Progressive Parallel Coordinates

René Rosenbaum *
Institute for Data Analysis and Visualization (IDAV), Department of Computer Science, University of California, Davis, CA 95616-8562, U.S.A.

Visual Computing and Computer Graphics group (VCG), Institute for Computer Science, University of Rostock, 18059 Rostock, Germany

Jian Zhi †
Department of Industrial Engineering and Operations Research, Columbia University, NY 10027, U.S.A.

Bernd Hamann‡
Institute for Data Analysis and Visualization (IDAV), Department of Computer Science, University of California, Davis, CA 95616-8562, U.S.A.

ABSTRACT

Progressive refinement is a methodology that makes it possible to elegantly integrate scalable data compression, access, and presentation into one approach. Specifically, this paper concerns the effective use of progressive parallel coordinates (PPCs), utilized routinely for high-dimensional data visualization. It discusses how the power of the typical stages of progressive data visualization can also be utilized fully for PPCs. Further, different implementations of the underlying methods and potential application domains are described. The paper also presents empirical results concerning the benefits of PPC with regard to efficient data management and improved presentation, indicating that the proposed approach is able to close the gap between data handling and visualization.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Theory and Methods

1 INTRODUCTION

Data visualization usually involves large data sets to be handled leading to two effects that hinder appropriate data analysis: (1) long processing and transmission times and (2) clutter in the visual data representation. In the past, solutions to solve both issues have been proposed mainly by the use of established data management solutions, like online analytical processing (OLAP), and scalable visualization techniques. The loose coupling of these two components, however, causes problems in interactive data exploration. This especially applies for the highly redundant access to the different level-of-detail (LOD) of the data causing existing management solutions to handled already processed data multiple times. The use of redundancies, however, is absolutely essential in visual data exploration usually performed by applying small incremental changes to current data views only. This dilemma clearly indicates a gap between data management and visualization.

Progressive data displays [15] adopting the principle of *progressive refinement* (progression) from the domains of image communication and volume rendering [11] unify all visualization stages into a single strategy. Proper data management is achieved by scalable data compression and random access allowing to handle and transmit the different LODs of the data with high resource efficiency. A novel kind of scalable data presentation is achieved by incremental data reconstruction. The resulting data previews support the analyst in gaining insight into the data early and successively. Solutions based on this concept have already been proposed for the visualization of hierarchical [15] and abstract data [14]. The multiple problems imposed by more complex data and their representation, however, have not been discussed so far.

*e-mail: rosenbaum@ieee.org
†e-mail: simon-hans-zhi@msn.com
‡e-mail: hamann@cs.ucdavis.edu

IEEE Pacific Visualization Symposium 2012
28 February - 2 March, Songdo, Korea
978-1-4673-0866-3/12/$31.00 ©2012 IEEE

This paper discusses how progression can be made available for the *parallel coordinates* (PC) plot. PC is probably the most widely referenced technique to display multi-dimensional data [9]. It is founded on a projection of the data into two-dimensional space that leads to dimension axes that are displayed as parallel line segments. An individual data point is represented by a polyline intersecting the axes dependent on its respective dimension-specific values. One of the disadvantages of PC is the heavy over-plotting even for a small number of data points (see Figure 1, d). Our goal is to introduce novel ways for *data management* and *presentation* for the PC display. The key contributions of this paper to the current state of research can be stated as:

1. Discussion of means for combining scalable data compression and representation for PPCs.
2. Discussion of means for demand-driven data ordering and representation for PPCs.
3. Empirical evaluation of technical and semantic benefits of progression for PPCs.

Section 2 is concerned with the analysis of progressive refinement and its relation to research that was done for PC. All associated aspects, as scalable compression and visual representation as well as options for a flexible demand-driven data ordering, are discussed in Section 3. Solutions for the associated problems either by novel approaches or the application of existing research are introduced leading to *progressive parallel coordinates* (PPC). Potential application domains of broad impact are introduced in Section 4. In order to show the achievement of our goals, Section 5 provides empirical results regarding data management and presentation. In Section 6 we conclude that PPC tightly combine both aspects leading to a much lower resource consumption.

2 RELATED RESEARCH CONCERNING PROGRESSIVE REFINEMENT

Prior analysis [2, 15] has revealed that all progressive refinement schemes first transfer the source data into an appropriate LOD hierarchy before they are incrementally transferred and displayed. As hierarchy-building is the most complex process in the whole refinement process, this hierarchy is created and stored only once and used multiple times for different demands. Thus, it must be highly flexible and basically satisfy the following two requirements: (1) *scalability* and (2) *compression and random access* to the encoded data. Its *visual representation* during refinement must be carefully chosen to avoid misinterpretations of the provided previews. The need to serve different requests requires mechanisms for flexible data *ordering*. Each of these requirements is now considered in more detail with regard to data visualization and the PC plot.

Scalability Introducing scalability in the data is a current research topic of broad interest. Many approaches and strategies for either data [4], presentation [8], or image space [13] have already

