Extensions of Parallel Coordinates for Interactive Exploration of Large Multi-Timepoint Data Sets

Jorik Blaas, Charl P. Botha, *Member, IEEE* and Frits H. Post

Abstract—Parallel coordinate plots (PCPs) are commonly used in information visualization to provide insight into multi-variate data. These plots help to spot correlations between variables. PCPs have been successfully applied to unstructured datasets up to a few millions of points. In this paper, we present techniques to enhance the usability of PCPs for the exploration of large, multi-timepoint volumetric data sets, containing tens of millions of points per timestep.
The main difficulties that arise when applying PCPs to large numbers of data points are visual clutter and slow performance, making interactive exploration infeasible. Moreover, the spatial context of the volumetric data is usually lost.
We describe techniques for preprocessing using data quantization and compression, and for fast GPU-based rendering of PCPs using joint density distributions for each pair of consecutive variables, resulting in a smooth, continuous visualization. Also, fast brushing techniques are proposed for interactive data selection in multiple linked views, including a 3D spatial volume view.
These techniques have been successfully applied to three large data sets: Hurricane Isabel (Vis'04 contest), the ionization front instability data set (Vis'08 design contest), and data from a large-eddy simulation of cumulus clouds. With these data, we show how PCPs can be extended to successfully visualize and interactively explore multi-timepoint volumetric datasets with an order of magnitude more data points.

Index Terms—Parallel coordinate plots, time-varying, multi-field, linked related views.

1 Introduction

Parallel coordinate plots (PCPs) [7] were developed as a method to create planar graphs of multi-variate data. In parallel coordinates, each N-dimensional data point is transformed into a polyline that intersects N parallel vertical or horizontal axes. Each axis represents a dimension, and the point at which the polyline intersects represents the value of the point on that dimension. Parallel coordinates have received acceptance in statistical data analysis and information visualization as a general method for visualizing arbitrary high-dimensional datasets [17]. As each high dimensional data point is represented uniquely by a polyline there is no loss of data due to projections, as is often the case with other methods such as scatter-plots. Another important advantage lies in the ability to visualize the geometry of high-dimensional objects, and not just the data.

To apply PCPs to large data sets, the limitation does not lie in the number of dimensions per data point, but in the number of points and associated polylines. Scalability of PCPs is limited by two major problems: visual clutter and reduced performance, hampering the use of PCPs for interactive exploration. A number of extensions have addressed these problems, such as the use of hierarchical and multi-resolution methods, smooth parallel coordinates, 3D PCPs, and techniques to reduce visual clutter (see section 2). However, PCPs have been mainly applied to scattered data of up to a few millions of points. One million points is just the lower boundary for volumetric data sets (100^3 voxels). To be usable for multifield volume data, extension of PCPs to the range of tens of millions of points would be necessary.

In this paper, we present techniques to allow the interactive exploration of multi-timepoint volumetric data in this range. This is achieved by using a combination of data quantization and compression, and the use of data structures to allow very fast computation and GPU-based rendering of joint density distributions, resulting in a quasi-continuous view of the line densities between each consecutive pair of parallel axes. Several facilities for interaction are available, including brushing data selection and data normalization using histogram equalization. Finally, a two-way linking is provided with spatial views of the data. This would make integration possible of PCPs in a full-blown interactive data analysis system with linked related views at the current data set sizes.

The contributions of this paper can be summarized as:

- Scalability of PCPs to the range of tens of millions of points for use with realistic multifield volumetric data sets.
- Maintaining high interactivity at this scale.
- Dynamic brushing for data selections.
- Dynamic two-way links between spatial views and PCPs.
- Support for multi-timepoint volumetric data sets.

To demonstrate the feasibility of these techniques, they have been applied to three large time-varying data sets: the contest data sets of Visualization 2004 (Hurricane Isabel) and 2008 (the ionization front instability), and an atmospheric large-eddy simulation of cumulus clouds.

The paper is structured as follows: related prior work is discussed in section 2. The processing pipeline will be described in section 3, and section 4 presents rendering and interaction methods. Section 5 gives performance data, and presents the three applications. Finally, section 6 draws conclusions and indicates future developments.

2 Related Work

In this section, we focus primarily on previous work on parallel coordinates for large, sometimes time-varying, datasets, as this is an important characteristic of our contribution. We conclude by briefly discussing recent work using linked and coordinated views for the visual analysis of large datasets, as we also demonstrate how to combine parallel coordinates on large data with other linked views.

Fua et al. defined large datasets as containing 10^6 to 10^9 data elements or more [5]. They extended XmdvTool [15] with a special form of parallel coordinates that employed hierarchical clustering. The user could interactively vary the level of detail, making possible a multi-resolutional visualization with smooth transitions to any level of the clustering or the raw data samples themselves.

In the same vein, Johansson et al. introduced the use of self-organizing maps (SOM) in order to cluster data samples and thus reduce large datasets [10]. The clusters were visualized as variable width

The authors are with the Data Visualization Group, Delft University of Technology, E-mail: {j.blaas,c.p.botha,f.h.post}@tudelft.nl

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

bands in parallel coordinates, with the width encoding information about the clusters that they represent. In this case, one could drill-down by expanding a cluster into its constituent samples in a linked view. Specifically, a large dataset is defined as one containing at least 10000 data elements with 16 dimensions.

To increase scalability without increasing visual clutter, line densities were introduced by Miller at al. [11]. Artero et al. proposed the use of Interactive Parallel Coordinate Density and Frequency Plots [2]. Bi-dimensional frequency histograms were calculated for every pair of consecutive parallel axes. For each position in a frequency histogram, a line was drawn between the relevant two axes, with its brightness proportional to the frequency. Maximum intensity compositing was used so that higher frequency samples always had precedence. In this way, large data could be aggregated for a more effective visualization. The method was tested on datasets with up to a million data elements with 50 dimensions.

