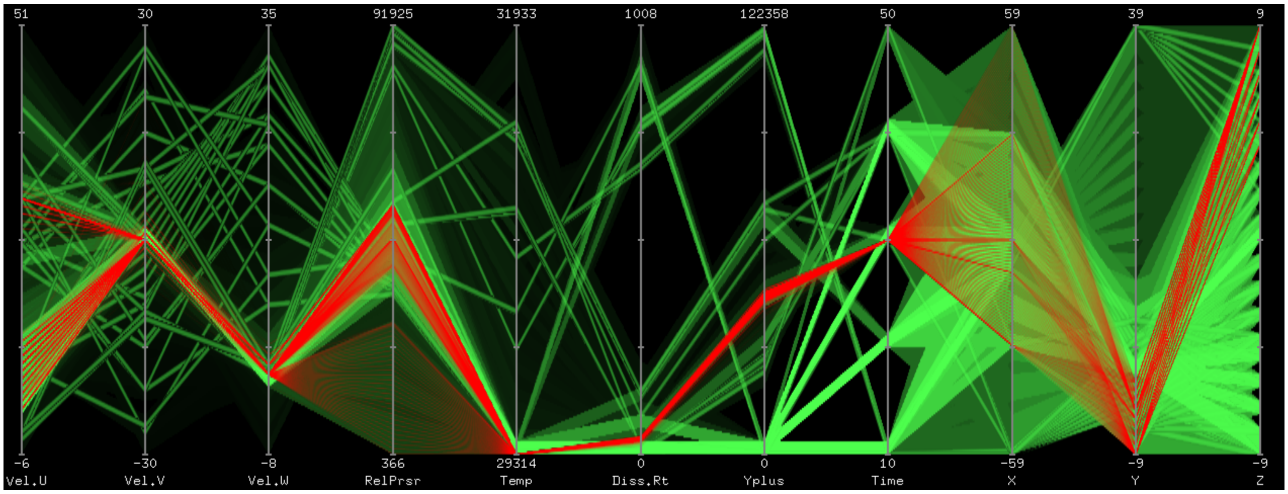

Figure 1: Outlier-preserving focus+context visualization of a CFD simulation dataset (the mixture of two fluids). The outlier-preserving context visualization shows that with respect to flow directions (the three axes on the left), most data items cluster around the zero-values in the *v* and *w* component of the flow vector, whereas quite different values of the *u* component show up. We can also see that a number of visualization outliers with respect to flow velocities, pressure values, temperatures, etc., significantly contribute to the visualization. Finally, the focus visualization (red polylines) reveals more multi-variate details for a data subset which is characterized by low temperature and relatively low spatial *y* values.

Abstract

Focus+context visualization offers convenient solutions for specifically steering the investment of graphical resources in visualization so as to emphasize selected subsets of the data while at the same time also preserving a good overview through context visualization. In focus+context visualization, the context often is represented in a reduced and/or compressed form which can cause problems for small-scale features in the context such as data outliers.

In this paper we present an approach to focus+context visualization in parallel coordinates which is truthful to outliers in the sense that small-scale features are detected ahead to visualization and treated specially during context visualization. We introduce outlier detection and context generation to parallel coordinates on the basis of a binned data representation which leads to an output-oriented approach which only processes those parts of the visualization which actually affect the final rendition. The resulting solution is capable of producing context at several levels of abstraction and considers the outliers individually. By exploiting data binning, the performance of the algorithms is not really decreased by the size of the original data and outperforms the standard visualization technique of parallel coordinates. The system was successfully tested with datasets of up to 3 million records or 50 dimensions.

Keywords: Information visualization, parallel coordinates, focus+context, outliers, large data visualization.

1 Introduction

Visualization is established as a very useful approach to the exploration, analysis, and presentation of large and/or complex datasets. The extremely powerful human visual system, with its extraordinary capabilities of efficient parallel information acquisition and also processing, is exploited as a broad-band information access channel between the digital datasets and the human mind [16]. However, and even though this size-of-data argument is ubiquitously used to motivate data visualization of these days, also visualization has its bounds and limits when it comes to unboundedly increasing datasets.

Whereas the size limitations are an important topic in scientific visualization, not at the least due to performance challenges associated with very large datasets, this issue of size limitations especially also shows up in information visualization where many visualization techniques exist which not really are very well suited for really large datasets. This is especially true for any kind of information visualization where relatively large amounts of screen space are invested to represent individual data items, e.g., due to their high dimensionality or due to the representation of complex relations between the data items. Examples are glyphs-based data representations [3] as well as parallel coordinates [6, 5] which are known to give very good insight into the multi-dimensional character of a dataset, but at the cost of investing a relatively verbose visual representation per data item (a glyph, a polyline).

