Progressive refinement – more than a means to overcome limited bandwidth

René Rosenbaum and Heidrun Schumann
The University of Rostock, Germany

ABSTRACT

Progressive refinement is commonly understood as a means to solve problems imposed by limited system resources. In this publication, we apply this technology as a novel approach for information presentation and device adaptation. The progressive refinement is able to handle different kinds of data and consists of innovative ideas to overcome the multiple issues imposed by large data volumes. The key feature is the mature use of multiple incremental previews to the data. This leads to a temporal deskew of the information to be presented and provides a causal flow in terms of a *tour-through-the-data*. Such a presentation is scalable leading to a significantly simplified *adaptation to the available resources*, short response times, and reduced visual clutter. Due to its rather beneficial properties and feedback we received from first implementations, we state that there is high potential of progressive refinement far beyond its currently addressed application context.

Keywords: Progressive refinement, Device adaptation, Information visualization, Scientific visualization, Visualization system

1. INTRODUCTION

Dealing with massive data sets is one of the main challenges in visualization. The huge amount of data affects all visualization stages from the preparation of the data to its display to the viewer – there is simply too much data to be suitably processed and presented. This results in long response times and heavily overloaded displays. While already true for most stationary hardware, this especially applies for mobile environments characterized by strongly limited resources and a heterogenous device scenery. This problem is not new and many different approaches addressing single or multiple aspects of these issues have been proposed.[1–3] Due to a still missing final solution, it is eligible to explore new ways to visualize and present data.

This contribution motivates such a novel approach. We propose the application of *progressive refinement* (progression) in order (1) to improve the conveyance of information about the data, (2) to simplify the adaptation to the viewing device, and (3) to achieve a significant overall reduction in the resource consumption. Although, progressive refinement is already established to overcome limitations in the available resources,[4–6] we will show that the approach offers much more. During progressive display previews with multiple increasing detail levels of the data are presented allowing to convey information in early processing or transmission stages. Interestingly, progression supports many accepted visualization principles and is applicable to different data types and devices. With the aim to provide a *tour-through-the-data*, we will show that it may also be designed to convey aspects of the data. In order to avoids long response times and visual clutter, the application of progression also simplifies the *adaptation of the presentation* to the viewing device. The eligibility of the approach is further underlined by the highly integrated processing throughout the visualization process *offering much more performance* compared to existing solutions. Although, these benefits suggest much potential for progression, there are still many open questions to answer to make it a valid approach in visualization.

This publication is structured as follows: To show the novelty of our proposal, Section 2 reviews the State of Art in related research. Section 3 is concerned with the basic concepts and requirements of progression serving in Section 4 as the foundation to introduce the new application areas *progressive information presentation* and *device adaptation*. Section 5 introduces a first proposal for a generic description of progression in visualization. Properties and open questions of the approach are discussed in Section 6. Conclusions and directions for future work close this contribution in Section 7.

Further author information: (Send correspondence to René Rosenbaum)
René Rosenbaum: E-mail: rosenbaum@ieee.org, Telephone: ++49-(0)381-4987494
Heidrun Schumann: E-mail: schumann@informatik.uni-rostock.de, Telephone: ++49-(0)381-4987490

Figure 1. Progressive refinement of imagery provides first impressions of the contents and may also be used to reduce consumed system resources by stopping data processing or transmission whenever required.

2. RELATED WORK

By talking about progressive refinement the reader might be instantly reminded on the early days of the World Wide Web, when limited bandwidth was a real issue.[7] To shorten the long latency times during the loading of imagery, the proposal of dynamically refining contents was a real relief – with little data received, first conclusions could already be drawn (cf. to Figure 1). Due to its success, the approach has later also been applied to other kind of data in order to overcome resource limitations in computational power[4] or memory.[5] Image and video communication, however, have always been its main application field.[6,8]

In order to apply progression suitably, all previews must be simple to interpret. Such views are easier to create for data with spatial reference than for abstract data. Thus, there are many progressive techniques for the presentation of geometrical data, such as meshes[9] or scientific data.[10] However, there are also first proposals for abstract data. Everitt and Yee[4] use progression to visualize the result of massive data queries. Each of the previews shows a combination of already retrieved data with abstractions of items still to come. This allows for conveying information about the data even in early stages of such complex tasks and saves resources by the option to finish the refinement process at any time. Beside such technical concerns, however, progression is also able to support semantic aspects. By creating well-defined successive previews, it is possible to highlight data characteristics and to significantly simplify device adaptation.[11] In spite of these advantages, it is remarkable that progressive technology has only sparely been applied to abstract data. This might be due to the different constraints and basic conditions which hamper a suitable parametrization. Furthermore, there is no literature to progression, which abstracts from specific problems and provides generally valid statements to the fundamentals, requirements, and general procedure of the approach. This would be of great value for the understanding of the feature as well as the development of future technology and suitable applications.

Due to long response times and *visual clutter*,[12] large data volumes can often not completely processed and presented. To overcome these problems, the *Information seeking mantra*[13] – *Overview first, zoom and filter, then details-on-demand* – is one of the mostly applied and well-accepted paradigms. The progression approach is quite similar to the mantra. It provides an initial overview of the data which is later on successively increased in detail. Due to early availability of data abstractions, first insight is gained much in advance to the final representation. This requires little data, as the volume of data becomes significant only in the later, more detailed previews. The information seeking mantra, however, only describes the basic process but does not provide statements to the actual visual representation of the data. To achieve this, numerous visualization techniques for different kinds of data,[2] devices,[3] and applications[1] have been developed. One approach, often applied in existing visualization technology, is the use of multiple parallel views to the data. Contrary to the *partition in time* achieved by progression this represents a *partition in space* and thus further aggravates existing device limitations especially regarding display space.