By transforming each K-means-derived cluster into three high resolution textures, namely an animation, outlier and structure texture, and then compositing all cluster textures onto a polygon, Johansson et al. managed to create cluster visualizations that included information about the internal structure of clusters [9] . Cluster colors were pre-determined, but opacity was configurable by specifying a transfer function, also non-linearly, mapping from local intensity to opacity. Outliers were determined by inspecting the inter-quartile range on each dimension and visualized by making use of the outlier textures. These techniques were tested on datasets with up to a hundred thousand data elements.

Johansson et al. also investigated temporal parallel coordinates, focussing on visualizing changes over time by adding depth cues and temporal density[8].

Novotny et al. use a method similar to the approach of Artero [2], where bi-dimensional histograms, or bin maps are computed [13]. Outliers were detected directly in the bin maps and removed for separate rendering. Inspired by image processing techniques, clustering also took place directly on the bin maps. Clusters and outliers were separately rendered to retain visibility of the outliers in the final visualization. The largest dataset tested on consisted of three million data elements over sixteen dimensions.

WEAVE [6] and SimVis [4] are examples of systems that make use of linked scientific and information visualization views, including parallel coordinates, in order to explore complex datasets. More recently, SimVis was applied to large dynamic datasets [12], but without significant involvement of traditional parallel coordinates.

Ten Caat et al focus on a clinical application scenario in which temporal EEG data is explored by students, researchers and experts to assess latencies, amplitudes and symmetries [3]. Their work is a good example of how PCPs can be successfully adapted to improve the assessment of complex medical data.

The framework by Akiba et al. explores the concept of data exploration through linked views in the temporal, spatial and variable domain[1].

Existing work on parallel coordinates for large data employs a combination of clustering, binning and other feature extraction, such as outlier detection, in order to cope with large datasets. These techniques reduce both scalability and visual clutter problems.

Our work builds on the bin map idea, but adds a number of refinements in order to show how parallel coordinates can be effectively used for the interactive visualization of even larger multi-timepoint datasets with 25 million data elements per timestep over 10 dimensions. We have developed these extensions also with the idea of integrating our large data parallel coordinate pipeline with current multiple linked view systems.

3 Processing Methods

To make PCP rendering and processing fast enough for interactive exploration, an optimized data processing pipeline was adopted. Our on-disk data structure was designed to provide fast access to the data needed during interaction, and it can cope with the data access patterns that arise in multi-timepoint data.

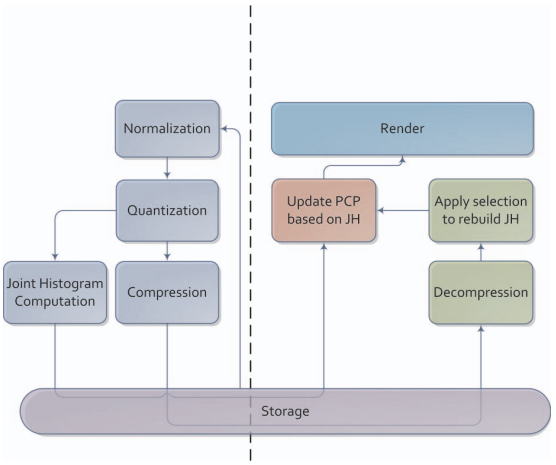


Fig. 1: The processing pipeline. The methods to the left of the dotted line are performed during preprocessing, while the other methods are continuously performed during user interaction.

We identified a number of common tasks that are necessary to fully exploit the PCP-based exploration.

- Rendering the parallel coordinate plot
- Selection of a range of points in the plot by defining an attribute range on one of the axes
- Data probing operation at a spatial location
- Spatial display of a selection
- Re-ordering the axes
- Moving between time points

Our processing pipeline, further explained in the following sections, is designed to provide a fast structure to perform the above-mentioned operations.

3.1 Processing Pipeline

The processing steps (see Figure 1) are separated into two categories: preprocessing and interaction. The preprocessing steps are only needed to be ran once, to convert the source dataset into a compact and easily-accessible storage format. Once preprocessing has been performed, the stored data is loaded on-the-fly during the exploration process.

The following sections further explain the details of the methods shown in the pipeline diagram.

3.2 Histogram Equalization

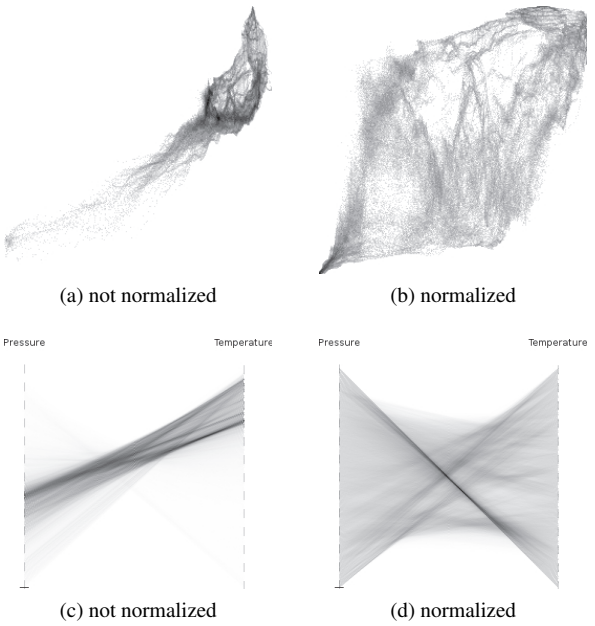
One of the problems with the large datasets is that the distributions of the attribute values are often not very smooth. When such a distribution is highly skewed, a large fraction of the values will be in a small range of numbers. When such a dataset is linearly rescaled to a range $[\min, \max]$, only a small part of the vertical axis is used in the resulting plot. This leads to a high level of clutter, and makes it difficult to spot the internal correlations that occur inside the dense area.