Without any special solution applied, parallel coordinates usually are limited to about a few tens of thousands of data items per view. With hundreds of thousands of data items already, parallel coordinates usually are heavily overplotted and the resulting screens are of almost no use at all. To accommodate with this issue of large

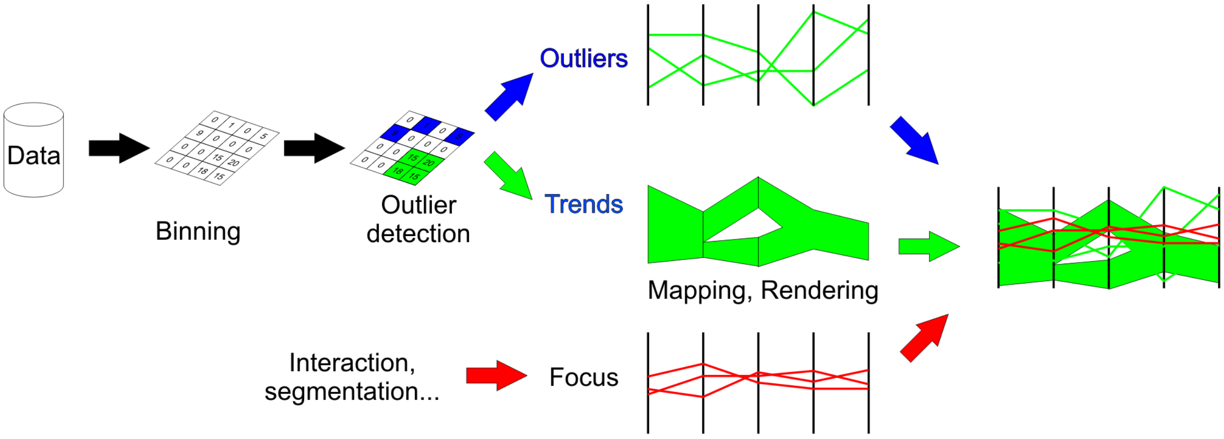

Figure 2: The workflow for outlier-preserving focus+context visualization in parallel coordinates.

data visualization in general, different approaches have been proposed in the literature, ranging from the visualization of selected subsets of the data only (selection), over the visualization of aggregated data (aggregation), to the visualization of dataset segmentations (segmentation) - see the works of Daniel Keim et al. for an in-depth overview of these solution categories [8]. Related to the aggregation approach, and also related to similar approaches in scientific visualization [18], image processing [17], and many other fields, the idea of different levels of details in visualization (in conjunction with a hierarchical data representation) is very interesting. Hierarchical parallel coordinates, for example, are a very neat approach to accommodate large data within the great approach of parallel coordinates [4].

In this paper, we adopt the approach of output-sensitive visualization technology to enable the visualization of really large datasets. Similar to scientific visualization, where image-order techniques are opposed to object-order techniques [9] when the number of data items that are to be visualized significantly exceed the number of pixels which are to be filled on the screen, also in information visualization it is a promising approach to think of output-oriented approaches (instead of data item oriented ones). At the latest, when the number of data items significantly outnumbers the number of pixels to fill, it becomes more effectively to ask in which way a set of pixels is affected by the visualization instead of asking in which way a set of data items affects the resulting visualization. One way to do so is to convert the data into a frequency-based representation, e.g., in a histogram or a histogram-like representation. The technique of parallel sets [2], for example, integrates such a frequency-based data representation with the layout metaphor of parallel coordinates. Other related approaches also utilize frequency-based representation [11]. Also the recent development of PC graphics hardware and GPU programming opened the domain for approaches that process the graphical representation of parallel coordinates as textures [7] to enhance the view.

2 Basic Idea

The basic idea of our here presented contribution is the following. We construct an output-oriented data representation on top of the original data, which consists of one fairly fine resolved 2D bin map for every neighboring pair of axes in the parallel coordinates view setup. A polyline across all axes, representing one data item, contributes to exactly one bin per bin map, i.e., for every line segment of all the polylines one bin count (in one 2D bin map) is advanced by one.

Accordingly, every 2D bin map represents the line distribution between two neighboring axes in a frequency-based form. Based on these bin maps, we identify parts of the data which do not align with major other parts of the data, i.e., some sort of visualization outliers.

In statistical data analysis there are several concepts of data outliers [13], [12], often considered in a holistic sense, i.e., taking all the data dimensions concurrently into account. In our approach we identify *visualization outliers*, i.e., line segments which are isolated from the other dominating parts of the visualization. We separate visualization outliers to treat them separately during our output-sensitive visualization, which is useful because of the following consideration.

For rather coherent substructures of the data visualization which also represent a rather large number of individual data items, an aggregated visualization really makes sense since it is the natural synonym to just drawing loads of data items individually and waiting for the visualization to account for some kind of a low-pass filtering or smoothing, for example, through the use of semi-transparency in the visualization or the like. Visualization outliers, however, easily can get lost during such a process (they are smoothed away) or they account for a distorted visualization (prominent visualization cues pop up where only very few data items are referenced). For reasons argumented in Section 5 we believe this is an important issue. It is therefore useful, to first separate outliers and then treat them separately during visualization, e.g., by awarding them a separate visual representation (individual polylines instead of parallelograms of an aggregated visualization).

Since data visualization with parallel coordinates only unfolds all its potential when interaction is strongly supported, i.e., when axes can be reordered/repeated/scaled/flipped/distorted/etc. and when brushing and focus+context visualization is provided, we also incorporate means of interactive visual analysis within our prototype implementation. To do so, we employ an intelligent scheme of how and when to refer back to the large data and of when just to work with the bin maps.

The two main parts of the synergic approach – outlier separation and output-sensitiveness through data binning are explained in Sections 3 and 4. They way they cooperate and how is the actual outlier-preserving focus+context visualization in parallel coordinates produced is described in Section 6.