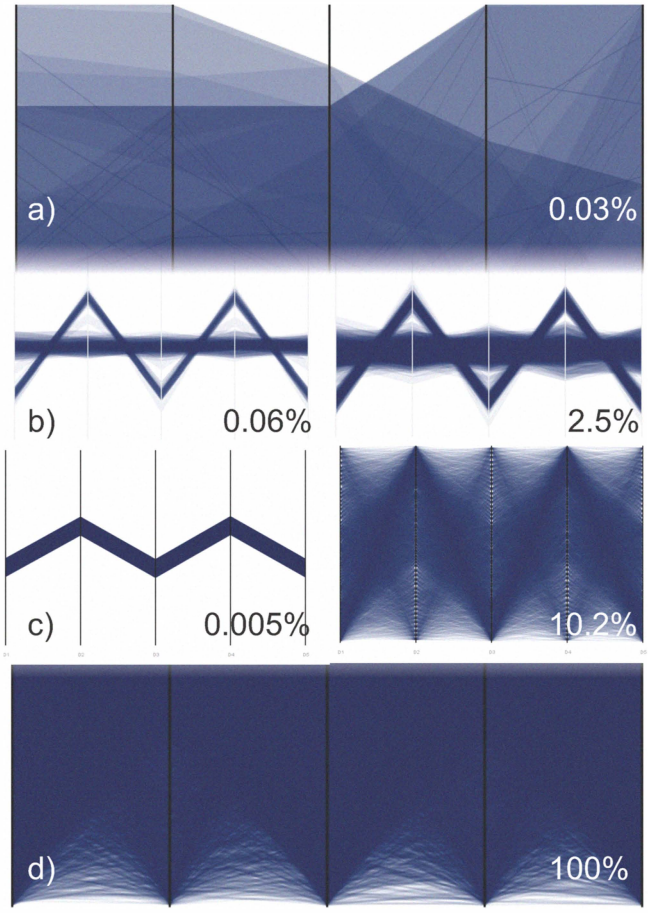

Figure 1: PPC can provide additional insight into the data, such as outliers (a) or trends (b,c). This information is usually hidden by clutter (d), but can be made visible by using an appropriate strategy for hierarchy-building like clustering (a), wavelet transform (b), or recursive interval subdivision (c). Due to the strongly decreased number of displayed primitives (in %), early refinement stages are especially well-suited to convey data properties covered within the fully detailed display. This also reduces resource consumption.

been proposed. Probably the most significant methods in data space are general (e.g., wavelet transform) or importance-driven transformations (e.g., principal component analysis), subsetting (e.g., random sampling), segmentation (e.g., cluster analysis), and aggregation. With regard to progression all these approaches can be applied as long as the process leads to a meaningful LOD hierarchy. Leaf nodes of the hierarchy thereby are required to represent the original data items and the inner nodes the aggregate items of all the items associated with the respective subtree [5].

Most of the scalability approaches proposed in the literature are designed for the geometry used by a specific visualization technique. YANG ET AL. [19] presented a framework for interactive hierarchical displays and introduced many examples including multi-dimensional data. More recently, the articles of ELLIS AND DIX [4] and ELMQUIST AND FEKETE [5] covered more general aspects and strategies for introducing scalability in visualizations.

Scalable presentation methods have also been proposed for PC. Prominent examples are the methods by FUA ET AL. [6], YANG ET AL. [19], and more recently ZHOU ET AL. [20]. They are mostly based on hierarchical clustering and allow for a meaningful scalable representation. However, they do not touch on the data management aspects limiting the potential application domains to high-performance systems. A strategy based on scalability in image-space was proposed by NOVOTNY AND HAUSER [13]. It is fundamentally different from our approach that is applied in data-space.

Compression and random access In order to reduce resource requirements, appropriate means to compress the LOD hierarchy and to access its individual values in compression space must be found. *Hierarchical compression approaches* remove redundancies that exist between the different levels and usually transform absolute to relative values allowing for quick *random access* to single LODs. In visualization, most associated compression methods have been developed for data with spatial references, such as volume data [11] or geometry [8], and cannot be applied broadly. A generic approach for geometry compression has been proposed by Deering [3]. It is referred to as $\triangle$-coding and can be meaningfully applied where ranges of values continuously decrease. Not much is known about a generic scalable compression of abstract or multi-dimensional data. This is probably due to the fact that these data sources cover a wide range of properties that must be taken into account for meaningful compression. We adopt the principle of $\triangle$-coding to propose a novel compression and random access strategy for multi-dimensional numerical data.

Visual representation The visual appearance of aggregated values is important to interpret a data preview and thus is strongly technique-dependent. Existing hierarchical representations for PC mainly use polygons to show the distribution of the data [6, 19, 13, 20]. These polygons are usually colored to allow for visual distinction and made transparent to reduce the effect of over-plotting. We adopt polygons as one option and propose a novel means to display aggregated items.

Ordering The LOD hierarchy and means for random access provide options to represent the compressed data in many different ways. This, however, is highly dependent on the respective data type. Different concepts for trees (node-of-Interest, [15]) or geometry (geometry-of-Interest, [15]) have been introduced. Common to all is that items-of-interest are handled and presented first. We adopt and enhance this concept for the demand-driven presentation of multi-dimensional data.

To summarize, technology related to progression has already been proposed and can be used to solve selected aspects of its implementation. An appropriate application of progression to multi-dimensional data and PCs is still missing. This paper closes this gap by proposing novel solutions and taking advantage of established research.

3 Progressive Parallel Coordinates

In order to make progression available for PC, meaningful solutions for all of its requirements must be provided. We present three different approaches to achieve *scalability* of the source data. For *compression and random access* we introduce a strategy that is based on $\triangle$-coding. We also discuss two strategies for an appropriate *visual representation*. Based on the created and compressed data hierarchy, novel and widely applicable concepts for data *ordering* are introduced later. All individual solutions are independent modules within the progressive processing pipeline [15]. To accommodate other needs, they can be easily substituted by different implementations.

3.1 Scalability

In this paper, we focus on multi-dimensional numerical data. *Recursive interval subdivision* is a novel approach for hierarchy-building and introduced first. From the class of existing strategies for hierarchy-building, we selected the *wavelet transform* and *hierarchical clustering* covering a broad range of abstraction approaches.

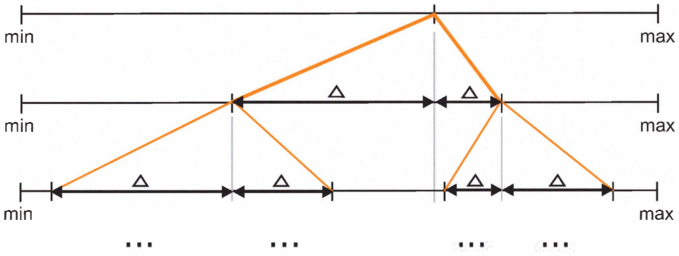

Figure 2: Recursive interval subdivision of the values of a selected dimension is based on incremental splits of a given interval in two new intervals. This approach successively reduces the value range leading to efficient and low-complexity forms of data compression founded on △-coding.