This is why we optionally perform a non-linear univariate normalization. For each attribute, a histogram is created of the data-values for a single attribute over all voxels and over all timesteps. Once the histogram of each attribute is known, its values can be mapped through histogram equalization to values in the range $[0, 1]$ in such a way that the values will have a continuous density in the target domain.

The effect of normalization of both the pressure and the temperature in a single slice of the hurricane Isabel dataset is shown in Figure 2.

Note that the intricate internal patterns that occur in the high density area of the unnormalized data are well visible after normalization.

It is crucial that the histogram equalization is not performed on each timestep individually. If each timestep was normalized individually, then each timestep would have a different normalization mapping, and the correspondence between attributes over time would be lost, making it impossible to compare values between timesteps or to spot changes that occur over time.



(a) not normalized (b) normalized

(c) not normalized (d) normalized

Fig. 2: The effects of normalizing the data by histogram equalization. A two-dimensional dataset consisting of pressure and temperature data is shown before and after normalization. (a) shows a scatterplot of the data without normalization, (b) shows the scatterplot after the joint histogram has been normalized. (c) and (d) show the parallel coordinate plot corresponding to the data in figure (a) and (b), respectively.

3.3 Quantization

Parallel coordinate plots have little to gain from a high precision floating point representation of the data. When data values are rounded to a lower precision representation, the maximum erroneous displacement that a line in the plot will have is directly related to the rounding error. Since the axes of a parallel axis plot are generally not higher than 512 pixels, a quantization to 8-bit values will yield a maximum displacement of a single pixel, which is adequate for our purpose.

This quantization step reduces the data from 4 bytes to a single byte per point per attribute, greatly reducing the necessary storage. As we will see later, storing the data in a fixed-point format also improves the compressibility.

3.4 Joint-histogram Generation

A parallel coordinate plot without any selections can be quickly generated solely from the joint-histograms of the data. We make use of the binning approach proposed by Artero et al. [2]. Using this technique, only the joint histogram between each pair of neighboring axes is needed to build the parallel coordinate plot.

Fast exploration of the data over time is made possible by precomputing joint-histograms of all pairs of axes. For N axes, this costs $N*(N-1)/2*bins^2*\text{sizeof}(uint32)$ space. Whenever a new timestep is selected, $N-1$ reads suffice to quickly produce a new parallel coordinate plot based on the joint histograms between each pair of axes.

3.5 Storage and Compression

As seen in Figure 1, two types of stored data are used during interaction: pre-computed joint-histograms and raw compressed data volumes.

For each timestep, the joint-histograms are stored as raw blocks of $bins^2$ unsigned integers. Since the axis order determines which of the joint histograms are needed, we store all $N*(N-1)/2$ joint histograms in separate files. This makes it easy to load the $N-1$ needed files for any axis order at runtime.

The raw data volumes each represents a single scalar defined over the full volume. Since the data values have been quantized, only one byte per voxel is needed. Since these datasets are fairly large, the disk access still forms a major bottleneck when changing to another timestep. To partly alleviate this bottleneck, a compression step is performed.

We have selected to use the LZO (Lempel-Ziv-Oberhumer) compression, of which a public implementation is available [14]. LZO compression is well suited for realtime decompression, since it has a very high decompression speed while still maintaining a good compression ratio. Depending on the compressed size, the rate at which decompressed data was produced ranged from 110 to 250 MB/sec, measured using a single core of a 2.0 GHz AMD Athlon64 X2 3800+ processor.

To store the compressed data on disk, one file per variable/timestep combination is used. This makes it easy to load the complete volume for a specific timestep, while still keeping the possibility of using a reduced set of variables to speed up processing when necessary.

The volume of the currently loaded timestep is loaded fully in memory. The in-memory structure of the volume is such that the fastest changing axis corresponds to the attribute number, followed by the three spatial axes. This ordering enhances the spatial locality during the histogram creation phase, resulting in faster selection processing.

The following table shows the order in which the variables are stored in memory:

T_0	P_0	LW_0	T_1	P_1	LW_1	...

T_x, P_x and LW_x represent the temperature, pressure and wind-speed at location x. Each cell represents a single memory location, ordered in a left to right fashion.

4 Interaction Methods

We implemented a demonstrator to show which type of interaction is possible with the data sets of the intended size.

4.1 Rendering

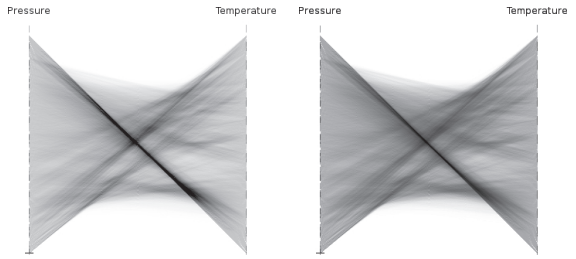
We adopt the rendering approach of Muigg et al. [12], where the histogram bins form a direct basis for drawing the primitives. Instead of having to draw a line for each data point, only a single primitive is drawn for each histogram bin.

To combine all drawn primitives together, additive blending is used. Since the intensity over the plot varies widely, a high precision floating point framebuffer is used as a render target, so that no clipping of color values is necessary. The added advantage of this technique is that intensities can be converted to color values in a post processing step.

We use a logarithmic intensity scale (see Figure 3), as to prevent over-saturation of high-density areas in the plot, while keeping a good visual contrast in low intensity areas. The contrast can be modified interactively by the user to emphasize the high or low intensity areas of interest.