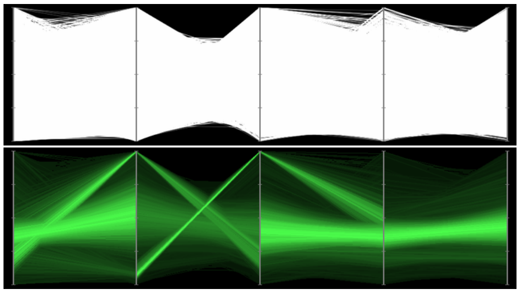

Figure 3: Remote sense data [15] (interpolated to 100.000 samples) rendered using conventional parallel coordinates (top) and after binning to 128×128 bins (bottom). Not only the binned representation is precise enough to preserve the details (note the width of a bin) it also clarifies the visualization thanks to the density-based nature of its.

3 Output-oriented Visualization

The recent development of data acquisition leads to high volumes of the visualized data. The growing ratio between the original information and its graphical on-screen representation brings up the question of how much does a data-oriented process affect the final rendition and in what ways. Aware of these facts, an output-oriented visualization can benefit from the knowledge that many actions or features in the data space do not change the visual appearance in a significant way. Hence, for purposes of visual exploration, these items require much less processing (often none at all) and can save the precious processing resources.

Not necessarily the data records have to be inspected for whether they do or don't affect the final rendition. By converting the data to an aggregated form, as presented in Section 4.1, the actual visual importance of the particular items reflects also in the aggregated form without having to decide on any per-item basis. We then approximate the original data by visual cues.

The resulting visualization differs only little (or not at all) from the output-insensitive one but it outperforms it significantly and it allows for better interaction. Moreover, and thanks to the intelligent design of the abstract information, the output-oriented demonstration is capable of communicating a clearer and less cluttered information for smaller processing expenses compared to the original one.

The Figure 3 shows a comparison of standard parallel coordinates and parallel coordinates displaying binned data. The output-oriented version communicates more information and renders much faster.

4 Data Binning

Generally, binning is a process during which the original data is converted to a frequency-based representation by dividing the data space into a set of multidimensional intervals – called bins – and assigning to every bin an occupancy value which determines the number of data records that belong to the bin [14].

Consider a case in which data of n dimensions is normalized to a $[0,1] \times [0,1] \times \ldots \times [0,1]$ interval. If each dimension is divided regularly into b intervals

$$\left[\frac{j}{b}, \frac{j+1}{b}\right[\qquad j = 0, 1, \ldots, b-1$$

then a data record $X = \{x_1, x_2, \ldots, x_3\}$ belongs to a bin $B_{j_1, j_2, \ldots, j_n}$ if the following holds

$$\frac{j_i}{b} \leq x_i \leq \frac{j_i + 1}{b} \qquad i = 1, 2, \ldots, n$$

The binning transformation preserves the data distribution and it replaces the (often large) data with the frequency-based binned representation. The total number of bins grows exponentially with respect to the number of dimensions and will produce enormous memory demands if a real multidimensional data had to be binned.

This unpleasant property of the binning holds not for the aggregated output-oriented approach presented here. The parallel coordinates plot is a projection that displays multidimensional data by placing drawing them onto a two-dimensional plane. Naturally, considering the visual output, we only perform a two-dimensional binning and aggregate the data in the following way:

For each pair of adjacent axes representing a pair of dimensions we bin the particular two-dimensional subspace into $b \times b$ bins. The resulting set of bins forms a so called bin map and can be thought of as a two-dimensional histogram of the subspace. Following the nice distribution-preserving properties of binning, this creates a frequency-based, output-oriented, representation of the original data. In addition, such a representation occupies only a fixed amount of memory and does not depend on the size of the input data. The total number of bins is kept low and the current computer systems are capable of holding even the complete binned information of all possible two-dimensional subspaces for datasets as wide as 50 dimensions.

Utilizing binning in a visualization environment brings several major improvements. First the bin maps, and through them the whole approach, are well scalable. The size and the number of the bins determine the precision of the aggregation result. The bins are in many cases reusable without having to re-bin the original data. Hierarchical or adaptive binning is possible to improve the situation even more. The binning, as described in this solution, is a suitable choice for an aggregated output-sensitive visualization.

4.1 Two-dimensional Binning in Parallel Coordinates

The parallel coordinates plot consists of m parallel axes placed onto a two-dimensional plane. Each axis A_i represents a certain data dimension and utilizes a certain mapping transformation that maps a data value x_i to its screen value y_i. In the sense of the rendering, y_i is basically the position of value x_i on the axis A_i. Usually the default mapping function linearly scales the values to fit onto the axis. Some of the basic interaction options of the parallel coordinates include flipping the axis orientation or adjusting the mapping function so that it 'zooms' to a certain sub-interval within the data dimension. Therefore it is vital to bin the data after the mapping transformation. Otherwise the resulting binmap and the visualization derived from it would not correspond to the standard visual output.

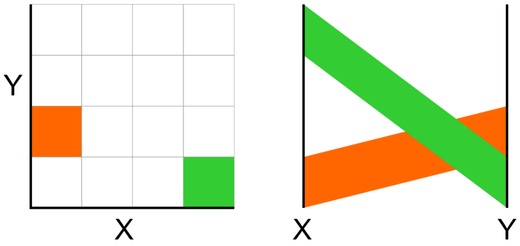

Figure 4: The binning map (left) containing two bins (orange and green) and the same bins visualized in parallel coordinates (right).