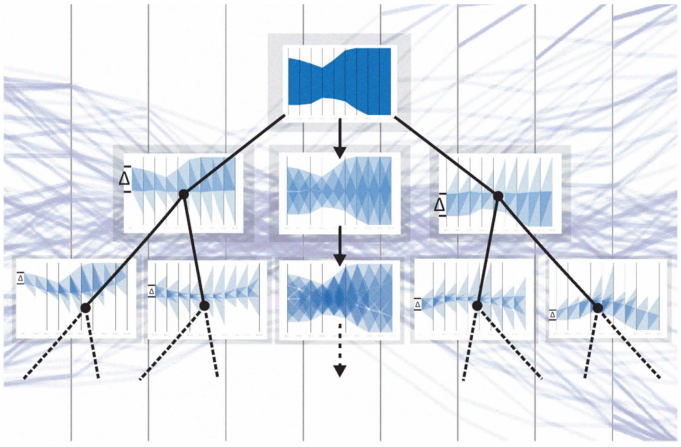

Figure 3: Illustration of an LOD hierarchy for PPC constructed by RISD: Inner nodes of the hierarchy represent aggregations associated with the corresponding data interval. All nodes of a hierarchy level contribute to the visual appearance of the presentation at the particular level (center). The presentation gains accuracy with each incremental refinement stage. Only incremental data (△) are needed to encode these changes efficiently.

3.1.1 Recursive Interval Subdivision (RISD)

RISD is a top-down approach and applied to each dimension of a given data set independently. Starting with all the values of a selected dimension, it divides a given interval into two new intervals (see Figure 2). These two intervals form a new hierarchy level and are again recursively subdivided until there is just a single value left. This value represents a leaf node of the hierarchy. As illustrated in Figure 3, the reconstruction of such a hierarchy leads to increasing detail with each additional level. The value used for separation is considered to be the aggregated item of the interval. Although, different measures are imaginable, we propose to take advantage of the median value as it is widely considered to represent a set of numerical values meaningfully. Examples of a preview sequence resulting from RISD are shown in Figure 4, a and b.

Due to the fact that each interval is subdivided into two new intervals, RISD leads to a single binary LOD hierarchy for each dimension. As the median value represents a data value that exists in the original data, there is no additional overhead by hierarchy-building that might lead to an increased data volume.

3.1.2 Wavelet Transform

The wavelet transform is widely used in image communication and was also applied to multi-dimensional numerical data. There are two main approaches: the application of an n-dimensional wavelet transform [12] leading to a single n-dimensional LOD hierarchy or the independent application of a one-dimensional wavelet transform to each of the n dimensions leading to n LOD-hierarchies. Both can be applied for PPC.

The wavelet transform is typically implemented as a bottom-up approach. Starting from the data values, aggregated values are chosen from given intervals and further merged until there is only a single value left – the root node of the hierarchy. The respective value of an aggregate is computed by a wavelet kernel. Different kernels exist, but there is basically no limitation by PPC. The simplest kernel is the Haar kernel. It merges two values into one resulting in a hierarchy similar to that of an average-based RISD. We used Daubechies filters of different length in our experiments. They exhibit better smoothing properties than a Haar wavelet-based method, which in turn leads to higher visual coherence between the different refinement levels and previews. Due to the fact that multiple values are aggregated, the kernel length influences the depth of the hierarchy. The shorter the kernel, the deeper the hierarchy. The previews shown in Figure 4, c, were created by using a filter of length eight. For more detailed information to wavelets applied to multi-dimensional numerical data the interested reader is referred to [17] and [12].

3.1.3 Hierarchical Clustering

There are many different ways to cluster multi-dimensional data hierarchically (see [6] for a classification and list), but a particular method is however not relevant to this paper. We applied an existing bottom-up approach that clusters the data in n-dimensional space. This leads to a single n-dimensional LOD hierarchy. The clustering strategy is based on the merging of data points that have a small Euclidian distance to each other. If points are within a chosen range, they are merged into a cluster. If there are no more points left that can be merged, the lowest level of the hierarchy is considered to be found. The procedure continues with the next higher level based on the cluster centers used as aggregated items. Thus, clusters and points are successively merged into super clusters until there is only one cluster left. Selected previews resulting from such a strategy are shown in Figure 4, d.

3.2 Compression and Random Access

An LOD hierarchy is highly redundant. For compression we take advantage of its inherent property that each aggregated item has a similar value with regard to all values of the associated subtree. In terms of signal processing this value represents the power that is inherent in all detail values and can be removed without loss of information. In order to achieve that, we adopted the general △-coding approach proposed in [3] for the compression of numerical data. Instead of storing absolute values for all hierarchy nodes, relative values are used. Starting from the root node, they can simply be found by removing the power associated to an aggregation item from all direct child nodes.

By taking advantage of the increasingly smaller distances between parent and child nodes within the hierarchy (see Figures 2 and 3), compression is achieved by reducing the number of bits needed to encode these distances. While all bits are required to encode the value associated to the root node, much less bits are needed at the leaf nodes. Only the associated parent nodes are required to decode individual values allowing for random access.

This strategy offers a good trade-off in complexity and compression efficiency, and ensures that all data values can be reconstructed without loss. It is also highly generic. We applied △-coding to all discussed scalability approaches. Dimensions were compressed independently.

