Journal of Universal Computer Science, vol. 11, no. 11 (2005), 1766-1779
submitted: 1/9/05, accepted: 1/10/05, appeared: 28/11/05 © J.UCS

Scalable Visual Data Exploration of Large Data Sets via MultiResolution

Daniel A. Keim
(University of Konstanz, Germany
keim@inf.uni-konstanz.de)

Jörn Schneidewind
(University of Konstanz, Germany
schneide@inf.uni-konstanz.de)

Abstract: During the last decade Visual Exploration and Visual Data Mining techniques have proven to be of high value in exploratory data analysis since they combine human visual perception and recognition capabilities with the enormous storage capacity and the computational power of today's computer systems in order to detect patterns and trends in the data. But the ever increasing mass of information leads to new challenges on visualization techniques and concepts. Due to technological progress in computer power and storage capacity today's scientific and commercial applications are capable of generating, storing and processing massive amounts of data. Most existing visualization metaphors and concepts do not scale well on such large data sets as interaction capabilities and visual representations suffer from the massive number of data points. To bridge this gap, Visual Analytics aim to incorporate more intelligent means than to just retrieve and display the data items to filter the relevant from the non-relevant data. In this context the paper introduces a new approach based on a Multiresolution paradigm to increase the scalability of existing Visual data exploration techniques. The basic idea is to provide relevance driven compact representations of the underlying data set that present the data at different granularities. In the visualization step the available display space is then distributed according to the data granularity, to emphasize relevant information. The paper aims at introducing a technical base of Multiresolution visualization and provides an application example that shows the usefulness of the proposed approach.

Key Words: Visual Data Exploration, Visualization Technique, Multiresolution
Category: H.0, H.4

1 Introduction

Due to the progress in computer power and storage capacity over the last decade, today's scientific and commercial applications are capable of generating, storing and processing massive amounts of data. Examples are historical data sets including census data, transaction data from credit card-, telephone- and e-commerce companies[12] and financial data. Additionally there exist many dynamic processes, arising in business, network or telecommunication, which generate tremendous streams of time related or real time data like sensor data, web click streams or network traffic logs. The analysis of such massive data sets is an important and challenging task, since researchers and analysts are interested in patterns in the data including associations, correlations or exceptions. These information is needed in order to turn collected data into knowledge, e.g. to identify bottlenecks, critical states, fraud or any other interesting information hidden in the data. Visual analytics and Visualization techniques have been proven to be of great value in analyzing and exploring such large data sets, since presenting data in an interactive, graphical form often fosters new insights, encouraging the formation and validation of new hypotheses to the end of better problem solving and gaining deeper domain knowledge [11]. But the increasing size of today's datasets poses a challenge for developers of Visualization models and methods, since many Visualization approaches still rely on interaction and visualization approaches developed 10 years ago, and its doubtful that they are able to meet the demands of the increasing mass of information.

Eick and Karr[4] proposed an scalability analysis and came to the conclusion that many visualization metaphors do not scale effectively, even for moderately sized data sets. Scatterplots for example, one of the most useful graphical techniques for understanding relationships between two variables, can be overwhelmed by a few thousand points. Additionally, there are two limiting factors for all visualization techniques: human perception and display area. On one hand, human perception, that means the precision of the eye and the ability of the human mind to process visual patterns, limits the number of perceptible pixels and therefore affects visual scalability directly. One the other hand, monitor resolution affects on visual scalability through both physical size of displays and pixel resolution. At a normal monitor viewing distance, calculations in [20] suggest that approximately 6.5 million pixels might be perceivable for the human, given sufficient monitor resolution [4]. In typical application scenarios monitor resolution rather than human vision is the limiting factor.

Based on these facts, the analysis of large data sets reveals two major tasks. The first one is the question, how visualizations for massive data sets can be constructed without loosing important information even if the number of data points is to large to visualize each single data point at full detail. The second important task is to find techniques to efficiently navigate and query such large data sets. Therefore our approach considers a upper bound for the number of data objects that can be visualized in a single view. This limit can be given by the visualization methaphor, the available display space or as user parameter. In massive datasets this limit is typically much lower than the number of data items contained in the data set, resulting in visualization bottlenecks. Therefore our goal is to take the maximum number of presentable data points as initial limitation, and try then to produce a maximum valuable visual layout with respect to this boundary. Since it is almost impossible for the human mind to search and construct hypothesis about interesting pattern by reading multiple visualizations of portions of the data, our goal is to increase visual scalability by generating an initial compact representation of the whole datasets that avoids clutter and information overload and presents the most important information, from an analysts point of view, as a vantage point for the interactive visual exploration process. We realized this by providing different data granularities based on the importance value of portions of the data and employed a tree structure to manage the different levels of detail.

