On-the-fly device adaptation using progressive contents

Conrad Thiede, Heidrun Schumann, René Rosenbaum*

University of Rostock, Germany
University of Rostock, Germany
University of California Davis, U.S.A.

Abstract. In this publication we propose a device adaptation approach based on progressive contents. Such representations are inherently scalable, created once, and multiply used for different kinds of device. Computationally inexpensive on-the-fly adaptation is achieved by a preview-wise progressive data refinement with fully client-based resource assessment and estimation continuously predicting whether provided system resources will be exceeded. Our approach considers resource consumption in screen space, computing power, and during transmission of the required data. Due to its flexible and uncomplex manner, it is a much more general solution to content and device adaptation problem compared to related approaches. This is underlined by empirical results we got from first experiments and a typical use case.

Key words: device adaptation, progressive refinement, resource assessment, smart environments

1 Introduction

The constantly increasing volume of contents to be processed and displayed are one of the main challenges in modern computing. The sheer amount of data results in long response times, heavily overloaded displays, and thus, unacceptable content presentation. While this already affects most stationary hardware, the problem is much worse in smart environments [17] characterized by a heterogeneous device scenery. These problems are not new and different approaches to overcome them have been proposed [1]. However, usually only single aspects of the associated issues are addressed, a limited range of output devices is considered, or the solutions lack of flexibility [8]. Thus, new and more general ideas and approaches are required.

Progressive transmission and refinement is a strategy to overcome resource limitations and has a long tradition in image and multimedia communication [15]. By providing the viewer with continuous previews and thus intermediate feedback, a highly responsive viewing and browsing experience can be achieved. It

* Thanks to Deutsche Forschungsgemeinschaft (DFG) for funding parts of this research.

has been shown, that this feature can also be used to prioritize and highlight important contents [12] and may be applied to various kinds of data [7].

In this publication we propose a novel approach for device adaptation of graphical contents based on the beneficial properties of *progression*. Progressive contents are inherently scalable and thus allow for flexible adaptation to heterogeneous devices using a single "*multi-purpose*" data-stream only. The main idea of the introduced approach is to refine the contents as long as the respective device is able to provide the resources required for its appropriate transmission, decoding, and display. Resource estimation based on the assessment of past consumption is applied to predict if the data belonging to the next progression stage can still be transmitted and handled without violating predefined user demands. If the required resources can no longer be provided, the refinement is stopped and the best content representation for the respective output device is considered to be found. This process is computed on-the-fly at client side and requires no prior knowledge to data or device. We assessed the approach for raster imagery and geometry resulting from the hierarchical treemap display and achieved a highly adapted content representation.

To show the novelty of our proposal, Section 2 reviews the State of Art in related research. Section 3 is concerned with the fundamental concepts of progressive content presentation serving in Section 4 as the foundation to introduce our device adaptation approach. Section 5 discusses the achieved results followed by a typical use case (Section 6). Conclusions and directions for future work follow in Section 7.

Fig. 1. Example and application of a *progressive treemap* providing a tour-through-the-data and uncomplex adaptation to a variety of viewing devices.

2 Related work

Smart environments are not limited by presenting visual contents at certain orders or at specific output devices. Thus, the diverse range and strongly varying properties of possible configurations as well as the potential change of a particular output device require means for flexible adaptation.

Most of the adaptation approaches proposed in literature take only into account the properties of the data and the visualization goal [5]. Although of exceptional importance, device characteristics are usually neglected due to the sheer complexity of the problem if multiple of the influencing factors are considered. Related solutions are usually concerned with single aspects only [1]. The proposal by Mao [9] focuses on the communication channel between two devices. Thereby, it is distinguished in bandwidth, minimal bandwidth, and latency. The paper concludes that it is of exceptional importance to consider the amount of data during adaptation. In [6] the authors state that the viewing properties of the output device as screen size and resolution are important factors to also take into account.