Due to strong heterogeneity in the currently available device range, the capacity to handle certain work items varies strongly and may be less than required for a certain task. The needed adaptation of the presentation is an important problem not only existent in visualization. A research field especially dedicated to solve the associated issues is *Multiple User-Interfaces* (MUI).[14] Interestingly, its aim to achieve access to centrally managed information and the resource-driven display have strong analogy to common visualization technology[15] and the progression approach. As MUI comprise many benefits for any application field requiring visual output in heterogenous environments, first works in visualization strive to apply this approach to different kinds of data and tasks.[16,17] This, however, is still a very novel research topic. Except for digital imagery, progression has yet not been proposed for this purpose.

3. THE FUNDAMENTALS AND REQUIREMENTS OF PROGRESSION

Successive refinement of raster imagery is one of the commonly known applications of progression and found in many implementations for resource-limited hardware. There is simply a strong need for technology able to deal with the constantly growing increase in detail. Multiple strategies for progressively refining raster image contents regarding resolution, quality, or color have been proposed. Common to all strategies, however, is that there are multiple successive previews each adding incremental detail to the display. Concepts of *Regions of Interest* (RoI) and *Levels of Detail* (LoD)[18,19] known from common computer graphics thereby allow for adaptation of the refinement to external needs. Latest developments even support resource-efficient reaction on interactive changes.[20] This is possible due to two generic procedures:

1. **Transfer into another data structure:** To prepare raster imagery for progressive refinement the content is typically transferred into a more suitable data structure. The goal of this process is to achieve (1) *scalability* of the content, (2) *random access* to image regions, and (3) better *encoding efficiency*. Thereby, the general problem is to allow for a later *unambiguous interpretation* of the scaled abstracted contents. The fact that progression successively adds detail to the display suggests to use a tree structure (LoD-hierarchy) to organize and access the different data abstractions. A typical example to achieve these aims for raster imagery is the well-known Discrete Wavelet Transform (DWT).[20]

2. **Traversal of the data structure:** The evolving data hierarchy may be traversed in different orders. This order strongly influences the resulting previews and presentation and must be chosen in accordance to the respective presentation *goals*. As full detail is often not required or possible, the length of the traversal path may be chosen depend on the capabilities of the viewing device and the user preferences. The implementation of the RoI/LoD-concept based on the traversal of an LoD-hierarchy is the most superior option in interactive environments.[20]

Technology found in literature typically assumes a client/server environment to use the full potential of progression. Thereby, an appropriate data structure is selected by an *author*, stored for multiple use at server side, and traversed dependent on the *goals and demands* of author, user, or device. This implements the paradigm – *encode once, decode many ways*[21] – significantly reducing the consumed system resources within the whole environment. The viewing device just displays the received previews successively.

Although raster image data is well suited to support progression, the stated properties and requirements may also be applied to other kinds of data. The following list generalizes the statements provided for imagery to generally valid requirements of progressive refinement:

Goal-Orientation There are a variety of ways to refine the content. In order to steer the presentation suitably, meaningful goals to be achieved during progressive refinement must be defined.

The goals may cover technical as well as semantic aspects. Very common technical directions are the alignment to system constraints as computing power or bandwidth. A typical semantic goal used in most already existing progressive technology is the support of a data overview. As described in Section 4, however, progression is also able to support other presentation goals.

Scalability Progressive presentations show abstractions of the data in multiple subsequent previews with increasing LoD. This requires scalable access to the data and its representation. If not inherent, this property must be included.

Generating scalable data representations is not a simple task and depends on various conditions, as the current structure, kind of data, and even the goal of the presentation. Current successfully applied approaches are abstraction, aggregation, clustering, and transformation.[22–24] Along with these strategies, advanced subsampling and filtering may also be seen as valid approaches as long as the selected values allow for suitable abstraction.

Unambitious interpretation As all previews show abstractions of the data, an unambitious interpretation of each preview must always be guaranteed in order to avoid drawing wrong conclusions (*Apprehension* *principle*[25]). It is also required that each subsequent preview logically builds on its ancestor in order to provide a causal flow in gaining information about the data. This also implies that the whole presentation is unambiguous. In order to support the effortless achievement of the presentation goal, it is also desirable that each preview and the whole refinement process is easy to interpret (*Congruence principle*[25]).

Random access Often not all data is of same importance. To allow for prioritization during refinement and presentation, single data values and their abstractions must be independently accessed, attributed, and assigned to the different previews. Random access is also crucial for a dynamic and resource-efficient adaptation to interactive changes.

Random access strongly depends on the internal structure the data is transferred to and is used by concepts like *Regions of Interest, Geometry of Interest,* or *Item of Interest*.[11]

Encoding Huge data volumes require efficient encoding to avoid problems in transmission and storage. This especially applies if additional abstractions of the data are stored for scalability reasons.

To accomplish an eligible encoding of scalable data, it is useful that in order to add detail to data items only an increment must be added. Typical examples for such an efficient handling are the DWT and *delta coding*.[26]

It is worth noting that some of the stated requirements, e.g. random access and encoding, are contrary leading to the problem that not all demands can be fully implemented. As already shown for image data, this may be solved by developing solutions that consider the specifics of the environment, goals, and user preferences[20] and which provide reasonable trade-offs in the implementation of the respective demands.