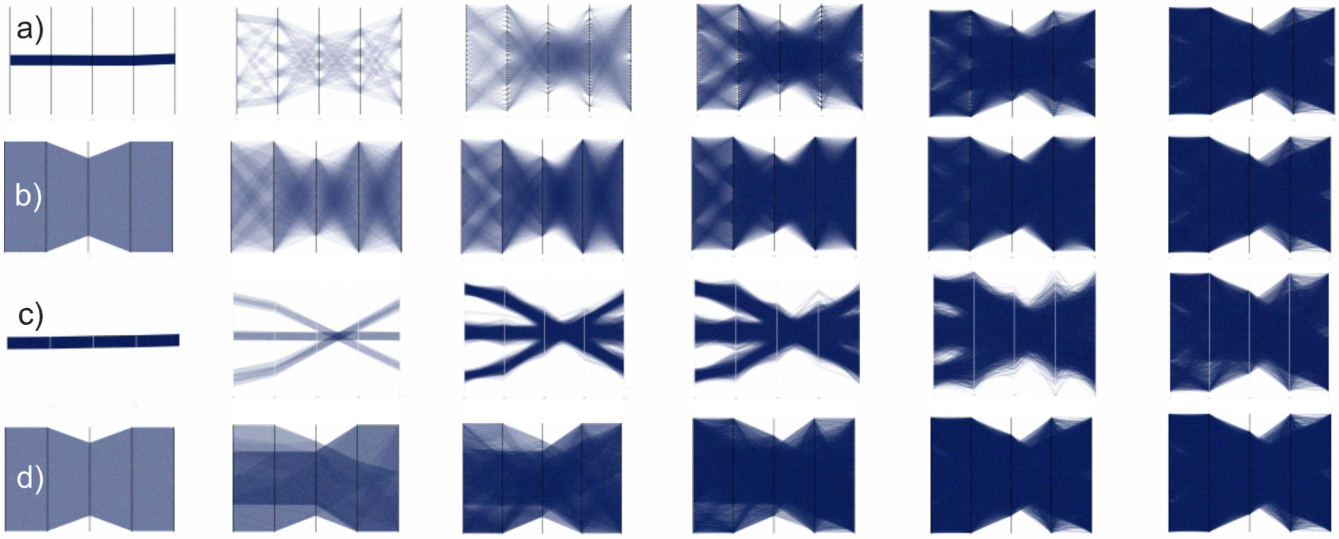

Figure 4: Visual representation of the LOD hierarchies resulting from *RISD* by lines (a) and polygons (b), from a *wavelet transform* by lines (c), and *hierarchical clustering* by polygons (d). The individual previews represent the refinement stages 1, 3, 6, 9, 11, and 14 (from left to right). The strongly different representations and refinements of an identical data set can be used to convey different properties of the data. Refinement to the highest LOD leads to the traditional PC plot for all approaches.

3.3 Visual Representation

The scalable display of a PC plot requires appropriate means for the representation of its aggregated items. To achieve this, we propose a novel approach that is based on *polylines*. We also show how *polygons* can be used for PPC. The resulting representation and refinements depend on the purpose of the progressive display and also on the used decomposition strategy.

3.3.1 Polylines

The merit of a polyline-based representation of aggregated items lies in its ability to highlight data properties that are of interest to the viewer. If the underlying principles used to generate the aggregated values are known in advance, the associated information is inherently conveyed.

All hierarchies created by the discussed scalability approaches can be represented by polylines. The resulting previews, however, strongly differ from the representation of all data points. Therefore, polylines should only be used when the aggregates have a meaning to the viewer. Aggregates resulting from RISD represent median values, the aggregates resulting from the wavelet transform and clustering average values of the given value range. Initial previews of the refinement sequence shown in Figure 4, a and c, clearly convey these properties of the underlying data set. As shown in Figures 1, b and c, this can help to detect global trends within the data.

Statistical values must be considered in context in order to be meaningful. We encoded the number of data points within a certain interval in the thickness of a line. This also conveys the approximate distribution of the data (see Figure 4, a and c). To improve visualization quality, we applied transparency that has a constant value for all lines and refinement stages.

3.3.2 Polygons

Polygons allow one to create the appearance of the fully detailed view (see Figure 4, b and d), and thus are the most natural choice for the representation of aggregated items in PC. They are especially meaningful when a range of values is to be conveyed.

All introduced strategies for hierarchy-building can be used together with a polygon-based representation. Each refinement stage leads to two (RISD, Haar wavelet) or more (Daubechies wavelets, clustering) new bands that are specified by the boundaries of the interval associated to an aggregate. Despite the use of much less graphical primitives in the previews, the use of polygons leads to over-plotting making the distinction of individual polygons difficult. To overcome this problem, we take advantage of transparency that is adaptive to the size of the primitives. It is usually high in first refinement levels and decreases towards the detailed values.

3.4 Ordering

The introduced data compression technology provides means to randomly access individual values within the LOD hierarchy. This flexibility can be used to design each preview and the whole refinement strategy dependent on pre-determined or current demands. This has crucial advantages for static and interactive visualizations.

The most simple ordering strategy is *data-driven* refinement. To support this functionality, the hierarchy is traversed breadth-first creating a single preview for each hierarchy level (see Figure 3). Ordering, refinement, and display of the data in a nearly arbitrary fashion is made possible by the modular hierarchy and options for random access (see Figure 5). For the introduced compression approach, this is only constrained by the requirement that all associated parent nodes of a considered node must be transferred first to enable its decoding. This, however, is not a strong limitation in most application domains. Two novel ordering concepts based on the general item-of-interest principle are proposed.

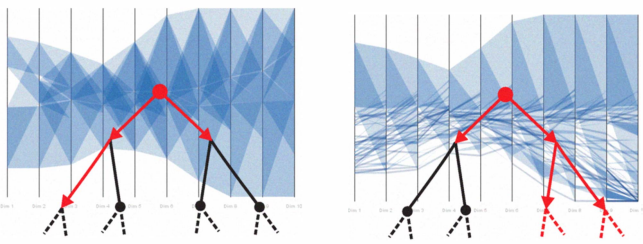

Figure 5: Ordering dependent on pre-defined or current interests: Different traversals of the LOD hierarchy lead to different previews.