By dividing an m-dimensional parallel coordinates plot into a set of $(m-1)$ two-dimensional subspaces – each belonging to one pair of adjacent axes – the whole screen can be rendered using only a binned representation. How a certain bin from the binmap maps to its graphical representation in parallel coordinates is illustrated in Figure 4

5 Outlier separation and Clustering

There are two main reasons to separate outliers from the rest of the data. The first reason is that even though in many application scenarios the outliers are considered as a flaw in the data or as an error in the measurement, there are many other applications where the outliers actually attract a lot of attention and steer the decision process. For example network security administrators inspect an outlier in their data as a potential intrusion. Business experts might find outliers in business statistics an intriguing investment option [10]. Losing an outlier by merging it with the context means obstructing the visual exploration on its way to find interesting areas in the observed data.

Another motivation for handling outliers separately is the quality of data abstraction in focus+context visualization. Every data abstraction strives to find the effective balance between simplicity and truthfulness. By introducing outliers to the abstraction, the truthfulness of the abstraction is often significantly decreased and a more complex representation has to be chosen to compensate for that. Figure 6 shows an example of how outliers might mislead the generation of context for an focus+context visualization and how treating them separately makes the context more coherent with the original data.

5.1 Outlier Detection in Binned Data

Once the data is binned into a two-dimensional density-based representation it can be manipulated in a way that is analog to two-dimensional signal processing. Operating on such a basis, an outlier can be detected using the following scheme: A low-pass filtering is used first, then the result is compared to the original and checked for differences. Bins that are emptied after smoothing and quantization to the digital resolution of the bin representation are considered as outlier bins. A comparison of two outlier detectors – the isolation filter and the median filter – is depicted in Figure 5. In both cases those bins that have population below a certain population threshold and are detected by the filter are marked as outliers (depicted in yellow in the Figure.)

The population criterion is set to prevent small and highly populated clusters from being taken for outliers by mistake. Such features are well visible even in the binned visualization and do not need to be treated specially. On the contrary, treating them as outliers would decrease the quality of the visualization since it would remove a significant portion of the whole data from the processing that leads to context generation and/or clustering (see Figure 2.) During our experiments and in the illustrations here, the population threshold was set between 1 and 10 percent of the maximum population.

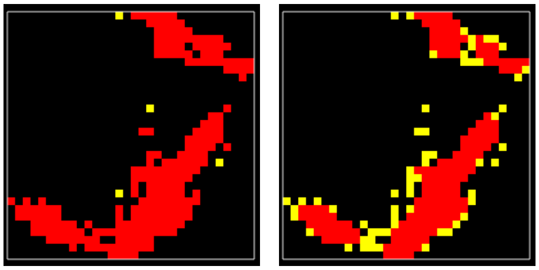

Figure 5: Two examples of outlier detection in a bin map. A simple neighborhood filter which detects isolated bins (left) and a more advanced median filter (right). The outliers are depicted in yellow.

The isolation filter considers a 3×3 support in the bin map and checks the occupancy values of the 8 bins that are adjacent to the central bin. If the number of empty neighbor bins is above a certain threshold (say 6 or 7) the central bin is declared an outlier. The threshold has to be adequately decreased for the bins on the borders of the bin map (e.g. 2 for the corners and 4 for the borders.)

Similarly, the median filter computes the median of occupancies of the neighbor bins and if it falls below the population threshold, the central bin is marked as an outlier.

Once the bins are separated into outlier bins and the bins containing the main data, the actual outlier extraction is performed. This is handled in the same way as brushing is handled in parallel coordinates. If a two-dimensional bin, which is basically a two-dimensional interval, is specified as $[b^i_{min}, b^i_{max}[\times [b^j_{min}, b^j_{max}[$ then its contents are extracted by queries on the original data asking for all the data records whose i-th attribute falls into the $[b^i_{min}, b^i_{max}[$ interval and j-th attribute falls into the $[b^j_{min}, b^j_{max}[$ interval.

This is performed on all the bin maps and the results are united into the final outlier sets. This set is then removed from the original data and is depicted using standard parallel coordinates poly lines.

5.2 Clustering

The here introduced clustering approach represents an interesting extension to the data binning and screen-oriented processing. A different form of clustering in the screen space of parallel coordinates (utilizing interaction and the grand tour) was introduced by Wegman and Luo [11].

The clustering presented in this paper performs in several steps. First the binned data is smoothed out using a Gaussian filter. As it is well known in the visualization domain [1], smoothing out the small-scale features (noise, outliers) and compacting the large features (clusters) improves the results of feature extraction and also of clustering process.

The smoothed out approximation of the original data is inspected starting with the bins of the highest population. The population threshold is iteratively decreased revealing either new cluster centers or expanding the already existing clusters. The process might run automatically to either search for a desired number of clusters or to finish at a given threshold limit. A reasonable choice is to set the limit to 5 – 10 percent of the highest occupancy in the bin map. Together with the domain knowledge and abstract thinking of the user clustering inside the bin maps provides an original way to observe the data structures and e.g. to visually detect axes with similar data distribution or to separate the data into structures using only several two-dimensional subspaces in the screen space instead of running a data-oriented clustering on the original vast and multidimensional data set.

6 Creating The Context

During the final stage of the here presented approach, the context is created in an output-oriented way. As the context usually describes a large portion of the data, it is naturally desired to map the data samples residing in the context to a simple graphical representation, i.e. an aggregated. This is achieved using the binned data representation as described above. Either the bins already used for outlier separation and cluster identification can be utilized or a different binning with a different precision can be computed.