4.2 Selections

Two commonly used types of selections are implemented in our demonstrator application. Basic selections are made by dragging the mouse over a range of values at any given axis. In addition, compound selections can be made by combining basic selections in an AND or OR-like fashion, so that more complex phenomena can be easily studied.



(a) Linear (b) Logarithmic

Fig. 3: The effect of using a logarithmic intensity scale. Where the high density center area on the left is completely saturated, the logarithmic intensity map on the right shows no signs of clipping and still has a high contrast in all areas.

Changes in selection criteria are processed by performing a linear scan over the data, splitting it in two sets of points: selected points and unselected points. The two sets of points are treated independently. Each of them is rendered as a parallel coordinate plot, which, after coloring, can be blended to form a clear visual representation of the selected points.

4.3 Spatial Exploration through Linked Views

Multiple linked views present a good way of linking the different representations of data. In this case we have chosen for a two-way linkage between the top and the bottom half of the screen (see Figure 4).

Whenever a point is selected with the mouse in the slice viewer, a probe determines the data values for the selected voxel and overlays these on the parallel axis plot.

Selections made in the lower part of the screen are similarly linked to the slice viewers. The slice viewers continuously display which pixels are included in the selection. The orange and blue color scheme corresponds to the parallel coordinate plot such that selected pixels are orange and unselected pixels are blue (see Figure 5).

To provide further insight on the selected values, the current selection can also double as a colormap definition. Each selection that the user makes will use the data values of that selected variable to produce colors. The colors are determined based on the relative position of the sample inside the selected range of values, so that low values correspond to dark colors and high values to bright colors. In this way, each range selection made in the parallel coordinate plot defines a single-color mapping (see Figure 6).

Fig. 6: Defining a colormap using two selections. Two ranges are selected in the PCP on the left, which correspond to two separate color maps in the slice view on the right. Bright blue pixels correspond to a high gas temperature, and bright green pixels to a high H_2^+ mass abundance.

4.4 Temporal Exploration

The PCP at a specific timestep visualizes the distribution of the datapoints, but in multi-timepoint data the changes in the distribution over time also posses key information. The demonstrator application provides a time slider through which the user can navigate through all available timesteps. During the movement of the slider, the PCP is rapidly updated using the joint histograms (Section 3.4). When a timestep is selected, the full dataset for that timestep is loaded and decompressed into memory. The typical loading time is 0.5 to 2 seconds, which is acceptable for this style of interaction.

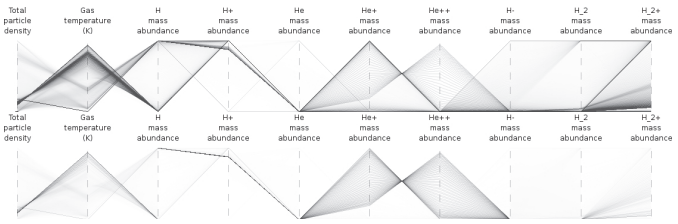


Fig. 7: Comparison against baseline. The top figure shows the PCP for timepoint 126. The bottom figure focuses on the changes over time by showing the difference between timepoint 126 and a stored baseline at timepoint 125.

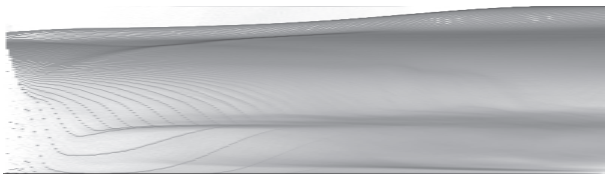


Fig. 8: Visualization of the change in distribution of the gas temperature over time. The horizontal axis corresponds to time, the vertical axis to gas temperature, while the intensity values in each pixel represent the number of data points within the corresponding temperature range. This makes each column of the plot a histogram of the temperature values for that timestep.

4.4.1 Comparison against Baseline

Since changes in volumetric simulation data between adjacent timesteps are often limited to a small number of points, the changes in the total distribution of the values over all points are quite small. To be able to focus on these smaller changes, a baseline distribution on one timepoint can be stored. The PCP can then be used to visualize the difference between the distribution at another selected timepoint and the stored distribution (see Figure 7).

4.4.2 Histograms over Time

In the PCP, at each labeled axis, the intensity of the pixels of that column corresponds to the histogram of a distribution of the corresponding variable. The changes in these distributions over time are often indicative for the specific chemical reactions or other events. Inspecting these changes often provides key insights into what is happening in the studied phenomena.

To monitor the changes in the distribution of the values, the PCP can be inspected while moving the timestep slider. In some cases however, this method can be tedious and it can be difficult to pin-point a specific point in time at which the distribution starts changing.

Therefore we propose a second method to inspect these changes. For a single selected axis in the PCP, a temporal view can be opened that displays a plot of the distribution on that specific axis over all timesteps. For each timestep, the intensity profile along the column of the selected axis in the plot is extracted. The extracted profiles are joined together in a single plot (see Figure 8).

4.5 Axis Order

The ordering of the axes is a vital part of any good PCP. The demonstrator application starts with a pre-computed axis ordering, but the user can interactively drag and drop the axes to reorder them if necessary.