1.1 Classical Data Exploration

The classical exploration of small and medium data sets usually follows a three step process: *Overview first, zoom and filter, and then details-on-demand* , which has been called the Information Seeking Mantra [19]. That means that the data analyst first needs to get an overview of the data. There he may identify interesting patterns or exceptions in the data and focus on one or more of them. For analyzing the patterns, the data analyst needs to drill-down and access details of the data. In the age of massive data sets all three steps of the Information Seeking Mantra are difficult to realize. An overview visualization without losing any interesting pattern or subset is difficult to create, since the amount of pixels of modern data display systems to not keep step with the increasing flood of data. The plenty of information hidden in massive data sets make it very difficult for the human to understand the really interesting or relevant information. We believe, that for the analysis of massive data sets, it is not sufficient to just retrieve and display the data using a visual metaphor, it is rather necessary to support the analyst by filtering the underlying data by it's value of interest, but at the same time providing interaction models which still allow the user to get any detail of the data on demand. The data analyst should control the granularities of interesting patterns or subsets by providing a relevance function.

For example, consider a stock market data analyst. The analyst may want to observe any detail for the daily stock prices, but on the other hand he wants to see the global trend of stocks over weeks, months, years. It is very difficult for the data analyst to perceive interesting trends by having a visualization that shows every data item at the same detail level. On the other hand it is very difficult to compute effective visualization without losing important trends since the screen size is limited at the data analysts workbench. We believe that the exploration process is more effective by choosing different relevance levels for the different time periods like minute, day, week, month and year. That means the daily stock market data gets more space within the visualization and will be shown at full level. With decreasing relevance the details get coarser and at last only the global trend within years is shown.

2 Related work

If the number of data objects of large data sets exceed the monitor resolution, the user usually faces the serious problem how to process and visualize the data in a reasonable way. This problem is intensified in interactive applications, where typically the amount of data which can be processed interactively is much smaller than the data which can be shown on the display. To this end, several strategies have been derived to handle such vast amounts of data. Hierarchical structures for example allow fast data access and search operations and allow different data granularities. Adaptive methods like Correlation analysis, Information gain or Statistical methods enable the concentration on the interesting parts of the data based on users objectives, techniques like Sampling, Clustering or Aggregation help to reduce or compress the amount of relevant data objects.

Several classical visualization and interaction methods which, in an adapted or extended form, may be useful in the exploration of massive data sets have been proposed in the past. Focus+Context techniques for example, display information of particular interest in detail while the remaining data set is shown in a smaller representation relative to the areas of interest. Interesting approaches are fish-eye views [7] [18], Bifocal displays [13], and perspective walls [14]. Magic Lense filters [3] change the visualization of data items by applying a viewing operation on the data items. Table Lens [17] applies fish-eye techniques to table-oriented data sets.These techniques have proven to be very valuable in the exploration of small and medium sized data sets, which can be mapped to the data display at full scale. But for massive data sets we can observe an important issues: The area of interest, i.e. the number of relevant data objects, is typically larger than the data display space, therefore only parts of the data can be shown in a single view and focus and context gets lost.

An interesting approach that separates individual views of subsets from the global view is presented by Beard and Walker [2]. It uses navigational maps to show a miniature version of the entire data set. The Information Mural [9] is a interesting technique based on the key concept of reducing a two-dimensional representation of an entire information space that fits completely within the data display space. The approach allows the overplotting of data items and uses gray scales, individual colors and shading to indicate data items that occupy the same position within the data display space. This approach addresses the problem of the limited screen space, but at the same time creates the new problem of superposed data objects. Our approach is based on the assumption, that for massive data sets it is necessary to show the data items on different object resolutions based on relevance and available screen space. Several techniques for reducing data sets by providing coarse presentations have been proposed in the past. Typically these techniques construct a series of coarse data representations by employing tree structures or wavelets [5, 15].