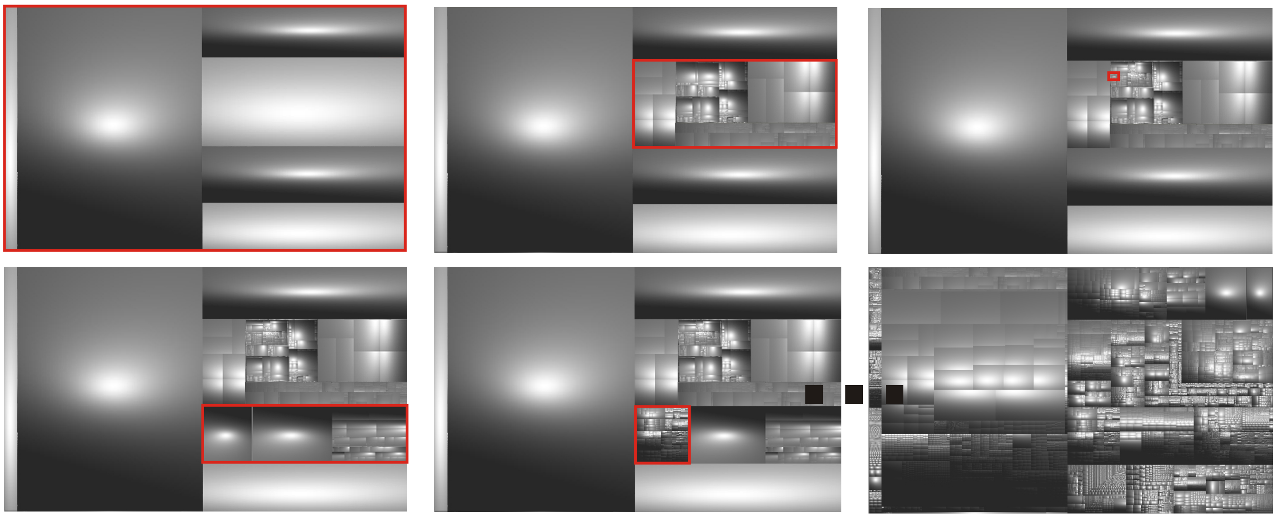

Figure 2. A *progressive treemap* designed to convey location, context, and structural belonging of certain data items (red marks indicate refined data and are not part of the presentation).

4. PROGRESSIVE INFORMATION PRESENTATION

4.1 A novel kind of information display

Beside visual analysis and exploration of characteristic data aspects, an eligible presentation of the results is one of the main objectives in visualization. Due to the sheer volume of the data and the imposed visual clutter meaningful results are often difficult to achieve. The beneficial properties of progression allow for a completely novel kind of animated information display that is able (1) to improve the *conveyance of information* about the data and (2) to simplify the *adaptation of the presentation* to the viewing device.

Conveyance of information Instead of showing multiple views at the same time, progression is able to provide a successive presentation sequence. As each preview adds detail to the presentation, this allows for an incremental buildup of knowledge about the data. Progressive presentations may usually be pre-designed by an author picking a valid data structure for suitable abstraction and scalability, adapting the traversal of the structure to existing demands, and selecting a proper visual representation of the different data items. Thus, each preview and the whole sequence may be aligned to support specified presentation goals. The thereby applied paradigm – *progression instead of selection* – allows for the extension of the traditional data exploration by a progressive presentation in terms of a *tour-through-the-data*. In case the intended goals change during presentation, however, it is also possible to interactively influence the presentation. To achieve this, demand-driven modification of the traversal adapts the following previews to the new demands.

Although progression seems similar to traditional animation, the different views are tightly bound and not independent. This is an important difference to most related approaches (e.g.,[27]). However, many of the properties found in successful animations,[25] as provided means to help users to maintain a mental map of the data,[28] are supported.

Adaptation of the presentation Properties and resources of currently available viewing devices are manyfold. Thus, presentation author and viewer have often no clues if data to be presented exceeds the computing power, bandwidth, or screen space of the device-at-hand. If so, the user is confronted with long response rates and visual clutter. By taking advantage of progression such situations can be avoided – the presentation is shown as long as the device is able to provide the required resources. Thereby, the used overview-then-detail principle nicely corresponds to the increasing resource consumption during presentation – first previews require little, later previews much resources. If it is estimated that the consumption of one resource exceeds the capabilities of the device, e.g. by measuring response rates or clutter,[29] the presentation stops. Thus, the visual representation is highly adapted to the viewing context. To achieve this, again, only essential parts of one single *"multi-purpose" data structure* determined by the respective traversal path and depth are required. This underlines the appropriateness of the progressive approach for this application domain.

The related beneficial properties are now demonstrated by two case studies for progressive information display. Thereby, we focus on the semantic aspects of information presentation and show possible implementations of the required procedures. Technical aspects of the approach are discussed in Section 6 by showing the achieved save of resources. While the advantage of the first study is achieved by an adaptation of the *traversal* stage, the benefits of the second study are mostly founded on a use of a sophisticated *data structure* and appropriate *visual representation* of the data.