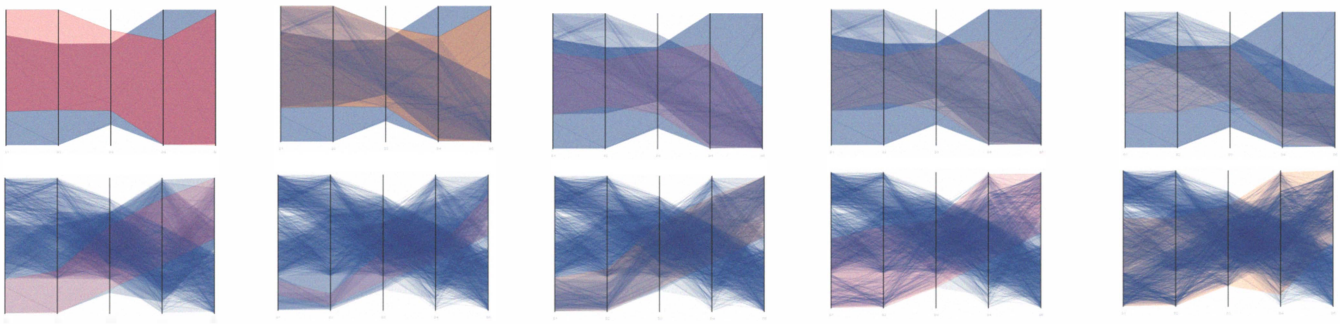

Figure 6: Cluster-of-interest refinement to prioritize and refine on a cluster basis. Red polygons represent clusters selected for refinement in the next preview, orange polygons clusters that are currently refined. Interesting clusters can be pre-defined or interactively selected.

Cluster-of-interest (COI) This concept refers to the independent refinement of individual clusters and is meant to be used with LOD hierarchies obtained by multi-dimensional clustering approaches. It allows one to highlight one or multiple selected clusters and to refine the associated values earlier than others (see Figure 6). As a result, outliers can be emphasized (see Figure 1, a) or clusters at different detail levels compared.

Dimension-of-interest (DOI) This concept refers to the prioritized refinement of individual dimensions and can be applied to scalability approaches leading to a single LOD hierarchy for each dimension (RISD, wavelets). DOI is especially useful when there are dimensions that are more important than others, e.g., as determined by methods proposed in [1] and [18]. An example for RISD assuming an interest ordering of dimensions is shown in Figure 7.

It is worth noting that in data transmission environments it must be ensured that each received data value can be identified to reconstruct the LOD hierarchy. We used a simple protocol assigning to each data value a unique id for this purpose.

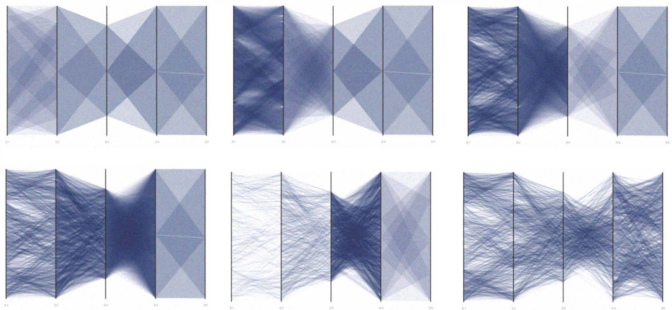

Figure 7: Dimension-of-interest refinement prioritizing important dimensions. In this example importance decreases from the left to the right axis. Dimensions may also be prioritized interactively.

4 Applications

In this section, we present application domains for PPCs that can strongly benefit from their support for efficient *data management* and provided means for improved *data presentation*. The given examples are of broad impact in data visualization and shows the wide applicability of the approach.

4.1 Data Management

4.1.1 Interactive Visualization

Interaction is an important part of data visualization systems. PPCs can support many different demands a user might have during browsing the data. Common data exploration is supported by two means: (1) scalable data representation based on the LOD hierarchy and (2) efficient strategies for data handling and transfer.

Many interactive data exploration techniques, such as hierarchical parallel coordinates [6], show the data at multiple levels-of-detail. Calculating meaningful LODs is often a complex task. PPC efficiently supports any kind of hierarchical data exploration by transferring and storing the data in a scalable manner by a one-time pre-process. The resulting aggregations can then be accessed quickly, multiple times, and during different sessions. Due to the introduced COI and DOI orderings, previews can show aggregated and data items together not limiting the user in the exploration process. Data refinement can easily be adapted to the individual and even frequently changing interests of the analyst.

Networked displays that require data transmission before data is to be displayed can take full advantage of this functionality. The tight coupling of random access and demand-driven ordering of still encoded data pieces ensure that all requests can be served in a resource efficient manner. Interactive changes during an ongoing refinement require only a re-ordering of data pieces that have not yet been transmitted enabling truly non-redundant data dissemination. The fact that only selected data pieces are transmitted leads to the crucial advantage that the transferred data volume does no longer depend on the original data source, but on the current interests of the analyst only. This is of strong advantage especially for very large data sets and strongly resource-limited devices.

4.1.2 Device Adaptation

The variety of available viewing devices increases steadily making a meaningful adaptation of the displayed data mandatory. It is not surprising that most existing solutions are based on scalable data representation [13]. PPCs are inherently scalable and therefore well-suited for device adaptation.