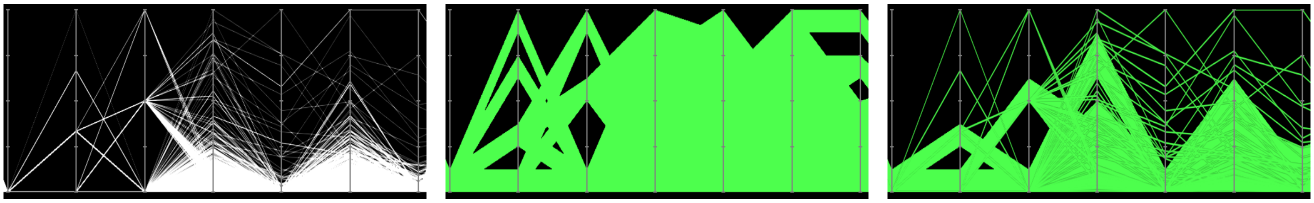

Figure 6: The original data (left) unnaturally stretch the context (center) because of the outliers in upper part of the plot. The context becomes more truthful with outlier preservation (right).

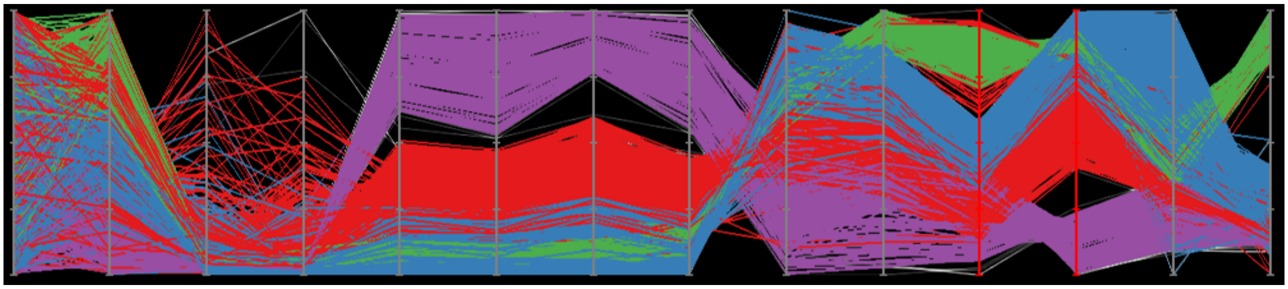

Figure 7: Clustering of the binned data performed between the 11th and the 12th axis. The according two-dimensional subspace was binned to 64×64 bins (as shown in Figure 8) and clustered according to the occupancy values of the bins. The distinct colors show four clusters extended to all the dimensions.

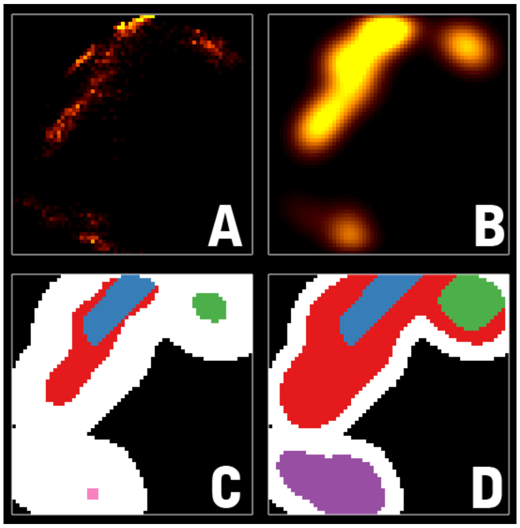

Figure 8: A scatterplot visualization of the bins and the clustering process in Figure 7.
A – Original binned data, occupancy is mapped to color lightness.
B – The binned data smoothed out using gaussian filter
C – An early stage of the clustering. The occupancy threshold is set to 80% of the maximum value. Unclustered data is shown in white, clusters are colored distinctly.
D – Final stage of the clustering. Threshold set to 10%.

Every bin is eventually rendered as a quadrilateral connecting a pair of adjacent axes with its vertexes placed on the respective positions of the minimum and maximum bin borders with respect to the particular axes. To keep the context part visually simple, only one color is used for all bins. To discriminate the dense areas from the sparse ones, the brightness of the color is used as an information channel though. Before actual rendering, the bins within a certain segment are ordered according to their ascending occupancy and rendered one over another so that the most populated bins are rendered on top of the others using the brightest color.

This approach compensates for disregarding semi-transparency of the quadrilateral depiction of the bins. It is popular and usually very helpful to incorporate semi-transparency to visualization. However for the purposes of generating a visually undemanding context in focus+context visualization the large number of visual attractors caused by overlapping translucent segment is not a suitable effect.

After the context is generated from the core part of the data, the outliers are rendered using the standard parallel coordinates technique, i.e., by drawing poly lines for all the outliers . Their color is chosen to be the same as of the context, so that they are not mistaken for focus items, that are also rendered in full detail using the usual poly lines representation.

6.1 Levels of Detail in Context

Although the context occupies only a small part of the visual information bandwidth between the computer display and the human mind, it does not necessarily need to be rendered at the lowest possible precision. By adjusting the level of details in the context the user can accommodate the precision of the context to his/hers actual needs or to the domain knowledge.

7 Demonstration and Results

The implementation of the concept has been done with large multivariate data in mind. Output-oriented rendering is embedded into the application in many ways. Moreover the demonstration introduces ideas for next research, like e.g. clustering in screen space or in the binned data representation.