4.2 Progressive treemaps

The treemap is an approach to visualize structural dependencies and properties of hierarchical data.[30] It provides a single intuitive view to the data and is therefore well-known and widely accepted by the community. However, if applied to large data sets the approach shows certain drawbacks. The most important disadvantage is that displayed data items become very small and thus can not be identified and recognized anymore. Often also the structural belonging of items and levels within the hierarchy is lost.[30]

In order to overcome these problems, we propose a progressive presentation of the treemap instead of its traditional static display. The goal of the presentation is to emphasize structural dependencies and context of single or multiple data items. This allows for an enhanced visual identification, localization, and relation of these items to others. The general layout and graphical primitives of the *visual representation* are mostly predetermined by the underlying treemap visualization. Single items of the hierarchy are displayed as nested rectangles. As the provided *data structure* is already a valid LoD hierarchy[31] the transformation of the original data may be skipped. Due to this, visualization techniques for hierarchical data are generally well suited for progression. Of crucial interest to achieve the strived presentation goals is the *traversal* of the data structure. For each node of the hierarchy, a geometrical primitive, a rectangle in this case, is available after mapping. The traversal stage determines which subtrees of the hierarchy are traversed and thus, when belonging primitives are included into a preview. Thereby the concept of Geometry of Interest[11] is applied. It allows for flexible prioritization between geometry items and determines the formation of the resulting presentation. Again, not all available geometry may be considered to save systems resources. During presentation, the data is simply be viewed by a successive preview-wise display. The viewer can always manipulate a pre-defined presentation sequence by selecting interesting subtrees. The traversal is adapted and belonging geometry is prioritized during generation and transfer of next previews.

A visual example of this strategy is shown in Figure 2. To provide context and to fulfill the overview-then-detail principle, the progressive presentation starts with the first level of the LoD hierarchy (Figure 2/top-left). In order to prioritize a certain item, now only primitives of the belonging subtree are handled and displayed. This leads to the refinement of the item and the surrounding areas (Figures 2/top-middle and right) and thus to further context and information to its structural dependencies and relations. Such local refinement is another general advantage of progression. Inconsistencies between two subsequent views attract attention of the viewer to the belonging regions. As soon as the item is displayed in full detail, our example continues with the refinement of another item in low LoD (Figures 2/bottom-left). Here, the refinement stops at the belonging inner node of the hierarchy accordingly (Figures 2/bottom-middle). By a sequential or simultaneous refinement of single or multiple items, the presented example continues until all data is displayed (cf. Fig. 2/bottom-right). In case the strived information is already conveyed or system resources are used to capacity the presentation may be finished at any time.

Figure 3. One possible implementation of *progressive parallel coordinates* and its application to an uncomplex adaptation to multiple viewing devices.

4.3 Progressive parallel coordinates

The second case study is concerned with parallel coordinates[32] (PC). Similar to treemap, the PC approach is well-accepted and constantly enhanced. However, if large data volumes are to be presented, visual clutter is a serious issue. We apply the progressive approach to enhance conveyance of information about the data even if their amount is huge. PC have been applied to a variety of purposes, e.g. the highlighting of correlations, clusters, or outliers. Although, these aims might also be supported, we want to show the potential of progressive presentations by two distinct presentation goals: (1) to provide different overviews to the data and (2) to convey statistical information about the data. The first goal is of crucial importance if the representation is to be shown on multiple and heterogenous viewing devices. The second goal allows insight into the data, their distribution, and structure. Despite the fact that both aims and the belonging visual representation of the data are rather different, a quite similar parametrization is applied.

We chose abstract multivariate quantitative data without an inherently given hierarchy as source to be presented. Thus, the data must be transformed. This is achieved directly in data space by a successive attribute-wise interval subdivision based on the statistical median value. Although, many other hierarchisation strategies and criteria are imaginable,[33] this method is simple to retrace and comprehend. Furthermore, it allows for efficient and flexible compression of the hierarchy by delta coding.[11] The result is an appropriate *data structure* consisting of a single LoD hierarchy for each data attribute. Due to the fact the data has been transformed into another structure, its appropriate *visual representation* is crucial. Thereby, the unrestricted mapping of the data to geometry opens up for a strong alignment of this process to the respective visualization goal. Consequently, we propose an independent mapping for each considered goal. Common to both, however, is the general procedure to connect belonging points on adjacent axes by geometry. To provide an overview of the data (Presentation 1), we propose to adapt line strength and shape dependent on the position of the respective preview within the presentation sequence. First previews show only a few thick and bulky lines, later previews many fine lines (cf. to Figure 3). This allows for a constant overall appearance of the data, although later views provide much more detail. In order to allow for the communication of statistical information about the data (Presentation 2), a different mapping must be applied. As shown in Figure 4, the line strength depends on the data distribution within the respective interval. All sub intervals m on the adjacent axes a_i and a_{i+1} $(0 \leq i < n-1$, n states number of axes) are connected by an individual line. If there are many data items within interval $int_{a_i,j}$ belonging to the interval $int_{a_{i+1},k}$ $(0 \leq j, k \leq m)$ on the adjacent axis, a thick solid line is drawn. If there are no corresponding data items the line is transparent. The origin and end of each line are determined by the median of the respective interval provided by the respective node of the data structure. Thus, important statistical information about the distribution of the data is conveyed and increased with each provided preview. To support this, only primitive *traversal* equal for both kinds of presentation is required. The produced geometry is sequenced by a width-first traversal of the hierarchy, whereby all nodes of a certain level are assigned to a single preview. In case interactive prioritization of single items or whole hierarchy subtrees is required, a traversal strategy similar to that described for the progressive treemap may be applied.

Examples for the described procedure are depicted in Figures 3 and 4. The different stages of Presentation 1 (Figure 3) have almost the same visual appearance. The presentation, however, is much more detailed in later stages. Contrary, the individual previews of Presentation 2 (Figure 4/1-3) are quite different from the representation of all data (Figure 4/4). Due to the rather different aim, it is more a *statistical than a visual abstraction.* However, the presentation conveys much more information. Preview 1 and 2 clearly indicate that most of the data have higher values. Preview 3 allows for even more statements: 1.) a strong inverse correlation between the first and second axis, 2.) an inverse correlation between the forth and fifth axis, 3.) a correlation between the second and third, and third and forth axis, and 4.) the second interval at top of the sixth axis includes the highest number of data elements of all intervals on this axis. Especially, the 1. and 4. statement can not be derived from a cluttered representation of all values. As shown, the presentation on a single viewing device stops if the respective resources are exceeded.