Device adaptation involves many different system parameters

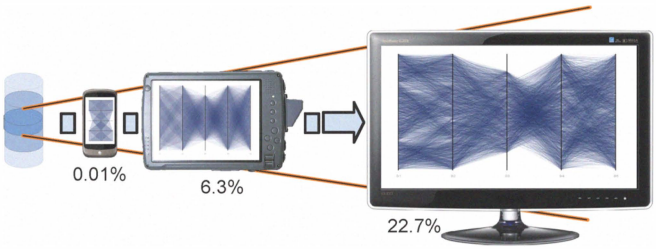

Figure 8: PPC applied to quickly adapt the presentation to the screen estate available on the respective viewing device.

and is highly complex. Prior research has already made suggestions for the use of progressive data displays for low-complexity device adaptation. Common to all available solutions is that the data is refined incrementally until the handling and display of additional items exceeds the device's capabilities in terms of computing power, bandwidth, or screen estate. Such solutions are fully supported by PPC and can be applied without modifications making PC plots instantly available for a broad variety of viewing devices (see Figure 8). More details to the implementation of such device adaptation strategies can be found in [14].

4.2 Data Presentation

Data presentation by narrative visualization strategies has gained increasing interest [16]. PPC supports such data presentations especially in their author-based and passive forms, as the Martini glass style [16]. Progressive story-telling can mainly by accomplished by a well-designed preview sequence allowing for an incremental build-up of insight about the data. First levels may show a coarse overview consisting of a few simple items or properties only. In the following, interesting parts of the presentation can be further refined to convey associated details. Thereby, context is always provided by the higher level aggregations of unrefined items. An implementation of such a story based on PPC is straightforward as the overview is inherent in the LOD hierarchy and the interest-driven refinement sequence can be achieved by the introduced data ordering concepts. Data ordering may be initially pre-defined and interactively modified to accommodate individual interests.

The different line-based representations were designed to convey information about statistical properties of the data. Thus, no specific sequencing, but only simple data-driven refinement is needed to create narrations that aim to convey statistics about the underlying data set. Two typical examples are illustrated in Figure 1. Telling stories about *outliers* in a data set can be achieved by taking advantage of the proposed hierarchical clustering approach. It keeps outliers separate during hierarchy-building positioning them at high levels. Thus, they will appear in early refinement levels and shown in relation to the data clusters (see Figure 1, a). Another example is the support for the visualization of *trends*. The proposed RISD and wavelet-based hierarchy-building are based on the averaging of values. As shown in Figure 1, b, c and d this leads to an inherent highlighting of trends that are otherwise hidden by clutter.

5 Results

In this section we show the results obtained from our implementation of PPC. Thereby, we focus on aspects related to our goal for efficient *data management* and means for improved *data presentation*. We also comment on some of the *drawbacks* of the approach.

5.1 Data Management

We discuss aspects that are valid for stationary and networked environments: *complexity of hierarchy-building and encoding*, the resulting *data volume*, and the time required for *decoding and display*. Ordering has very low complexity and is therefore neglected. The results were obtained by a PPC implementation in JAVA taking advantage of the PREFUSE toolkit [7]. The experiments were executed on a modern desktop computer equipped with an Intel Core i7-920 and 6GB RAM.

5.1.1 Complexity of Hierarchy-Building and Encoding

Due to their inherent scalability, PPCs are basically not limited in the number of data points that can be processed. Most resources are consumed by one-time hierarchy-building. The results we obtained regarding the computing power required by the introduced scalability approaches and compression using △-coding are depicted in

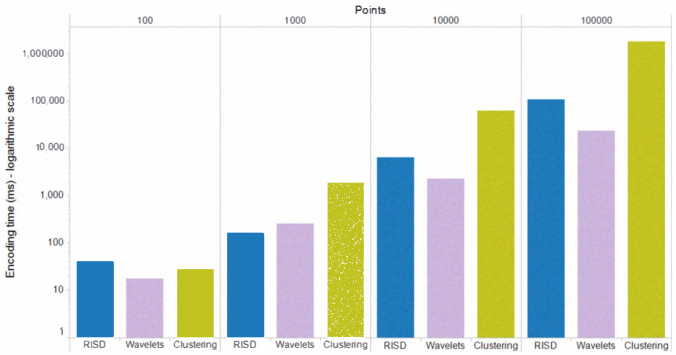

Figure 9: Time needed to calculate and △-code the LOD hierarchy for data sets of different volumes using the discussed scalability approaches.

Figure 9. We experimented with data sets ranging from small volumes to volumes that are still challenging for current data management and display solutions for PC.

RISD and wavelet transformation exhibit a similar performance for all data sets. The gain in higher detail levels is based on the fact that the used wavelet transform was implemented in C++. Thus, it is up to 10 times faster. As the complexity of the RISD algorithm is comparable to the used wavelet transform, similar performance of its C++ implementation can be expected. The complexity of the used hierarchical clustering approach increases exponentially with each additional detail level. This is due to the fact that it compares in a pair-wise fashion the different points to form clusters. It shows highest complexity.

△-coding has very low complexity as only the relevant distances of a data value to its parent value must be determined. This method only requires fast memory look-up operations and simple calculations.

5.1.2 Data Volume

Figure 10 illustrates the volumes of the data set plotted in Figure 1 (50,000 points, five dimensions) resulting from the application of *no*, *constrained range-*, and △*-coding*. The shown accumulated contributions for each level indicate the amount of data required to provide a certain LOD. The hierarchy was created using RISD.