As it is common sense that regarding the usability of interactive viewing devices, fast response rates are of higher importance than quality. Thus, processing speed must be considered as a crucial factor for device adaptation. It can be exactly assessed at client-side and only approximated at server-side. However, most of recently published adaptation strategies considering processing speed are server-based. A related approach focusing on the adaptation of tree-dimensional content has been introduced by Kim et al. [8]. Here, content customization is based on a generic schema of the MPEG-21 framework. The approach relies on a-priori knowledge to the output device, but does not require any modification at client side. Although this is a useful strategy for smart environments, the approach can not handle interactive changes and an individual content representation for each viewing device is required.

To summarize, several approaches are concerned with the adaptation of visual contents to heterogeneous output devices. Most of them, consider semantical aspects to adapt the visual representation only and neglect other main characteristics of the output device. Due to the complexity of the problem, a meaningful solution is still an open research question.

3 Progressive content presentation

Progressive processing became popular in the early days of the World Wide Web, when limited bandwidth was a big issue. To shorten the long latency times during the loading of imagery, the proposal of refining contents was a real relief. The basic idea of progressive imagery is to organize the encoded data in such a way that the decoding of a truncated stream leads to a restored image with less detail [15]. Thus, content previews can be given during a still running transmission, and with little data received, first conclusions can already be drawn. Due to its success, the approach has later also been applied to other kind of data [7, 11] or to overcome other resource limitations [10].

Although technical benefits have always been in focus of research, progression may also be applied to support semantical aspects of content display. One of the applied concepts are Regions of Interest (RoIs) [12] that allow for demand-driven previews by placing most important data at the beginning of the stream. As the resulting presentation sequence allows for an incremental buildup of knowledge, it can be a valid means to enhance the conveyance of data characteristics (cf.Figure 1) [11, 12]. Such a static or even interactive *tour-through-the-data* supports well-accepted visualization principles as the Information seeking mantra [13] – Overview first, zoom and filter, then details-on-demand – and is able to provide uncluttered views even for small devices and large amounts of data.

All progressive contents are based on an inherent hierarchical structure that allows for the data abstractions required to provide previews. This structure is key for the scalability feature and its flexible "multi-purpose" application. By implementing the paradigm - *Compress Once: Decompress Many Ways* [14, p.410], progressive contents are assumed to be created once, but to used multiple times. This is possible by transmitting, processing, and displaying the respective data in different traversal orders of the hierarchy and requires no further processing if the stored data is appropriately compressed and allows for random access [12].

While examples for a successful application of progressive imagery can widely be found in literature and practice, progressive geometry is not very common. An example demonstrating the mentioned technical and semantical benefits is the *progressive treemap* [11] illustrated in Figure 1. Treemap is a widely accepted technique to visualize hierarchical data [16]. Each node of the hierarchy is represented by a single rectangle. These rectangles are shown as nested sets in a way that a certain node spans all the nodes of its subtrees. The progressive treemap refines the hierarchy top-down, whereby the resulting presentation sequence can be interactively influenced by user-defined RoIs.

4 On-the-fly device adaptation

Viewer of large visual contents have often no clues if the data to be displayed exceeds the resources available at the output device. If so, the user is usually confronted with heavily delayed interaction feedback or overloaded displays. To overcome this, we introduce a strategy that takes advantage of progression to adapt the contents to the respectively available resources.

4.1 The adaptation procedure

In current communication systems supporting progressive content refinement (see Figure 2 and [11]), the data is assessed as *original content*. *Encoding* (hierarchisation, compression) and permanent *Storage* of the resulting *Progressive content* occur on *Server*-side. *Traversal* accesses the transformed content whenever it is requested and transmits the different previews to client side. The *Client* consists of a single *Decoding and Adaptation* unit. This tier manages the progressive content display and is the only processing stage that has to be extended to