An axis can be swapped with another axis, or it can be moved to a position between a pair of adjacent axes. As only the joint-histograms

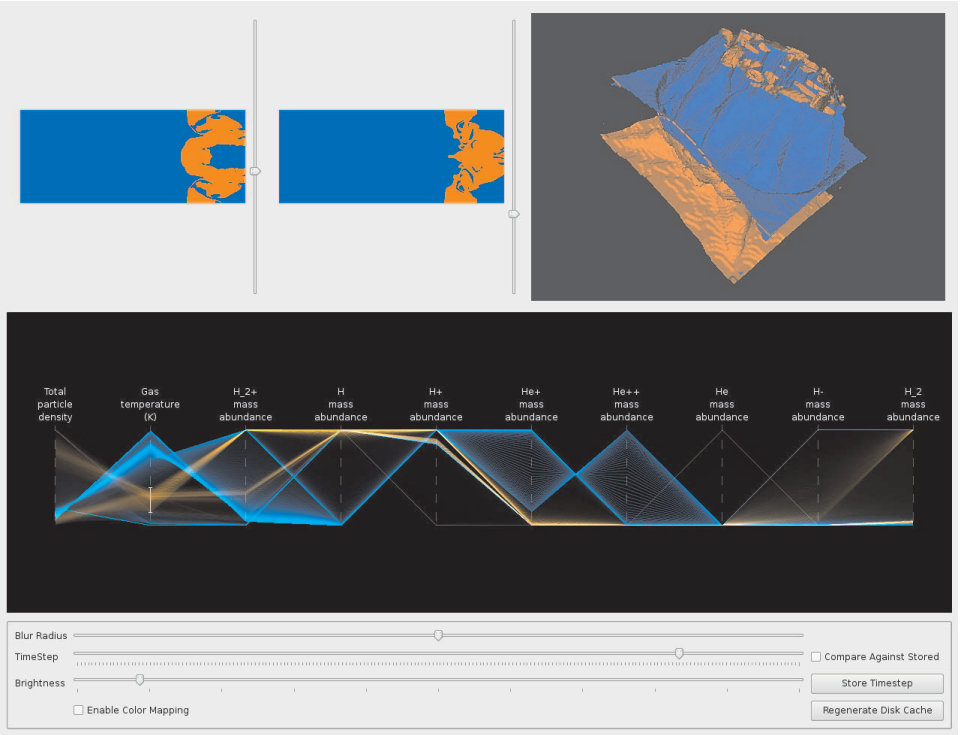


Fig. 4: The main user interface of the demonstrator application. The top part of the screen contains spatial viewing components (two slice viewers and a 3D isosurface view). The parallel coordinate plot is positioned in the middle, and the control interface is positioned at the bottom.

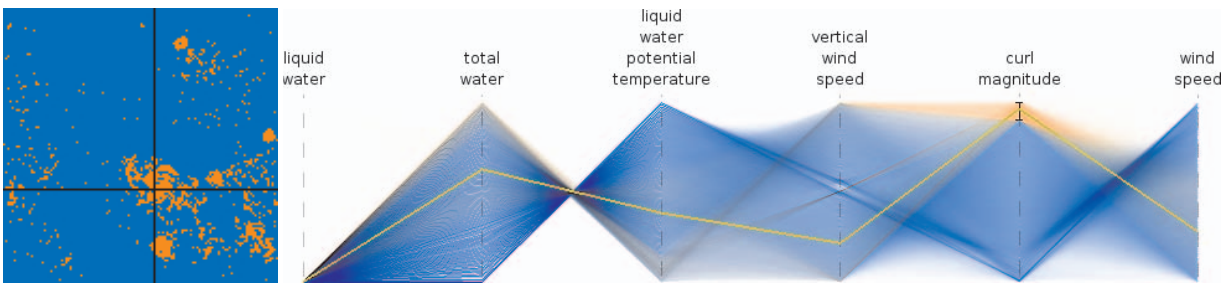


Fig. 5: A slice viewer in which a point is selected (left) linked to a parallel coordinate plot (right) which displays the selected value as a bright yellow line.

related to the current axis order are contained in memory, new joint-histograms have to be loaded from disk after such a manipulation. In the worst case, four new joint-histograms have to be loaded.

5 Results / Applications

To study how well the proposed techniques work on a real-life dataset, we have selected three large volumetric datasets to perform visual exploration on. We have selected the visualization contest datasets of 2004 and 2008, since they are publicly available and both good examples of multi-scalar temporal datasets. The third dataset we selected is an atmospheric simulation with the goal of studying cumulus clouds. Table 1 shows the characteristics of the explored datasets. The following sections describe the data and accompanying exploration.

All performance measurements were made on a desktop PC consisting of a 2GHz AMD Athlon64 X2 CPU, 2 gigabytes of RAM and an Nvidia GeForce 7950GT graphics card. Since no multiprocessing was implemented in the demonstrator application, only a single CPU core was used during the benchmarks.

5.1 Hurricane Isabel Dataset

The hurricane Isabel dataset is part of the visualization contest of 2004. It contains a detailed simulation of a hurricane moving over the west Atlantic region.

The hurricane Isabel dataset has the highest spatial resolution of all three datasets we explored. Each timepoint consists of 25 million points with 10 attributes each, resulting in 250 million data values.

5.1.1 Performance

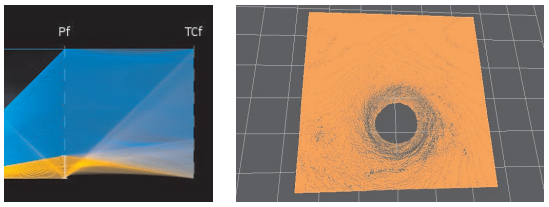
During navigation with the time slider, the loading time for the precomputed joint-histograms is consistently around 0.02 seconds per timestep. Combined with the 0.1 seconds it takes to draw the complete parallel coordinate plot, this results in 8-9 frames per second when moving the time slider. Once a new timestep has been selected, it takes 2.5 seconds to load and decompress the full data volume from disk. The recalculation of the histograms takes another 2.4 seconds. While this recomputation takes a considerable amount of time, it is important to note that this step is only necessary when the selection changes. Once the histograms have been recomputed, the slice view, probe and 3d-rendering run at over 30 frames per second.