We extend these approaches by adding relevance driven features in the aggregation and selection process and providing different data granularities even within single tree levels. The compact representation is presented in such a way that relevant regions or objects pop out and the analyst may then explore details by selection of this particular region, e.g. by drilling down to a finer data granularity. To provide these different data granularities and to manage the different levels of detail efficiently, a special tree structure is employed. The idea is similar to techniques like the SpaceTree [16], a tree-browser which adds dynamic rescaling of branches of the tree to best fit the available screen space. The effect that we achieve is similar to multiresolution approaches like HyperSlices [22] and Hierarchical Parallel Coordinates [6], a Multiresolution version of Parallel Coordinates [8], that uses multiple views at different levels of detail for representation of large data sets. Another interesting hierarchical multiresolution approach for dimension reduction was proposed in [23]. In [21] a multiresolution multidimensional brushing technique based on wavelets is presented.

3 Scalable Multiresolution Visualization

The techniques mentioned in the last section have shown that the scalability of Visualization techniques can be improved by Multi-resolution metaphors, since these techniques are able to compress the underlying data set and thus reduce occlusion and information overload. We adapt and extend this approach to present effective visualizations for large data sets. The basic idea of our MultiResolution approach is to compute a hierarchy of views that present the underlying data at different levels of detail. But in contrast to existing approaches which primarily visualize only high level views of aggregated data points, and the user is then forced to navigate through the hierarchy in order to explore regions of interest and detailed information, the goal of our method is to maximize the information content of the initial visualization. Therefore our technique not just generates a hierarchy containing different levels of detail, it rather shows important data objects at higher detail while presenting less important data objects at lower level of detail. This is achieved by employing analytical methods in form of objective functions in the hierarchy construction and visualization process. User interaction like drill down or zooming can be performed more efficiently, since potentially important regions are already emphasized in the initial visualization.

3.1 Basic Idea of MultiResolution

The basic idea of multi-resolution visualization is to decompose the data display space into local screen regions with individual object resolutions. These object resolutions control the granularity of the data points within each particular region. To provide and manage the different data granularities, a tree structure is

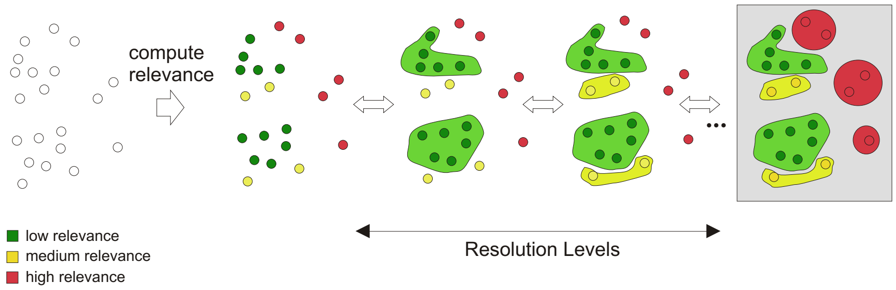

Figure 1: *Basic idea* – decomposition into local screen regions allows it to display an overview of the entire data set

employed. The structure of the tree highly depends on predefined analytical objective functions, which determine the relevance of single- or sets of datapoints. The goal is to provide an initial visual presentation of the whole data set with respect to the available screen space , that gives relevant parts of the data more space on the screen to present them at higher detail.

As an example, we consider a clustering method, which determines clusters in a given data set as shown in Figure 1. Without loss of generality we assume that the data set contains 2-dimensional data. A simple way to visually detect clusters in the data are classical scatterplots. But if the number of data points is high, overlap in some portions of the display would it make really hard to detect clusters or to distinguish clusters from noise. Our approach instead determines an importance value for each data point . Suppose we use a clustering algorithm to detect clusters in the data. Let the objective be the detection of clusters which contain very few data points, which may indicate outliers in the data. Therefore our objective function gives a high importance value to data points which belong to clusters with a small number of data points (red and yellow color in figure 1), while data points belonging to large clusters get a low importance value (green color). Based on the importance values of the data points, a hierarchy is created, similar to single linkage clustering. In each step the data subsets with the lowest average importance value are merged, resulting in a tree with different levels of detail like shown in figure 2. To visualize the data set, data objects from the tree structure are selected so that the number and relevance of the selected objects is maximized, depending on the given display space. Each single data point must be included in exactly one selected object.