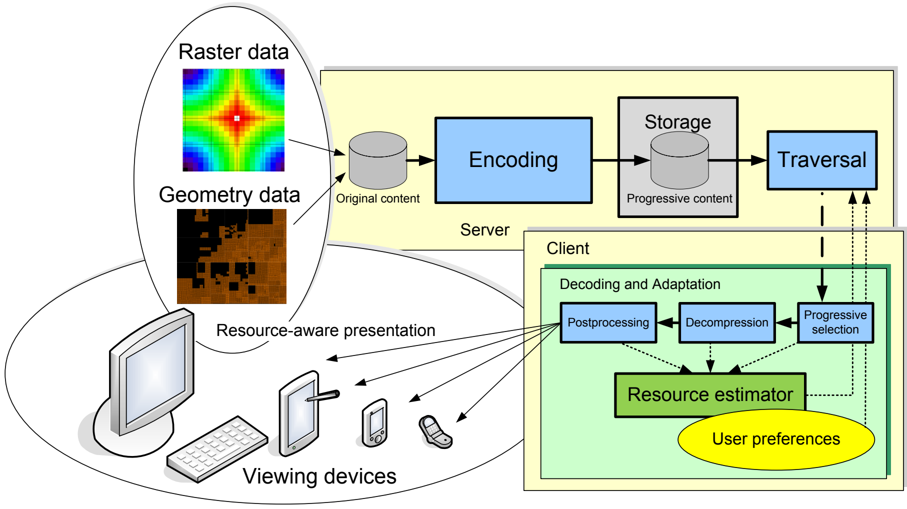

Fig. 2. A common processing pipeline for progressive data handling extended by components that allow for on-the-fly adaptation to heterogenous viewing devices.

allow for device adaptation. *Progressive selection* successively extracts the still encoded individual previews from the received data. The belonging data is then passed to a *Decompression* and *Postprocessing* component and displayed.

The main idea of our adaptation approach is to make use of the scalable coding and incremental presentation of progressive contents. During presentation the contents are preview-wise refined as long as the display device provides the resources required to transmit, process, and show the belonging data appropriately. The inherent overview-then-detail principle thereby nicely corresponds to the increasing resource consumption during presentation – first previews require little, detailed previews much resources. A *Resource estimator* decides whether the next preview can be displayed. This leads to either the next data request or signalization of the end of the adaptation procedure. The *Resource estimator* consists of the following 3 procedures:

1. Assessment of resource consumption of current preview in order to gain knowledge about the capabilities of the system (Section 4.2)

2. Estimation of future resource consumption in order to be able to predict system behavior for next preview (Section 4.3)

3. Completion of adaptation if estimated resource consumption exceeds predefined system criteria determined by *User preferences*.

User preferences describe the desired system behavior and might be different for every viewer. To allow for flexible adaptation, we consider the following common properties: (1) maximal *response time* as an important usability aspect, (2) maximal *data volume* to include limits in transmission time or costs, and (3) maximal *visual distortion* to consider quality reduction imposed by scaling large contents to small screen property. These attributes may be interactively specified or predefined based on prior empirically findings. *User preferences* may also be used to influence the generation of previews [11].

If the refinement process is completed, the most appropriate content presentation for the respective device is considered to be found.

4.2 Assessment of resource consumption

In our approach resource consumption is continuously assessed in order not to violate the specified user preferences. The following list indicates the resources that are related to the desired system behavior and shows how they can be assessed for scalably-coded imagery and treemap geometry.

Response time This demand strongly corresponds to the processing power provided by the system. To assess processing power for image or geometry data we propose to measure the time required for a full decoding and display of a certain preview at client side. The capabilities of the device are considered to be used up if the measured processing time exceeds the desired response time.

Data volume This demand corresponds to the use of the communication channel of the system. It can simply be assessed by measuring and summing the volume of the incremental image data bins or geometrical primitives that are received at client side. In contrast to others, this system resource can be very accurately measured. The calculated data volume must be smaller than the desired maximal data volume.