Dataset	Spatial Resolution	Points	Timesteps	Attributes	Original size	Quantized size	Compressed size
Hurricane Isabel	$500 \times 500 \times 100$	25.000.000	48	10	44.70 GB	11.18 GB	2.41 GB
Vis 2008 contest (halved)	$300 \times 124 \times 124$	4.612.800	200	10	34.37 GB	8.59 GB	0.78 GB
Cumulus cloud dataset	$128 \times 128 \times 80$	1.310.720	600	6	17.58 GB	4.39 GB	3.00 GB

Table 1: Overview of the used datasets and their size.

5.1.2 Eye of the Hurricane

To get a good overview of the dataset, we select the low pressure area near the eye of the hurricane by applying a range selection on the temperature axis. The 3D view shows the spatial shape of selected area, while the PCP provides us with the information that low pressure areas are in this case highly correlated with low temperature areas (see Figure 9). The hurricane's movement over time is directly visible when moving the time slider to a different timepoint.

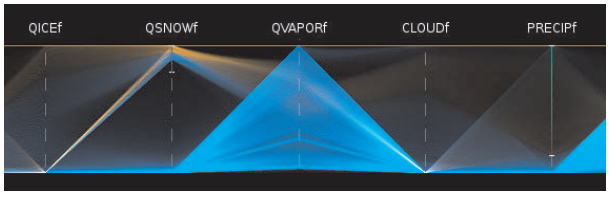


(a) PCP selection (b) 3D view

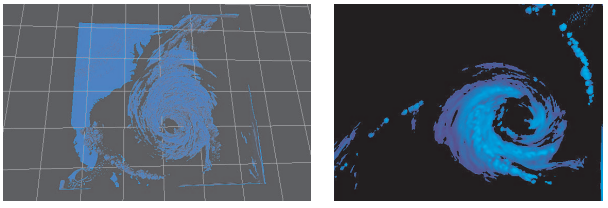
Fig. 9: Selection of low pressure areas (shown in orange) reveals the area of low-temperature near the eye of the hurricane. The compact horizontal shape of the orange band in the PCP reveals that low-pressure areas mostly have a low-temperature as well.

5.1.3 Snow and Precipitation

In normal conditions, precipitation leads to low humidity. We explored how the amount of snow is related to the precipitation (see Figure 10).



(a) PCP selection



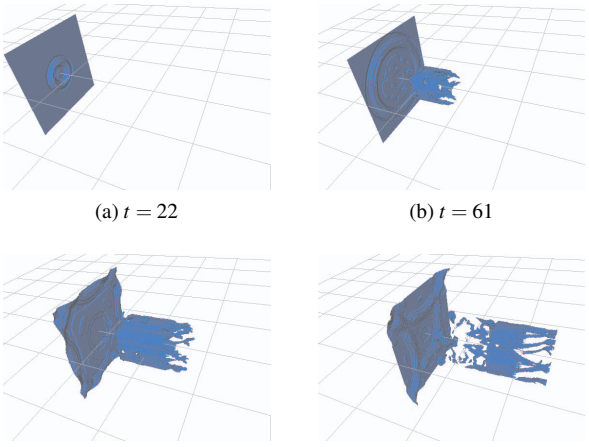
(b) 3D view (c) Color-mapped slice view

Fig. 10: Exploration of the hurricane Isabel dataset. The combination of high precipitation and snow has been selected so that the blue colors in the slice view correspond with snow while green corresponds to areas with high precipitation.

We found no real changes in the distribution over time. While the eye of the hurricane does move spatially, the overall composition does not change significantly, as the distribution of the values within the hurricane is rather constant.

5.2 Vis2008 Contest Dataset

The 2008 visualization contest dataset is a simulation of an ionization front instability [16]. There are ten attributes that describe the total particle density, the gas temperature, and the abundance of H, H^+, He, He^+, He^++, H^-, H_2 and H_2^+.



(a) $t = 22$ (b) $t = 61$

(c) $t = 91$ (d) $t = 118$

Fig. 11: Isosurface view of the movement of a high temperature front in the 2008 contest dataset.

We have down-sampled the data spatially to half the resolution, resulting in a volume size of 4.6 million points. The loading and decompression time of a single timestep was 0.6 seconds, while histogram updates took 0.4 seconds.

One of the most interesting features of this dataset is the fact that the distributions of the attributes change considerably over time. The simulation starts out as an ionization front hits a small spherical bump, causing the front to break in a turbulent matter. This fact can be seen in the PCP, as in the early moments of the simulation it shows only a single line, which quickly spreads out into a wide set of bands.

We were able to track the ionization front by selecting all high temperature points, and monitoring their three dimensional spatial structure over time (Figure 11). This clearly shows how the initially stable front breaks up in a highly turbulent structure.

We noticed that the presence of H and H^+ shows an interesting correlation (Figure 12), which we can pinpoint spatially through the use of color mapping on a slice view. After the selection of the attribute ranges, the color map feature was enabled so that high H mass abundance and H^+ mass abundance are represented by bright blue and bright green colors respectively.

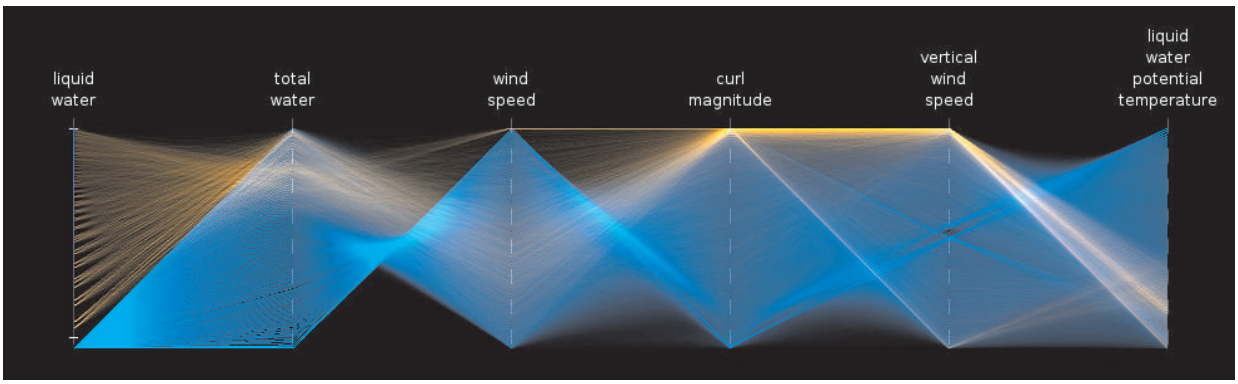
The distribution changes can be partly explained by the fact that the area of effect moves outside the bounds of the simulation.