For all set-ups very little data is required to provide the first previews (see also Figure 1). The assessments obtained for the applica-

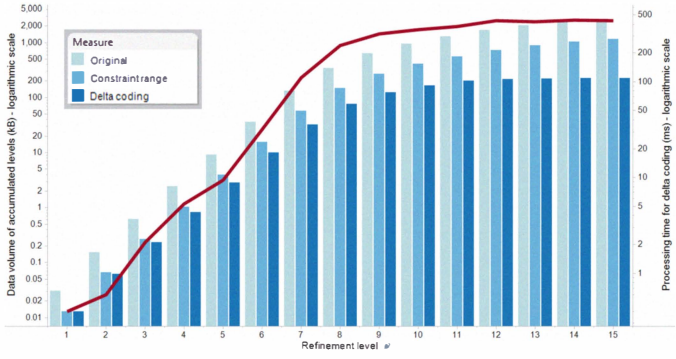

Figure 10: Data volume (blue bars) associated with the different refinement stages using RISD for *no*, *constraint range coding*, and △*-coding*. The computing power (red line) needed to decode and display △-coded data shows the correlation with data volume and the low complexity of the processes.

tion of no compression show that the data increases exponentially with each incremental hierarchy level. The increase is smaller at higher-detail levels as interval subdivision does often not lead to the same depth for all subintervals. As the data is stored in a common data type, the data volume is largest. Contrary, constraint range coding determines the range of all values in order to calculate the minimal number of bits required to store an individual value. All values are encoded in the resulting precision. This leads to a reduction of the required data volume, as often not the full precision provided by common data types is required. The best compression performance is achieved by △-coding determining the required number of bits for each individual interval and value. While its performance is almost identical at first decomposition levels, the number of used bits decreases constantly. Compared to the other strategies, there is hardly any increase in the volume by data residing at higher hierarchy levels. With regard to all data, △-coding requires only 10% of the data volume compared to no compression and 20% compared to the simple constraint range strategy. In other words, *△-coding* reduces the response times of PPC in networked environments by factor 10.

5.1.3 Decoding and Display

To illustrate the efforts required to provide the different incremental previews, we assessed the time that is needed to partially decode and display the associated data using △-coding. We used the same data set as for the illustration of the required data volume. Figure 10 shows the expected correlation between execution time and the amount of data to be handled. As decoding of △-coded data has little complexity, these processes are fast.

5.2 Data Presentation

Although it is widely accepted that hierarchical approaches can improve data presentation, this has never been verified for progressive data displays. We conducted a user study in order to evaluate whether PPC can provide a better conveyance of data properties compared to traditional PC. We focused on the detection of patterns often stated in the literature as one of the advantages of PC [10]. Our study was designed to answer the following questions:

Q1	Do PPC achieve a higher correctness in the detection of patterns within the data?
Q2	Do previews enable the viewer to detect patterns before all data is available?
Q3	Do PPC lead to an improved user experience?

5.2.1 Design and Methodology

We used the Amazon Mechanical Turk micro-task service as the general test environment. 43 distinct participants covering the whole spectrum of common users were recruited (no (13%), basic (37%), advanced (30%), and expert (14%) knowledge in data visualization; 6% did not state their expertise).

We created a test series comparing the performance of PC and PPC for various set-ups. It consisted of 20 individual test pages of identical structure. Section 1 asked for user expertise. Sections 2 and 3 each consisted of identical questions concerning two different data sets either visualized by PC or PPC. To answer Q1 we provided two multiple choice questions: *"Can you spot any of the example pattern in the plot? Where?"* and *"Which of the example pattern is it?"*. To answer Q2 we asked: *"In what refinement level did you spot the pattern first?"*. Section 4 was dedicated to receive user feedback to answer Q3. We required the participants to state their opinion to assistance (*"What kind of presentation did assist you most in decision making?"* and acceptance (*"What kind of presentation did you like more?"*.

With regard to the many options to design a PPC plot, we selected multi-dimensional clustering for hierarchy-building and transparent polygons for data representation. The data were refined uniformly with one hierarchy level per preview. We used 20 artificial data sets with different properties. One out of four different patterns had been included in each data set at an arbitrary dimension. These patterns are common and identical to the ones used in [10]. To increase the difficulty of the task, we added noise in different levels (0%, 10%, 20%, 40%, 50%). Each data set contained 2000 5-dimensional data points that were randomly distributed. We decided to use small data sets to guarantee appropriate data processing on all hardware of the heterogeneous Mechanical Turk environment. A PC plot was represented by a static image, a PPC by a JAVA applet, both of dimensions small enough to cause visual clutter with all considered set-ups.

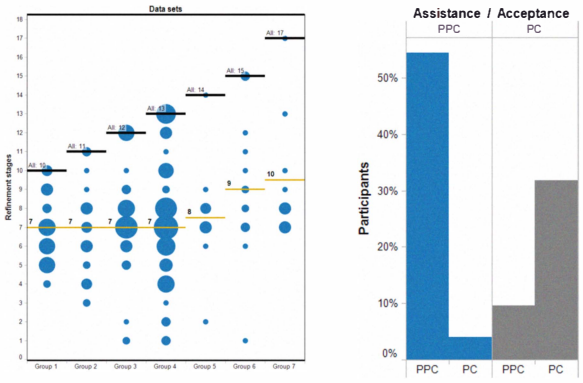

Figure 11: The number of refinement levels required to detect a pattern (left) indicates that the majority of users (encoded by point size) achieved the task by previews only. The median of the stated levels (yellow reference lines) is significantly different to the number of all available levels (black reference lines). There was also a large preference for PPC with regard to assistance and acceptance (right). We interpret this as an indication for an improved user experience.

5.2.2 Results

Before analyzing the given answers statistically, we removed the data of three participants who misused our test platform. They were identified by their arbitrary answers to the questions.