Visual distortion Visual distortion is caused by limited display property and can be assessed by measuring visual clutter. This involves many semantical factors, varies dependent on the kind of data and its representation, and thus is difficult to quantify in a rigorous manner.

Visual distortion can be seen as a measure how well the content representation matches the available display space. Due to the continuous detail enhancement during progressive refinement, the match steadily increases towards a certain threshold and decreases if it is exceeded. This threshold is considered to be the best match between representation and display space.

As for raster imagery all pixels are arranged on a uniform grid, there is no traditional cluttering. However, displaying an image on screen resolutions that are smaller than the image resolution leads to multiple image pixels for each screen pixel. Although often not negatively influencing data representation, it results in increased response times and data volumes and should therefore be avoided. Thus, it is meaningful to chose the screen resolution of the device as the matching threshold.

Assessing clutter for geometry is difficult and varies strongly dependent on the respective context [3, 4]. Due to the fact that primitives of the considered treemap display do not overlap, it is not subject to traditional clutter. However, some primitives may become very small and indistinguishable and it is meaningful to avoid their further refinement. As a generally valid clutter threshold is difficult to state, we apply a value that has been empirically determined for the used treemap implementation.

4.3 Estimation of future resource consumption

Prediction of the future resource consumption is introduced to avoid the needless transfer of data that leads to a violation of the desired system behavior. Here, we take advantage of the fact that progressive contents usually imply a specific characteristic in data distribution. However, we are aware of the fact that exact resource estimation is difficult if not even impossible.

Our prediction strategy is based on the assessment and evaluation of the resource consumption that was required for previous previews. Dependent on the ability to apply static or dynamically calculated values to estimated future consumption, the proposed strategies are labeled *static* and *dynamic estimation*.

Static estimation Due to the hierarchical structure of the underlying image or geometry data the data volume and processing time required to show two subsequent previews are often not arbitrary. Especially for scalable imagery usually encoded with defined decomposition schemes the hierarchy is a-priori known and thus can be used for prediction. A prime example is the Discrete Wavelet Transform (DWT) [14], where the number of reconstructed pixels increases by factor 4 with every successive level. For such schemes resource consumption of the next preview can be estimated by multiplying the respective factor to the requirements that were assessed for the current preview.

Dynamic estimation The structure of arbitrary hierarchical geometry is not required to provide a linearly increasing number of primitives with every successive level. Thus, static prediction is only of use if hierarchies are balanced and there is a-priori knowledge to the respective increase. To be able to cope with arbitrary geometries, we propose dynamic estimation. The foundation of this strategy is the ability to detect changes in data structure and system properties by calculating differences in previously estimated and exactly assessed resource consumption. In order to flexibly adjust the estimation to recent changes, the increase factor may vary for each successive preview and thus is calculated right before the preview is to be requested. To be able to consider a range of former preview, we take advantage of average and median filters of different length. These filters have the beneficial advantage to aggregate and smooth the gained prediction values and thus allow for adaptation to tendencies and temporal changes. The respective length thereby determines how many previous refinement stages are considered for current prediction. As the respectively chosen filter length strongly influences the quality of the estimation (cf. to results section 5), we also propose to adjust the length dependent on the quality of previous predictions. Filters of long length increase the performance of predictions for preview sequences that can be well estimated, filters of short length for refinements with strongly varying prediction quality. We propose to start with a short length filter and to increase its length if estimation is sufficiently accurate. If estimation quality decreases, the filter length should be reduced.

4.4 Adaptation in interactive systems

Modern communication systems usually allow for an interactive steering of data transmission and processing. This leads to individually designed previews and in turn to an unpredictable resource consumption that might cause the proposed estimation strategies to fail.