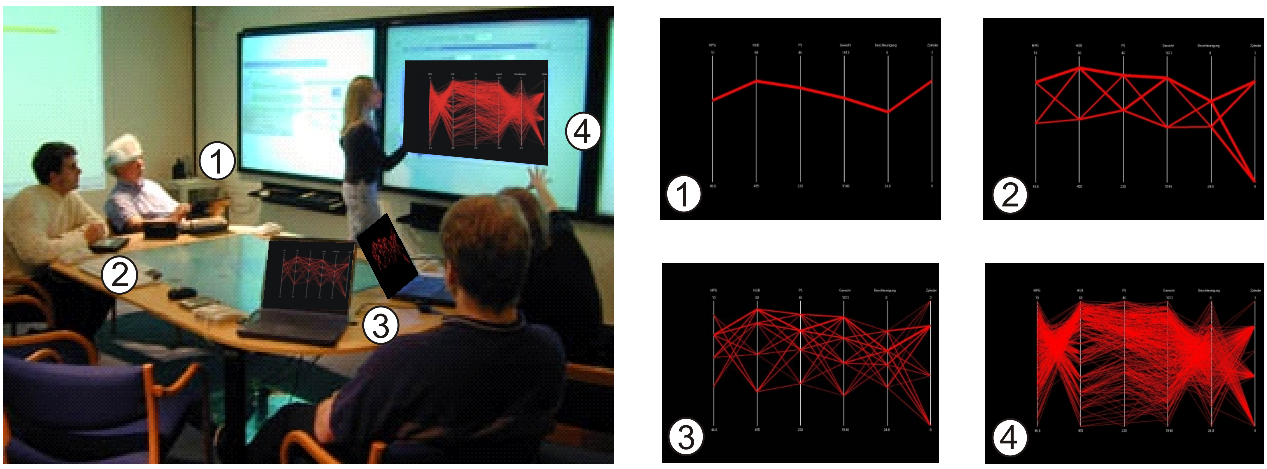

Figure 4. The effortless adaptation to different viewing devices is an inherent feature of progressive presentations and allows for further benefits in group work.

5. A FRAMEWORK FOR PROGRESSIVE INFORMATION PRESENTATION

This section describes a first attempt to throw some light on the technology required to achieve progressive refinement. We carefully analyzed existing approaches from different research fields and abstracted the applied ideas in order to make the general procedure more graspable. Consequently following the introduced applications, we focus on the presentation of information and thus, applied the traditional visualization pipeline as foundation of the proposed framework (Figure 5). However, progression may basically also be used for other kind of data and application fields.