7.1 Output-sensitive Implementation

To employ the concept of output-sensitive visualization in the most efficient way, three rendering modifications have been done to the original parallel coordinates plot in addition to the binning-related modifications presented in earlier parts of this paper. These rendering modifications strive to contain as much of reusable graphical information as possible, to save unnecessary rendering and data-to-screen mapping, both of which are demanding operations.

Layers The parallel coordinates plot is divided into several layers that contain different portions of the visual space. These portions are focus, context (without outliers), outliers, original poly line representation and clustering results. Different combination of these visual cues can be composed together via alpha blending without having to re-render any of them. Additional layers, such as data segmented out of the original data set, can be appended as well.

Segments To improve axes interaction, the screen is divided into rectangular segments, each one between a pair of adjacent axes. This tightly binds to the concept of two-dimensional binning and creates an effective framework in which only those segments have to be re-rendered that actually changed.

Rendering to texture Textures are nowadays used to store the results of visualization and rasterization because many interaction-triggered changes in the display are feasible to accomplish by transforming a texture containing the results of the previous rendering. Advanced and fast GPU-based processing in the texture can replace several complex data-oriented tasks [7]. In the presented prototype, each segment is first rendered to a texture to save the rendering results for future use and then the texture is rendered on screen.

In total, the parallel coordinates plot is divided into an array of *layers* × *segments* textures to allow for re-rendering only the necessary portions of the screen.

7.2 Context

The purpose of focus+context visualization is to put fine-scale details contained in a smaller portion of the data into focus while preserving their basic relations to the rest of the data – to the context. The solutions presented here enriches the context with additional information while keeping the nature of the context simple and easily readable. Therefore more relations between focus and the data inside the context can be observed.

An illustration of the contribution can be seen in Figure 9 The top picture shows a standard situation in a parallel coordinates environment. Focus (depicted in red) drawn over the rest of the data. Many parts of the plot are homogeneous and heavily overplotted. It is hard to decide on their intrinsic nature. By introducing the outlier-preserving focus+context visalization, a different view is produced and closer relations to the rest of the data can be drawn without putting too much additional load on the processing and visualization powers of the computer or on the perception abilities of the user.

Now it can be easily ascertained that e.g. the focus is eccentric with respect to the third axis and the main part of the data. On the contrary, it follows perfectly the behavior of the main data between the first two axes. These and other conclusions are not available at first glance using the original line-based rendering.

7.3 Information Increase

Due to overplotting in parallel coordinates the standard visualization might easily reach its capacity. In such cases the visualization

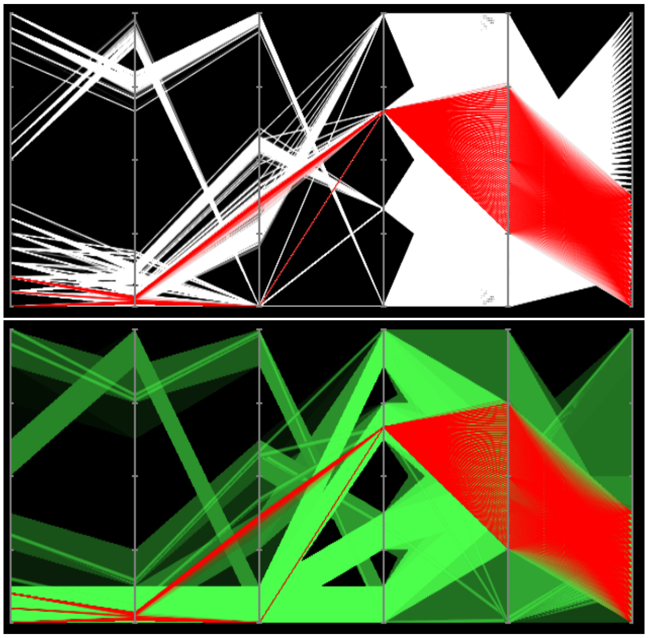

Figure 9: Standard focus visualization in parallel coordinates (top) and the same data configuration depicted using outlier-preserving focus+context (bottom). The structures inside the context are revealed and their relations to the focus can be observed now.

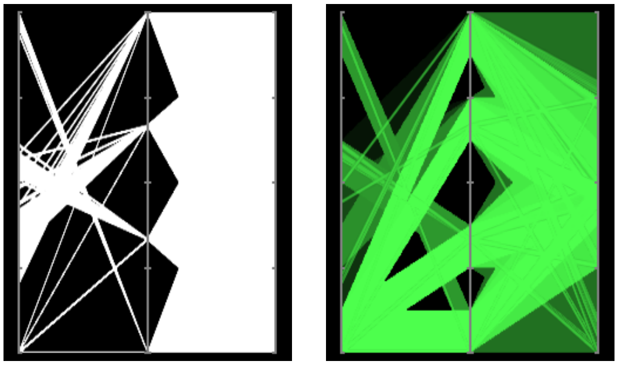

Figure 10: Original line-based rendering (left) could build an erroneous notion of the data distribution. The real distribution is better approximated by the density-oriented approach.

wrongfully steer the attention of the observer towards areas with higher graphical volume. If an intelligently aggregated information is used instead, the real nature of the data pops out.

A paradox as it is, a lower level of detail provided a higher level of information in the example depicted in Figure 10. Even though semi-transparency was used to render the original line-based display (left), the resolution of the alpha channel was soon exhausted. Therefore the area of high values in the first and the second axis draw a lot of visual attention. However the outlier-preserving context generated from the binned data (right) reveals that most of the data records reside in the lowest portions of the first axis. Also what seemed to be an even distribution on the third axis in the line-based display turns out to be a rather normal distribution when displayed using the binned data.