There are two different options to adjust the adaptation procedure to the properties of interactive systems: (1) extending the fully client-based approach to a client-server based strategy or (2) performing adaptation without resource estimation. The basic idea of the first proposal is to force the traversal component at server-side to deliver "predictable" previews. As the prediction approach is known at server side, this can be easily implemented and might even lead to better adaptation results. However, the original fully client-based approach can be much easier migrated into existing systems. The second option omits the resource estimation stage and thus might require the transfer and processing of data that exceeds resource limitations. Although a still ongoing data transfer or processing can always be stopped whenever this is detected, the used resources are spent without contributing to better adaptation.

5 Results

In this section, we provide results we got from first experiments using the introduced adaptation strategy applied to raster imagery and hierarchical geometry. The strategy is based on resource assessment and prediction. As the used assessment methods are not novel, we focus in the following on prediction. To demonstrate the wide applicability, we tested our proposals with different hardware and achieved similar results. The results stated below are gained on a modern workstation. The ordinary raster image used in the experiments is of size 3452×2300 and was scalably encoded using a 7-level DWT. For creating treemap geometry we used a data set that was provided for the 2003 *InfoVis contest* on the display of hierarchical data (*logs_A_03-01-01.xml*, 11 levels, 76551 nodes, strongly unbalanced). Thus, the following results cover the performance we got from simple and rather challenging test data.

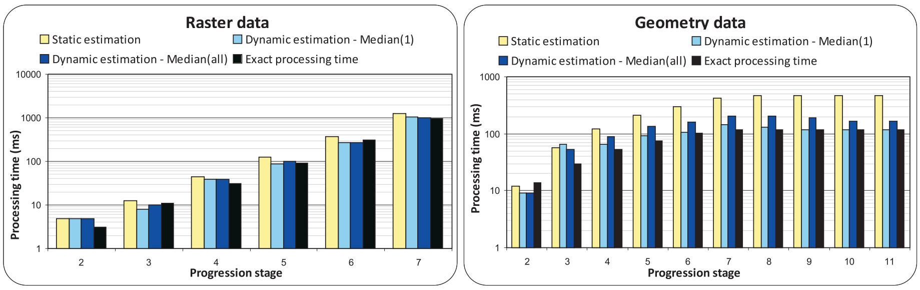

Fig. 3. Comparison of performance of the introduced estimation approaches and their parameters for image (left) and geometry data (right).

5.1 Estimating response time and amount of data

To show the applicability of our approach and to gain new insight for its appropriate parameterization, we assessed estimation accuracy for the introduced strategies. To achieve this, the time required to process the data up to a certain progression level was measured. We empirically determined a linear dependency between response time and amount of data (average difference: 6.7%). As the resulting statements are similar, they are shown for response time only. Figure 3 illustrate the achieved results for raster (left) and geometry data (right).

For image data it can be stated that prediction based on the Median(all) filter taking into account all currently available data performs best. It achieves the smallest average (avg: 18%) and smallest maximal prediction error (max: 54%). This is mainly due to the steadily increasing data amount and the dynamic nature of the prediction. Although, static prediction using a meaningful increase rate of factor 4 is able to produce the smallest minimal error (min: 1%), it estimates worse in other progression stages (avg: 27%). The Median(1) filter covers only the last progression stage and performs on average better (avg: 23%) than static prediction. We also tested the performance of average filters. Due to sensitivity to outliers, the estimation was in most cases worse then using Median filters. Outliers appear especially in time assessments at early progression stages. This is mainly due to the fact that the applied software-based time measurement fails if increments are very small. This, however, does not limit the applicability of our approach.

Contrary to well-structured image data, estimating response time for the given geometry was difficult. Strong variances from one preview to another and the property that the increment in data volume even decreases in later progression levels (cf. Figure 3/right) caused the static prediction to fail. Here, the Median(1) filter performed best (avg: 25%, min: 2%, max: 117%) as its short filter length adapts quickly to the changing structure of the data. The Median(all) filter has a longer coverage and thus provides slower adaptation (avg: 63%, min: 40%, max: 81%).