5.3 Cumulus clouds

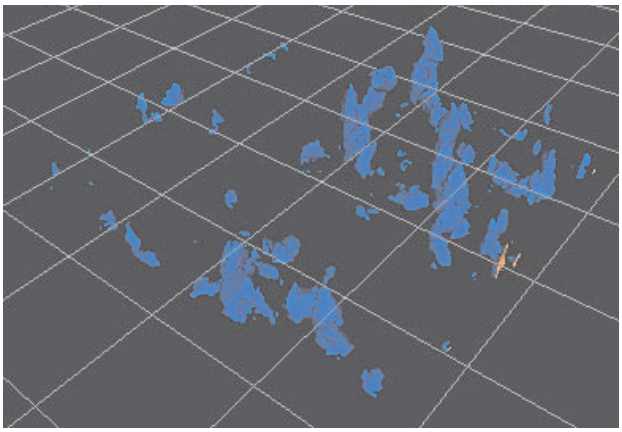
The cumulus clouds dataset is the result of a Large-Eddy simulation with the aim of studying cloud life-cycle patterns. The dataset contains 4 attributes, representing the amount of liquid water, the potential temperature, a wind vector and the amount of total water. We have added derived features in a preprocessing step, resulting in two additional features; the wind speed and the vorticity.

Since the spatial resolution of the dataset is quite low, the data does not compress as well as the other datasets. However, as the volume size is relatively small, the load times are still under 0.1 seconds per timestep. A full histogram update takes 0.07 seconds.

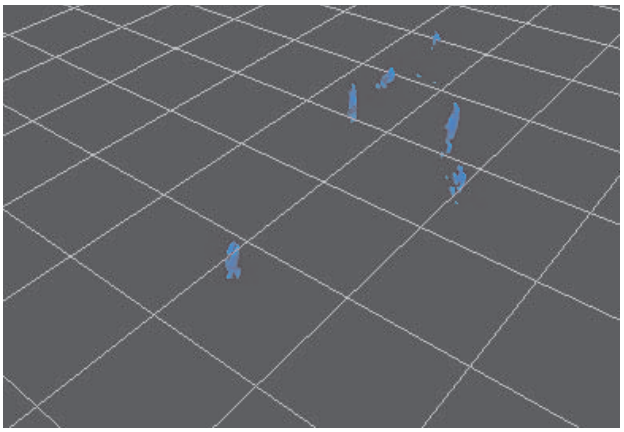
Firstly, we verified that the simulation is in a steady state, by examining the histogram over time of the main attributes. The results



(a) PCP selection of clouds

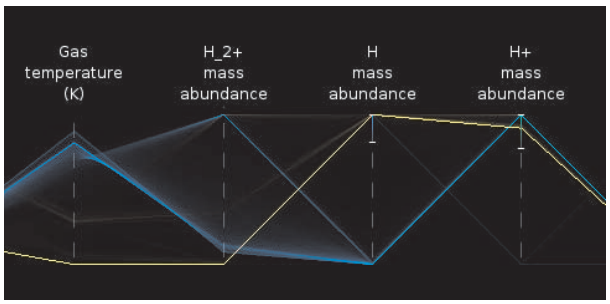


(b) All clouds



(c) High-windspeed clouds

Fig. 13: All clouds can be selected based on the liquid water attribute (a,b). The orange lines show the distribution of the other attributes within the clouds. When the high wind-speed outlier is selected, only a select subset of the clouds is visible (c).



(a) PCP

(b) Slice

Fig. 12: A PCP showing the correlation between H and H^+ mass abundance. The PCP uses two color-mapped selection ranges so that the colors in the slice viewer represent the H mass abundance (blue) and the H^+ mass abundance (green).

indicate that the distribution is almost uniform, and no major changes over time are visible. This corresponds to the expected distribution of a steady state.

A large part of the volume does not contain visible clouds. To visualize the clouds, we performed a selection based on the amount of liquid water present in a voxel (see Figure 13). When moving the time-slider, the three dimensional isosurface representation of the clouds corresponded to our expectations. The formation of new clouds is clearly visible.

The PCP shows that our selection produces one interesting outlier when looking at the wind speed. A secondary selection can be used to find where these outliers are located spatially. These high wind-speed parts are characteristic to a specific phase of the cloud life-cycle, and their formation is subject of current research.

5.4 Insights gained

Our main goal with these three examples was to show that the techniques presented in this paper enable the interactive exploration of large time-varying datasets with parallel coordinate plots. With dataset sizes ranging from 1.3 million points with 6 attributes over 600 timesteps to 25 million points with 10 attributes over 48 timesteps, and our system enabling navigation through timesteps at 8 to 9 frames per second, loading of a full timestep to updated linked views and selections at between 0.1 and 5 seconds and finally interaction with linked volume and slice views at 30 frames per second, we think that we have successfully reached our main goal.