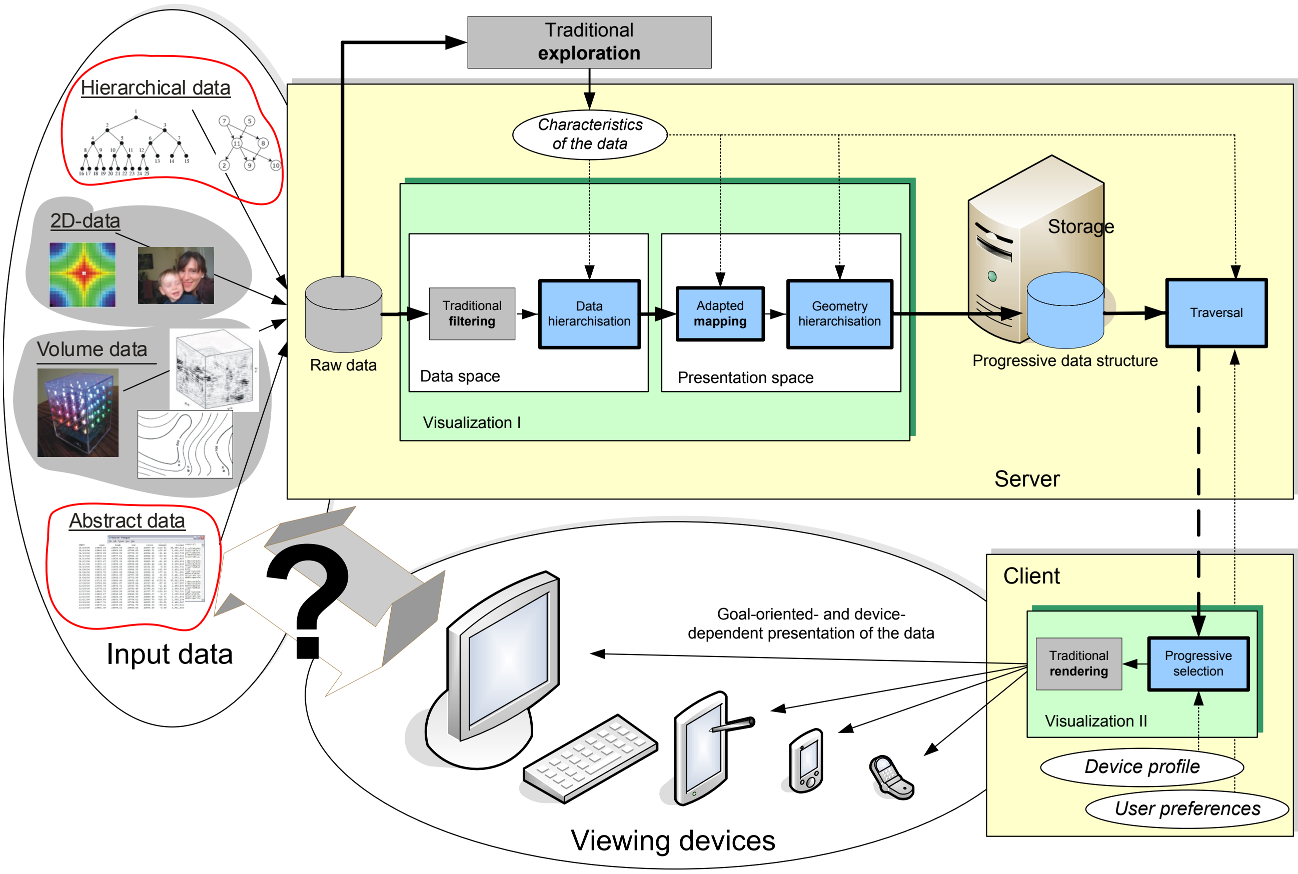

Figure 5. The problem of a goal-oriented presentation of huge data volumes on heterogenous viewing devices can be overcome by the proposed framework for progressive proceeding.

The framework supports different kind of *input data* passed to the system as *raw data*. Similar to the typical image-based approach (cf. Section 3), the framework is divided into *server* and *client*. Creation and *storage* of the *progressive data structure* appear on server-side to move most of the required computing power to the stationary component. *Traversal* accesses the data structure whenever content is to be served and generates the previews dependent on characteristics of the data or current *user preferences*. The client only adapts the received data to the requirements of the respective *viewing device*.

The traditional visualization stages *Filtering*, *Mapping*, and *Rendering* are distributed and split into two tiers.

Visualization I This tier creates the *progressive data structure* that consists of a hierarchy of graphical primitives used to form the visual data representation. First, the raw data is conditioned by completion or reduction during *filtering*. This component is part of the traditional pipeline and does not contribute to progression. The following stages, however, are crucial. The required LoD representation is achieved either by *Data hierarchisation* or *Geometry hierarchisation*.[34] In case of data hierarchisation the following

Mapping stage must often also be adapted in order to support the generation of meaningful geometry for the different abstraction levels. An external *traditional exploration* stage serves to detect *characteristics of the data* in order to support the identification of relevant presentation goals.

Visualization II This tier manages the progressive presentation on the respective viewing device. *Progressive selection* successively extracts the individual previews from the received data. By making use of a *device profile*, it is thereby checked whether the next preview can be displayed. In this case, the belonging data is requested from the server and passed to the traditional *Rendering* component. This component renders the preview and displays the result at the device.

The proposed framework illustrates the procedure of the progression approach schematically. The traditional (grey) and new components (blue) are logically divided to separate the associated procedures as much as possible. However, not all of the components are always required. To increase performance in practical implementations, components might also be combined to allow for faster integrated processing.

6. DISCUSSION

This section is dedicated to discuss the properties of the progression approach. We summarize its previously stated *advantages* and complete them by statements to technically oriented benefits. Furthermore, we will also discuss its *limitations* and still *open questions*.

Advantages The progressive approach is a new kind of presentation and offers many benefits regarding semantic as well as technical aspects. By providing different well-designed incremental previews to the data, an expressive, effective, and appropriate presentation[35] may be accomplished. As there are many options for a flexible parametrization, a progressive presentation can be aligned to the achievement of one or multiple visualization goals. The progressive approach supports and fulfills many of the principles and requirements stated in literature for successful information presentation. Due to its inherent support of scalability it also allows for uncomplex device adaptation.

Another more technically-oriented benefit of progressive refinement is its general resource-saving manner. By applying the approach to huge raster imagery, it could be accomplished[20] that the resource consumption does not need to depend on the original amount of data but only on the capabilities of the viewing device. Thus, for small size devices only a fraction of all available data must be transmitted and processed to provide an identic content representation. This important achievement may now also be available for other kinds of data. By assuming the client side to apply the mentioned mechanisms for response rate and clutter reduction or the viewer to stop the presentation as soon as the presentation goals have been achieved, there is a significant save in the resource consumption. Compared to original volume, only 0.34%, 1.17%, 4.27%, 6.40%, and 10.94% of the data must be transmitted and processed to accomplish the treemap previews shown in Figure 2. Considering the fact that in order to create a traditional treemap all data must be transmitted and processed, this achievement is of great importance for many application domains of this visualization technique. Similar results have been accomplished for the proposed progressive parallel coordinates (0.25%, 1.0%, and 1.75% for the previews (1, 2, 3) shown in Figure 4). As the presentation is created once and frequently reused, server load is negligible.

Beside significantly reduced response rates and avoided overloading, eligible data presentation on the respective device is crucial. With regard to the appearance of the data, most of the persons, who have been in touch with progressive presentations during development and testing, preferred overviews of the data as provided by the first presentation strategy for progressive PC (cf. to Figure 3). This might be mostly due to the yet not widely discovered additional advantages of the progressive display. With the broader availability of more sophisticated presentation strategies as shown in Figure 4 and increasing acceptance of the approach, this, however, might change in the future. Maybe also due to its applicability in ensembles with different cooperative viewing devices (e.g.*smart environments*, cf. to Figure 4/left). The problematic handling of heterogenous hardware and the multiple viewing options turn into a crucial advantage if a presentation is provided progressively. Whenever the presentation on the current device does not continue, the viewer is able to switch its view to another device in order to receive the remaining presentation stages. The fact that different devices stop at different presentation stages, also allows for comparison of previews and for extraction of further information from the refinement process. This is of beneficial value especially in group work.