To consider real-world applications characterized by varying system resources we also assessed the performance for strong changes in CPU load (5-95%). However, this seams not to influence the estimation as strong as expected. Best filter for imagery is again Median(all) (avg: 20%, min: 2%, max: 38%). For geometry, both median filters show similar performance (Median(1) - avg: 160%, min: 30%, max: 220% / Median(all) - avg: 120%, min: 6%, max: 310%). Worst performs static prediction (avg: 210%, min: 14%, max: 570%).

The achieved results demonstrate that exact estimation of the consumed computing power without prior knowledge to the data is difficult. The prediction often varies by 20-30% and thus gives just a rough estimation of actual resource consumption. However, if applied to our adaptation approach the performance is often sufficient. Only if the user places its thresholds within the error range, the refinement will terminate too early or late. The average prediction error is a good indicator for the probability for such unintended behavior. In applications where high adaptation accuracy is crucial, we suggest to apply one of the adjustments proposed for interactive systems (cf. Section 4.4).

5.2 Estimating visual distortion

The performance of the introduced approaches for distortion estimation are different for images and geometry. Due to the options for an accurate assessment of screen and image resolution, distortion prediction for image data is exact. However, if one of the inherent image resolutions does not match with the screen resolution the adapted resolution will be smaller than required for the device. In this case, we propose to check if the current or next progression level matches the screen resolution closer and to use the better solution. As already stated is it currently unsolved how to assess and estimate distortion for geometry in a rigorous manner. However, the assessment we used in our experiments is computationally inexpensive and a valid means for adaptation of treemap geometry.

6 A typical use case for our approach

The proposed strategies are valid solutions for multiple problems in smart environments. The discussed use case is a smart meeting room [2] consisting of many different display devices. Meetings are typical events where images and data visualizations are shown and discussed. As the treemap is a typical information display with a broad applicability and acceptance [16], it may also be applied in this scenario.

The traditional procedure for displaying contents is its static representation on the viewing device that provides the most resources, e.g. a high-performance workstation connected to a high-resolution projector. This allows for fast system response and little visual distortion, but has the significant drawback that just a single view to the content is shown on a single display. If the same presentation is displayed on a strongly limited device, content representation and interaction are poor.

By applying the introduced adaptation approach, the available resources may be used much more efficiently. In group work it is often useful to provide the respective contents on many viewing devices. We refer to this case as "*parallel refinement*". Presentation adapted by the proposed strategy are highly responsive, non distorted, and interactive for all shown refinement stages and viewing devices. The semantic benefits of progressive refinement further allow for an improved conveyance of the presented information by different views to the data. Whenever the final presentation on a certain device is reached, the user is able to switch to another device in order to see the remaining presentation stages. The fact that different devices stop at different stages further allows for later comparison of the different previews and for extraction of further information from those contents not part of a particular representation or hidden by clutter.

Information has often to be viewed privately, e.g. on a Smartphone, PDA, or netbook, before to be shown to the public. The proposed approach refines and presents the contents, maybe under interactive modification, until the resource limits of the device are reached. By moving the content to other viewing devices providing more resources, the viewer is able to share his insights whenever desired. We refer to this case as "*sequential refinement*".

Compared to the traditional content presentation using a single view and display, the introduced approach allows for much more flexibility. Single centrally stored content is shown in many incremental views and is adapted to all desired viewing devices. To achieve this, the approach requires no a-priori knowledge to the data, is low complex and completely accomplished at client side.

7 Conclusions

The diverse range of heterogeneous viewing devices requires strategies for flexible content adaptation. Existing solutions usually focus on semantical aspects and provide poor system performance. We proposed an approach that can be applied to most kinds of devices and considers response rate, data volume, and visual distortion. Our adaptation strategy is based on progressive contents and on-the-fly resource assessment and prediction. Device adaptation is achieved by completing refinement whenever it is predicted that handling the next progression stage violates desired system behavior. We achieved the best results by dynamic estimation using a median filter (average estimation error: 18-25%) that covers all (images) or just the last progression stage (geometry). A rather beneficial property of our approach is that progressive contents are encoded just once and can be multiply used for different devices. This solution is computationally inexpensive, overcomes many of the problems stated in literature for device specific content adaptation, and has a wide variety of applications.