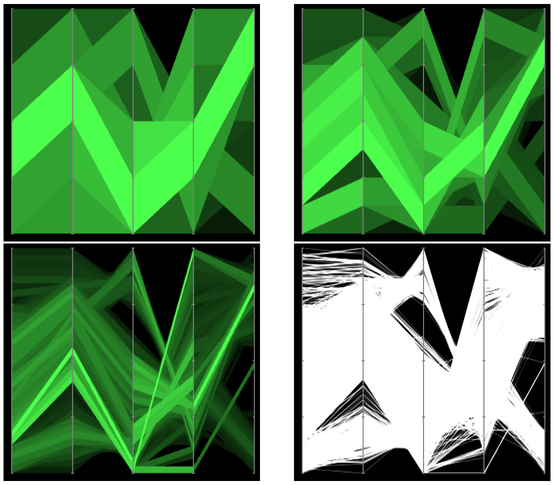

Figure 11: Different levels of detail for rendering the context.

7.4 Variable Context Precision

The context becomes much more flexible once we introduce binning (or data abstraction in general) to focus+context visualization. The context can either be depicted in the usual visually undemanding form, but it can also provide additional information for almost no costs with respect to processing or visual investments. Figure 11 shows a portion of a parallel coordinates plot depicted using several levels of detail. The top left image shows a possible trend. The brightest areas are those with the highest occupancy. When decreasing the size of the bins, the crude and large structure breaks up into finer ones.

7.5 Performance

In interactive exploration of large data through visualization, it is crucial to steer the performance of the application in a way that the display updates in less than, say, 200 milliseconds. In the case of output-oriented rendering, and in the form it is introduced in this paper, achieving interactive frame rates does not depend on the size of the input data. The original data is converted to bins beforehand.

The conversion is done once at the beginning of the process in a fine way (256×256 bins.) This information is precise enough to make it the origin for successive binning when creating coarser context representation or when performing simple axis operations like zooming or panning. The initial binning operation is the most costly operation in the whole framework. The largest data set tested with the setup had more than 3 million data records in 16 dimensions. Ewen though the binning of this large data set took a reasonable time in tens of seconds, the binned data was rendered instantly. In contrast to that, the standard polyline-based rendering without any pre-processing took about 2.5 times more to accomplish.

This implies that even though binning is a very time-consuming process (especially when it comes to large multidimensional data), it is nevertheless a very clever investment compared to the time taken by rendering the heavily overplotted display the standard way. Once the binning is done, the rendering of the bins finishes in real-time, not talking about the clearer and more informative display.

The memory requirements are so far very reasonable. A fine binning map (256×256) takes up 256 KBytes of memory. A dataset with 50 dimensions requires approximately 300 MBytes to store the bin maps for all the possible axis-axis combinations.

Once done intelligently, the binned data can be often reused and the original data has to be worked with only if the mapping for an axis changes, if a selection (brushing) is performed and when the focus has to be rendered in full details.

8 Conclusions and Future Work

The visualization approach and implementation, as presented in this paper, show that output-sensitive methods can be used to improve the visualization of large multivariate data in many ways. By combining this approach with data abstraction, namely with binning at different levels of detail, an efficient way of modifying standard parallel coordinates is presented. This extended focus+context visualization approach treats outliers separately from the rest of the data for two reasons – to prevent the context from having misleading shape or size and to prevent outliers from getting lost inside the context.

The resulting visualization is capable of showing the context at different levels of detail while leaving enough visual resources for the outliers and for the important focus. Thanks to the density-based data representation, which is introduced by binning, the performance does not depend on the size of the input data after the binning. Thus large data sets can be explored interactively and even new interaction options are created, e.g. changing the level of detail of the context. To our best knowledge, this is the first time that datasets with several millions of multi-variate data items can be swiftly visualized in parallel coordinates (after some considerable preprocessing, however).

The density-based information in parallel coordinates brings up interesting questions with respect to screen-oriented data processing such as clustering or approximation of the original real world model. Many of them are very demanding and complex in their original data-oriented form. But the scalable and simple density-based representation makes them available even for large data sets or interactive exploration. Some of them are addressed in this paper and thanks to the promising results there are many interesting options getting open for future research.

Some of them might improve the outlier detection process, while others might focus on hierarchical clustering in the aggregated information

9 Acknowledgments

This work is part of the visualization research at the VRVis Research Center in Vienna, Austria (www.VRVis.at), which is partly funded by an Austrian research program called Kplus. The research was also supported by the Comenius University Young Scientists' Grant.

The authors would like to thank to Jürgen Platzer and Prof. Peter Filzmoser for statistical background information on outlier detection and clustering.

Many thanks also go to Harald Piringer for many good ideas and suggestions. The simulation data set shown in this paper is courtesy of AVL List GmbH in Graz, Austria.

References

[1] D. Bauer and R. Peikert. Vortex tracking in scale-space. In *Joint Eurographics — IEEE TCVG Symposium on Visualization*, pages 140–147, May 2002.

[2] Fabian Bendix, Robert Kosara, and Helwig Hauser. Parallel sets: Visual analysis of categorical data. In *INFOVIS*, page 18, 2005.

[3] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68:361–68, 1973.