Limitations There is no approach without shortcomings. The following paragraphs show the drawbacks of progressive refinement with respect to the consequences resulting from unrealized constraints. It can be generally stated that if many of its requirements (cf. to Section 3) can not be fulfilled, progressive refinement looses most of its benefits. Due to the novelty of the proposal, however, these statements might also serve as a starting point for the development of new ideas able to overcome the stated limits.

Goal-Orientation Its currently unclear, if all visualization goals can be supported by progression. This applies especially for all goals not requiring, avoiding, or inverting the overview-then-detail principle.

Scalability The application of progression is meaningless if the data is not hierarchical or can not be transferred into a scalable representation. Although introducing scalability into an arbitrary data set is highly complex and an open research topic, much efforts are made to solve this frequently occurring problem (*Visual Analytics*[36]). Thus, sophisticated solutions are just a matter of time and may be used to "upgrade" progressive visualization systems. If scalability can not be achieved, traditional techniques, as simple sub-sampling, are better options to accomplish at least less significant previews and resource savings.

Unambitious interpretation If the data or geometry abstractions, their LoD, and arrangement within the presentation sequence are not appropriately chosen, the presentation may lead to wrong conclusions. This is a general problem currently unclear how to be overcome. Leaving this problem to the presentation author seems currently to be the best option.

Random access The goal-orientated sequencing of the data requires its division in independent parts. This might not always be possible. In consequence, the available LoDs are limited, and the previews may not be created as required for an unambitious interpretation. Solutions may tightly combine hierarchisation and traversal to a single component and consider the respective visualization goal by applying a holistic strategy.

Encoding Abstractions are often a supplement to the original data and thus increase the data volume. This is not acceptable for massive data sets or systems strongly limited in storage space. Due to the fact that encoded data must also support the random access property, sophisticated compression schemes are required. Scalability, however, is a great starting point for efficient encoding.

Beside these limitations belonging to the presentation task, there are also drawbacks of the described strategy for device adaptation. Although, the proposed approach delivers still better results than existing technology, the applied measure of response rate and clutter require additional computing power. This might be an issue for client devices strongly limited in this resource. Furthermore, all other factors influencing a suitable presentation (e.g. the presentation goals) are not considered by the approach. This, however, is not a limitation of progression itself, but the proposed adaptation strategy.

Open questions The application of the progressive approach to enhance data presentation is novel. Although many former issues[11] are solved by proposals introduced in this contribution, there are still some questions to be answered in order to get a holistic understanding of its application in visualization. To achieve successful presentations, much work must be done in determining *Which visualization goals and techniques benefit most* from the proposed procedure and which do not. Although the advantages of progression are obvious, its implementation is not. *How can progressive refinement be realized for specific kinds of data and visualization systems?* Still open are also the various questions around the design of progressive presentations, e.g *Which metaphers/geometry/representations are suited? How much additional information can be conveyed by the previews? How strong may changes between previews be*, and *Do meaningful animations reduce the cognitive switches between the previews?*. Beside these questions, it must be generally evaluated if the approach can be suitably applied to dynamic instead of static data, to uncertain data, or in combination with other paradigms as parallel processing. The rather promising results received by our first works on this topic, however, strongly encourage to carry on in order to make progressive refinement a meaningful visualization approach.

7. CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

As shown in this contribution, progression is not only a means to overcome limited system resources. It is also a great means to enhance information presentation and device adaptation. By a pre-defined or interactive *tour-through-the-data* designed to convey relevant aspects of the data, the viewer is successively guided by multiple incremental previews. This procedure supports the overview-then-detail principle and requires a single view only. The fact that progressive presentations are inherently scalable, significantly simplifies the adaptation to the available system resources. This allows for short response rates and reduces the negative effects of visual clutter. Due to this and the fact that the general approach satisfies most of the requirements for successful data visualization, it seams that progressive refinement is of beneficial value for this application domain.

In future work, we will focus on the stated open questions and the summarization of the achieved answers in guidelines. To quantify the stated benefits for information presentation, we will also conduct meaningful usability tests. First feedbacks also encouraged to apply the feature not only to convey aspects of the data, but also to describe the principles of a visualization technique. If the general foundations for progression are laid, its combination with a much more flexible model-based adaptation to the viewing device would lead to even more powerful technology in heterogenous environments.

REFERENCES

[1] Baudisch, Patrick, G. N. B. V. and P., S., "Keeping things in context: a comparative evaluation of focus plus context screens," in [*Proceedings of CHI02*], (2002).

[2] Gutwin, C. and Fedak, C., "Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques," in [*Proceedings of the 2004 conference on Graphics interface*], 145–152, Canadian Human-Computer Communications Society (2004).

[3] Xie, X., Liu, H., Goumaz, S., and Ma, W.-Y., "Learning user interest for image browsing on small-form-factor devices," in [*Proceedings of the SIGCHI conference on Human factors in computing systems*], 671–680, ACM Press, New York, NY, USA (2005).

[4] Everitt, K. and Yee, K.-P., "Phoebus: Progressive display of partial query results," in [*Research reports of EECS Computer Science Division, University of California, Berkeley*], (2002).

[5] Pascucci, V., Laney, D. E., Frank, R. J., Scorzelli, G., Linsen, L., Hamann, B., and Gygi, F., "Real-time monitoring of large scientific simulations," in [*Proceedings of ACM Symposium on Applied Computing (SAC 2003)*], (2003).