In future work we will implement this approach for other kinds of data and application fields. Furthermore, we plan to compare its performance to a profile-based approach founded on prior knowledge to the different influencing factors. However, our next step will be concerned with the development of more accurate predictions schemes specifically regarding visual distortion.

References

1. R. Bandelloni, S. Berti, F. Paternò, Mixed-Initiative, Trans-Modal Interface Migration Proceedings Mobile HCI, Springer, 216-227 (2004)
2. C. Burghardt, C. Reisse, T. Heider, M. Giersich, T. Kirste, Implementing scenarios in a Smart Learning Environment, Proceedings of 4th IEEE International Workshop on Pervasive Learning (2008)
3. G. Ellis, A. Dix, The plot, the clutter, the sampling and its lens: occlusion measures for automatic clutter reduction, AVI '06: Proceedings of the working conference on Advanced visual interfaces, ACM, 266-269 (2006)
4. G. Ellis, A. Dix, A Taxonomy of Clutter Reduction for Information Visualisation, IEEE Transactions on Visualization and Computer Graphics, IEEE Educational Activities Department, 13, 1216-1223 (2007)
5. I. Fujishiro, R. Furuhata, Y. Ichikawa, Y. Takeshima, GADGET/IV: A Taxonomic Approach to Semi-Automatic Design of Information Visualization Applications Using Modular Visualization Environment, InfoVis, 00, 77 (2000)
6. L. Skorin-Kapov, H. Komericki, M. Matijasevic, I. Pandzic, M. Mosmondor, MUVA: a Flexible Visualization Architecture for Multiple Client Platforms, Journal of Mobile Multimedia, 1, 03-17 (2005)
7. H. Lee, M. Desbrun, P. Schröder, Progressive encoding of complex isosurfaces, SIGGRAPH 2003, ACM Press, 471-476 (2003)
8. H. Kim, C. Joslin, T. Di Giacomo, S. Garchery, N. Magnenat-Thalmann, Device-based decision-making for adaptation of three-dimensional content, The Visual Computer, Springer, 22, 332-345 (2006)
9. S. Mao, D. Bushmitch, S. Narayanan, S. Panwar, MRTP: a multiflow realtime transport protocol for ad hoc networks, Vehicular Technology Conference, VTC 2003-Fall. 2003 IEEE 58th, 2003, 4 (2003)
10. V. Pascucci, D. E. Laney, R. J. Frank, G. Scorzelli, L. Linsen, B. Hamann, F. Gygi, Real-Time Monitoring of Large Scientific Simulations, Proceedings of ACM Symposium on Applied Computing (SAC 2003)
11. R. Rosenbaum, H. Schumann, Progressive refinement - more than a means to overcome limited bandwidth, SPIE - Electronic Imaging 2009, San Jose/US, January 18 - 22 (2009)
12. R. Rosenbaum, H. Schumann: Progressive imagery beyond a means to overcome limited bandwidth, SPIE - Electronic Imaging 2009, San Jose/US, January 18 - 22 (2009)
13. B. Shneiderman, 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. D. Taubman, M. Marcellin, JPEG2000: Image compression fundamentals, standards and practice, Kluwer Academic Publishers (2001)
15. D. Tian, G. Alregib, BaTex3: Bit Allocation for Progressive Transmission of Textured 3-D Models, IEEE Trans. Circuits Syst. Video Techn., 18, 23-35 (2008)
16. M. Wattenberg, Visualizing the stock market, CHI '99: CHI '99 extended abstracts on Human factors in computing systems, ACM, 188-189 (1999)
17. M. Weiser, The computer for the 21st century, SIGMOBILE Mob. Comput. Commun. Rev., ACM, 3, 3-11 (1999)