[4] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In David

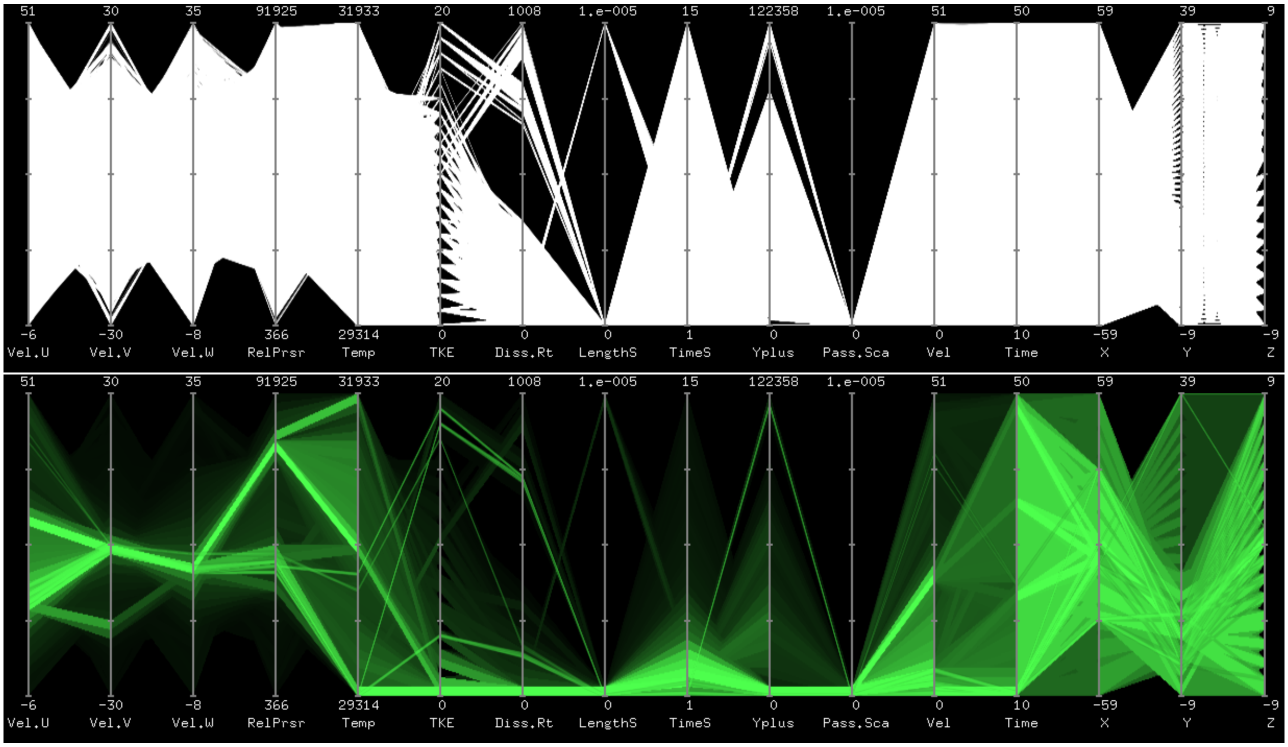

Figure 12: Output-sensitive, outlier-preserving focus+context visualization allows to even render more than three million data items into a parallel coordinates plot. Apart from the fact that the binned representation is much more clearer and informative than the standard plot with all the polylines, it also renders interactively.

Ebert, Markus Gross, and Bernd Hamann, editors, *IEEE Visualization '99*, pages 43–50, San Francisco, 1999. IEEE.

[5] A. Inselberg. Multidimensional detective. In *IEEE Symposium on Information Visualization (InfoVis '97)*, pages 100–107, Washington - Brussels - Tokyo, October 1997. IEEE.

[6] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multidimensional geometry. In *IEEE Visualization '90 Proceedings*, pages 361–378. IEEE Computer Society, October 1990.

[7] Jimmy Johansson, Patric Ljung, Mikael Jern, and Matthew Cooper. Revealing structure within clustered parallel coordinates displays. In *INFOVIS*, page 17, 2005.

[8] Daniel A. Keim and Hans-Peter Kriegel. Visualization techniques for mining large databases: A comparison. *IEEE Trans. Knowl. Data Eng.*, 8(6):923–938, 1996.

[9] Barthold Lichtenbelt, Randy Crane, and Shaz Naqvi. *Introduction to volume rendering*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

[10] R. Douglas Martin. Trellis displays for deep understanding of financial data. *The Newspaper of Financial Engineering*, 3, February 1998.

[11] John J. Miller and Edward J. Wegman. Construction of line densities for parallel coordinate plots. pages 107–123, 1991.

[12] Rousseeuw and Leroy. *Robust Regression and Outlier Detection*. Wiley Sons, 1987.

[13] Peter J. Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, 1999.

[14] B W Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, London, 1986.

[15] http://www.liacc.up.pt/ml/statlog/datasets.html.

[16] Edward R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[17] Horst Bischof Walter G. Kropatsch and Roman Englert. *Digital image analysis: selected techniques and applications*, chapter Hierarchies, pages 211–230. Springer, 2000.

[18] Manfred Weiler, Rüdiger Westermann, Chuck Hansen, Kurt Zimmermann, and Thomas Ertl. Level-of-detail volume rendering via 3d textures. In *VVS '00: Proceedings of the 2000 IEEE symposium on Volume visualization*, pages 7–13, New York, NY, USA, 2000. ACM Press.