With regard to PCP as a suitable visualization method, the fact that all points are explicitly linked over all dimensions is a clear advantage over many other multi-dimensional visualization techniques. For example, clusters over a subset of the dimensions are explicitly

linked and visible in all other dimensions, so that one can study divergence over the non-clustered dimensions. With scatter plots, one has to make use of multiple views with linked brushing to accomplish this, in which case one can only study selected versus non-selected points, as there is no other easy way to link points in different linked views. In parallel coordinates, the explicit linking is independent of the selection and valid for all points.

In all three examples, selection was used to investigate the data. However, due to the explicit linking explained above, relations over all dimensions are visible even before selections are made. For example, in the case of Hurricane Isabel's eye, the correlation between low pressure and low temperature was already visible. Based on this, the selection could be made to further study the relation and to show the eye in the linked views. Generally speaking, the point to polyline mapping characteristic of PCPs facilitates the selection of interesting patterns over all dimensions.

6 Conclusions and Future Work

We have demonstrated the use of our interactive tool for exploring large volumetric data sets using PCPs, linked views, and interactive brushing selection with three large time-varying data sets. The results in terms of clarity of visualization and interactive response times were quite encouraging.

We do not want to suggest that PCPs are the only or even the best way to analyze large high-dimensional data sets. Our aim is to show that PCPs can be fruitfully used with full-sized time varying volumetric data sets, and to make it possible to integrate PCPs in an interactive multiple-linked-views type of environment.

As noted in section 3.4, the number of joint-histograms that has to be pre-computed scales quadratically with the number of dimension of the dataset. This makes application of the proposed technique difficult when the number of dimensions exceeds about 20, as both preprocessing time and used disk-space grow quadratically. However, even without the pre-computed histograms the data can be explored, as interactive brushing does not depend on the joint-histogram data. The temporal navigation though, will be slowed down considerably, as each timestep has to be loaded from disk. We intend to partly alleviate this problem by calculating the joint-histograms in an on-demand fashion, so that only the histograms related to the current axis ordering are computed. Also, we intend to investigate the usage of automatic axis-ordering algorithms to further aid the user in exploring higher dimensional data.

The proposed normalization method worked well in our cases, but it focuses mostly on displaying the relative distributions, and does not allow for quantitative display of data values. To alleviate this, we intend to provide each visible parallel axis with a set of tickmarks that are equally spaced in the original data domain, so that an intuitive mapping can be made between the normalized and the original data values. Non-linear tickmark placement however is not a trivial task, and the ability to mentally transform them to data ranges will probably vary between viewers.

Although the techniques used were designed for efficiency, the current implementation can definitely be further optimized for speed. We intend to do this by avoiding operations on the full data as much as possible, and by applying a streaming data-on-demand strategy. This will allow us to use only a subset of data for previewing, and load the full data in the background. We also want to explore clustering techniques for reducing visual clutter. An interesting addition is to enable partial histogram equalization, so that the user can smoothly change from the original to the normalized data. Finally, we intend to make the PCP tool freely available to the research community.

Acknowledgements

The processing software makes use of the `teem` toolkit, available at `http://teem.sourceforge.net`.

The authors will like to thank Bill Kuo, Wei Wang, Cindy Bruyere, Tim Scheitlin, and Don Middleton of the U.S. National Center for Atmospheric Research (NCAR) and the U.S. National Science Foundation (NSF) for providing the Weather Research and Forecasting (WRF) Model simulation data of Hurricane Isabel.

We also would like to thank Thijs Heus and Harm Jonker of Delft University of Technology for providing the Large Eddy Simulation data sets.

References

[1] H. Akiba and K.-L. Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In K. Museth, T. Möller, and A. Ynnerman, editors, *EuroVis*, pages 115–122. Eurographics Association, 2007.

[2] A. Artero, M. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*, pages 81–88, 2004.

[3] M. Caat, N. Maurits, and J. Roerdink. Design and evaluation of tiled parallel coordinate visualization of multichannel eeg data. *Visualization and Computer Graphics, IEEE Transactions on*, 13(1):70–79, Jan.-Feb. 2007.

[4] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 239–248. Eurographics Association, 2003.

[5] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proc. IEEE Visualization*, pages 43–50, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[6] D. Gresh, B. Rogowitz, R. Winslow, D. Scollan, and C. Yung. Weave: a system for visually linking 3-d and statistical visualizations applied to cardiac simulation and measurement data. In *Proceedings of IEEE Visualization 2000*, pages 489–492, 597, 2000.

[7] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(4):69–91, 1985.

[8] J. Johansson, P. Ljung, and M. Cooper. Depth cues and density in temporal parallel coordinates. In K. Museth, T. Möller, and A. Ynnerman, editors, *EuroVis*, pages 35–42. Eurographics Association, 2007.

[9] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *INFOVIS '05: Proceedings of the IEEE Symposium on Information Visualization*, pages 125–132, 2005.

[10] J. Johansson, R. Treloar, and M. Jern. Integration of unsupervised clustering, interaction and parallel coordinates for the exploration of large multivariate data. *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*, pages 52–57, 2004.

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

[12] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser. A four-level focus+context approach to interactive visual analysis of temporal features in large scientific data. In *Proc. Eurographics / IEEE-VGTC Eurovis*, 2008. Accepted, to appear.

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

[14] M. Oberhumer. Lzo real-time data compression library, 2005. http://www.oberhumer.com/opensource/lzo/.

[15] M. O. Ward. Xmdvtool: integrating multiple methods for visualizing multivariate data. In *VIS '94: Proceedings of the conference on Visualization '94*, pages 326–333. IEEE Computer Society Press, 1994.

[16] D. Whalen and M. Norman. Competition data set and description. In *2008 IEEE Visualization Design Contest*, 2008. http://vis.computer.org/VisWeek2008/vis/contests.html.

[17] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques*, pages 3–33, Washington, DC, USA, 1997. IEEE Computer Society.