Correctness in pattern detection Although PPC always exhibited a higher average accuracy and much lower standard deviation, we found no significant difference in the correctness of pattern detection compared to PC. Similar results were obtained for all individual data sets and also when correctness is compared between different noise levels. This can be explained by the fact that the only difference between both approaches lies in the many additional previews to the data provided by PPC. If they do not highlight possible patterns, the correctness of their detection will not be higher. This was obviously the case for the applied generic hierarchical clustering and shows that a proper LOD hierarchy is important for the conveyance of data properties.
Q1 - Reasoning: The tested implementation of PPC *did not achieve a significantly higher correctness in pattern detection.*

Refinement level of detection Our results show that the majority of PPC users detected the embedded pattern during refinement (cf. Figure 11, left). The respective refinement levels exhibit a Gaussian distribution that is centered around the median. This allows for the conclusion that participants either saw the pattern at a refinement level that is close to the median or required all details which corresponds to the standard PC display. Depending on pattern and noise level, the distribution is often skewed towards the need for less refinement levels. Thus, most of

the perceived detections took place in the second and third quarter of the refinement process. We conclude that the provided visual aggregations achieved a close imitation of the data representation in high LOD levels and pattern detection can be accomplished much earlier than with PC. The associated saving in data volume required to detect the patterns is significant. This, in turn, leads to even higher gains in data management efficiency. Calculated across all tests and data sets, only 37.04% of all available data were required to achieve similar results in pattern detection.
Q2 - Reasoning: The tested implementation of PPC *enabled the viewer to detect patterns long before all data is available*.

User preferences As shown in Figure 11, right, there was a strong response for the progressive approach when participants were asked for assistance and acceptance. 54% of all users voted for PPC in both aspects. The majority of them (58%) stated that PPC assisted them better in accomplishing the task than PC. We relate this to the many uncluttered views, the incremental build-up of insight, and the animated data display provided by PPC.
Q3 - Reasoning: The tested implementation of PPC *led to an improved user experience*.

5.3 Drawbacks

Progressive refinement is based on a single data hierarchy that is incrementally streamed to the client. This also imposes drawbacks. Probably the largest issue is the *lack of support for data sets that change frequently*. In this case the hierarchy must be renewed often and data communication started from scratch leading to higher resource consumption. Another drawback directly related to the hierarchy is the *inability of progression to handle data sources that cannot be aggregated*. Many and meaningful levels of aggregation are required. When only a limited number can be provided, the available functionality is strongly constrained. Transferring data in a *hierarchical representation usually also introduces redundancies*. This, however, is not a strong limitation as the redundancies can be removed by suitable compression or appropriate scalability approaches, e.g., RISD.

6 Conclusions and Possible Future Research

We introduced progressive parallel coordinates (PPC) with the goal to overcome existing data management and presentation issues caused by the handling of large data volumes. The advantages of PPC are explained by efficient data compression, access, and transfer as well as their tight coupling with the final display. Data previews support gaining insight early on with much less data and visual clutter. The application domains for PPCs are broad and timely. The results we obtained from empirical tests show that the volume of the data to be stored or transmitted can often be decreased be a factor of 10. Options for random access decrease the data to be transmitted to the essential pieces making possible the handling of large volumes on strongly resource-limited hardware. In a user study we revealed that by using PPC on average only 37% of all data is required to achieve a similar degree of pattern detection as with the standard approach. Although it was preferred by the majority of participants, we also found that the tested implementation of PPCs did not lead to higher detection accuracy.

Future research could be directed at each of the requirements of PPC. One example is hierarchy-building and recursive interval subdivision that might focus on different statistical measures. The conducted user study represents a first attempt only to quantify the eligibility of PPCs. A more comprehensive study may obtain a better understanding concerning their ability to convey data properties. In conjunction with the development of components able to highlight patterns, outliers, or trends, this may provide the required profound evaluation of the merits of PPCs for data presentation.

Acknowledgements

The authors gratefully acknowledges the support of Deutsche Forschungsgemeinschaft (DFG) for funding this research (RO3755/1-1).

References

[1] M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the 1998 IEEE Symposium on Information Visualization*, page 52, North Carolina, 1998. IEEE Computer Society.

[2] Y. Chee. Survey of progressive image transmission methods. *International Journal of Imaging Systems and Technology*, 10(1):3–19, 1999.

[3] M. Deering. Geometry compression. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 13–20, New York, NY, USA, 1995. ACM.

[4] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007.

[5] N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 99, 2009.

[6] Y. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the 10th IEEE Visualization 1999 Conference*, Washington, DC, USA, 1999. IEEE Computer Society.

[7] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, page 421–430, New York, NY, USA, 2005. ACM.

[8] H. Hoppe. Progressive meshes. *Computer Graphics*, 30(Annual Conference Series):99–108, 1996.

[9] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009.

[10] J. Johansson, C. Forsell, M. Lind, and M. Cooper. Perceiving patterns in parallel coordinates: determining thresholds for identification of relationships. *Information Visualization*, 7:152–162, Apr. 2008.

[11] D. Laur and P. Hanrahan. Hierarchical splatting: a progressive refinement algorithm for volume rendering. In *SIGGRAPH'91*, pages 285–288, 1991.

[12] T. Li, Q. Li, S. Zhu, and M. Ogihara. A survey on wavelet applications in data mining. *ACM SIGKDD Explorations Newsletter*, 4:49–68, Dec. 2002.

[13] M. Novotny and H. Hauser. Outlier-Preserving Focus+Context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.

[14] R. Rosenbaum, A. Gimenez, H. Schumann, and B. Hamann. A flexible low-complexity device adaptation approach for data presentation. In *Proceedings of Electronic Imaging - Visualization and Data Analysis 2011*, 2011.

[15] R. Rosenbaum and H. Schumann. Progressive refinement - more than a means to overcome limited bandwidth. In *Proceedings of Electronic Imaging - Visualization and Data Analysis 2009*, Jan. 2009.

[16] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, 2010.

[17] P. C. Wong and R. D. Bergeron. Multiresolution multidimensional wavelet brushing. In *Visualization Conference, IEEE*, volume 0, page 141, Los Alamitos, CA, USA, 1996. IEEE Computer Society.

[18] J. Yang, M. Ward, E. Rundensteiner, and S. Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *Proceedings of the Symposium on Data Visualisation*, 2003.

[19] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical displays: a general framework for visualization and exploration of large multivariate data sets. *Computers & Graphics*, 27(2):265–283, Apr. 2003.

[20] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum (Proceedings of EuroVis'08)*, 27(3), 2008.