[6] Schwenke, D., Vetro, A., and Hata, T., "Server-driven progressive image transmission of jpeg 2000," in [*Proceedings of Electronic Imaging - Image and Video Communications and Processing 2008*], (January 2008).

[7] Lohscheller, H., "A subjectively adapted image communication system," *IEEE Transactions on Communications* **32**, 1316–1322 (December 1984).

[8] Dürst, M. and Kunii, T., "Progressive transmission increasing both spatial and gray scale resolution," in [*Proceedings of the International Conference on Multimedia Information Systems*], 175–186, McGraw-Hill, Singapore (1991).

[9] Hoppe, H., "Progressive meshes," *Proceedings of ACM SIGGRAPH* **30**(Annual Conference Series), 99–108 (1996).

[10] Lee, H., Desbrun, M., and Schröder, P., "Progressive encoding of complex isosurfaces," in [*SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*], 471–476, ACM Press, New York, NY, USA (2003).

[11] Rosenbaum, R. and Schumann, H., "Chances and limits of progression in visualization," in [*Proceedings of SimVis2007*], (March 2007).

[12] Jerding, D. F. and Stasko, J. T., "The information mural: A technique for displaying and navigating large information spaces," *IEEE Transactions on Visualization and Computer Graphics* **4**(3), 257–271 (1998).

[13] Shneiderman, B., "The eyes have it: A task by data type taxonomy for information visualizations," *Proceedings of the IEEE Symposium on Visual Languages* , 336–343 (1996).

[14] Seffah, A., Forbrig, P., and Javahery, H., "Multi-devices "multiple" user interfaces: development models and research opportunities," *Journal of Systems and Software* **73**(2), 287–300 (2004).

[15] Seffah, A. and Javahery, H., [*Multiple User Interfaces: Cross-platform applications and context-aware interfaces*], Wiley&Sons, West Sussex, UK (2003).

[16] Mkovec, Z., *Spatial data adaptation for interaction in special environments*, PhD thesis, Czech Technical University in Prague (2007).

[17] Fuchs, G. A., Holst, M., Rosenbaum, R., and Schumann, H., "3D Mesh Exploration for Smart Visual Interfaces," in [*Proc. 10th Intl. Conference on Visual Information Systems (VISUAL'08)*], LNCS, Salerno, Italy (11-12 September 2008).

[18] "JPEG 2000 image coding system, part1," Final Publication Draft, ISO/IEC JTC 1/SC 29/WG 1 N2678 (19 July 2002).

[19] Rosenbaum, R. and Schumann, H., "Flexible, dynamic and compliant region of interest coding in JPEG2000," in [*Proc. IEEE ICIP2002*], (September 2002).

[20] Rosenbaum, R. and Schumann, H., "Jpeg2000-based image communication for modern browsing techniques," in [*Proceedings of Electronic Imaging - Image and Video Communications and Processing 2005*], (January 2005).

[21] Taubman, D. and Marcellin, M., [*JPEG2000: Image compression fundamentals, standards and practice*], Kluwer Academic Publishers, Boston (November 2001).

[22] Duchaineau, M., Porumbescu, S., Bertram, M., Hamann, B., and Joy, K., [*Hierarchical and Geometrical Methods in Scientific Visualization*], ch. Dataflow and remapping for wavelet compression and view-dependent optimization of billion-triangle isosurfaces, 1–17, Springer-Verlag, Heidelberg, Germany (2003).

[23] Sips, M., Schneidewind, J., Keim, D. A., and Schumann, H., "Scalable pixel-based visual interfaces: Challenges and solutions," in [*IV '06: Proceedings of the conference on Information Visualization*], 32–38, IEEE Computer Society, Washington, DC, USA (2006).

[24] Hao, M., Keim, D., Dayal, U., and Schreck, T., "Multi-resolution techniques for visual exploration of large time-series data," in [*Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization*], (23 – 25 May 2007).

[25] Tversky, B., Morrison, J. B., and Betrancourt, M., "Animation: can it facilitate?," *Int. J. Hum.-Comput. Stud.* **57**(4), 247–262 (2002).

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

[27] Pulo, K., "Navani: Navigating large-scale visualisations with animated transitions," in [*IV '07: Proceedings of the 11th International Conference Information Visualization*], 271–276, IEEE Computer Society, Washington, DC, USA (2007).

[28] Bederson, B. B. and Boltman, A., "Does animation help users build mental maps of spatial information?," in [*INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*], 28, IEEE Computer Society, Washington, DC, USA (1999).

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

[30] Shneiderman, B., "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Trans. Graph.* **11**(1), 92–99 (1992).

[31] Holst, M. and Schumann, H., "Efficient rendering of high-detailed objects using a reduced multi-resolution hierarchy," in [*Proceedings of GRAPP 2006*], 3–10 (February 2006).

[32] Inselberg, A., "N-dimensional coordinates," in [*Proc. of IEEE Pattern Analysis and Machine Intelligence, Picture Data Description and Management, Asilomar*], (1980).

[33] Fua, Y.-H., Ward, M. O., and Rundensteiner, E. A., "Hierarchical parallel coordinates for exploration of large datasets," in [*Proceedings of the 10th IEEE Visualization 1999 Conference*], IEEE Computer Society, Washington, DC, USA (1999).

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

[35] Schumann, H. and Mller, W., [*Visualisierung - Grundlagen und allgemeine Methoden*], Springer Verlag, Heidelberg (2000).

[36] Thomas, J. and Cook, K., [*Illuminating the Path: The Research and Development Agenda for Visual Analytics*], National Visualization and Analytics Center (2005).