Figure 2 indicates a possible selection. The higher the importance value of each single object, the more screen space may be given to visualize the object. Note that the construction of the hierarchy is independent from the underlying

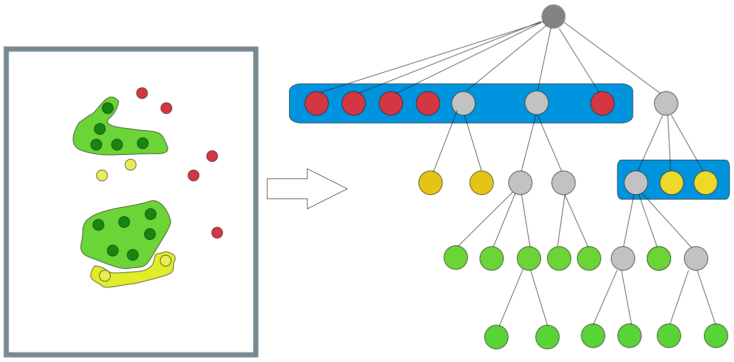

Figure 2: *Hierarchy of MultiResolution Objects (MRO)* - The relevance value of the MRO's decreases with increasing tree level, the blue marked nodes are used for visualization

visualization, and may therefore be combined with most visualization techniques to increase scalability. The next section describes our approach in detail and gives a more formal definition for the concepts of MultiResolution visualization paradigm.

3.2 Basic Concepts

Like described in the last section our approach consists of 3 steps: 1. Determine the relevance of each data point in the data set based on a given relevance function 2. Provide a compact hierarchical representation of the data with different levels of detail based on the relevance of data points 3. Select a given number of objects from the hierarchy so that the relevance is maximized and each data point is contained in at least one selected data object. The relevance function allows us to determine the object resolution of the local screen spaces. Of course the relevance function depends on the application scenario and must be determined by the user. Further research will focus on providing predefined relevance function for certain tasks like clustering or outlier detection. In general the relevance function is defined as:

Definition 1 *Relevance Function*
Let $A = \{a_0, \cdots, a_{N-1}\}$. *The relevance function* $\psi : A \longrightarrow \mathbf{N}$ *assigns every data point* $a_i \in A$ *a relevance value* $\psi(a_i)$.

The relevance function ψ depends on the application scenario and can be given by the user. The computation of the relevance function should be done on the fly. Based on the relevance of single data points we are now able to construct an hierarchical structure by merging or combining data points with lowest relevance values in order to get a compact representation of the data set. Therefore the Multi-Resolution Objects are defined as:

Definition 2 *Multi-Resolution Objects*
Let $A = \{a_0, \cdots, a_{N-1}\}$ *be the input data points and* $\Psi = \{\psi(a_0), \cdots, \psi(a_{N-1})\}$ *their associated relevance values. A MultiResolution object* MRO *is a set of locally close data points which have similar relevance values*

$$MRO = \{a_i \in A : \forall a_j \in MRO : |\psi(a_i) - \psi(a_j)| \leq \kappa \wedge d(a_i, a_j) \leq \epsilon\}$$

Within every multiresolution object we define a object resolution level l_i which is application dependent. We suggest to identify the object resolution level l_i as the average of the relevance of all multi-resolution object members. Application dependent other functions (e.g. min or max) may be used.

Definition 3 *Object Resolution*
Let $mro_i = \{\xi_0, \cdots, \xi_{N-1}\}$ *be a multiresolution object and* $\Psi = \{\psi(a_0), \cdots, \psi(a_{N-1})\}$ *the associated relevance values of the members of the multi-resolution object* mro_i*. The object resolution level* l_i *can be determined as:*

$$l_i = \frac{\sum_{i=0}^{N-1} \psi(a_i)}{N}$$

For a given number of maximum allowed multiresolution objects n_{max} , the MultiResolution object tree is then constructed similar to Single Linkage as follows:

Algorithm 1 Construction of MRO-Tree

INPUT: $\{a_0 = mro_0, ..., a_{max} = mro_{max}\}$, with $l(mro_i) = l_i$, $n_{max} > 1$
while Number of relevant MRO's $> n_{max}$ **do**
  Create new MRO mro_{new} by merging MRO's mro_i, mro_j with lowest relevance
  Set $l_{new} = f_l(l_i, l_j)$
  $mro_{new}.parent = ROOT$
  $mro_{new}.childs = (mro_i, mro_j)$
  Mark mro_{new} as relevant, mark mro_i, mro_j as not relevant
end while

3.3 Problem Definition

Once the MRO tree is constructed the question is how to map the relevant MRO objects to the display space, since the multiresolution mapping function f must satisfy some important visualization constraints. In the following we formally define this problem.

Let the data display space $DS \subseteq INT^2$ be defined as $DS = \{0, \ldots, x_{max}\} \times \{0, \ldots, y_{max}\}$, where x_{max} and y_{max} are the maximal extension of the display. Because the dataset A is assumed to be massive, the number of pixels is not sufficient to place all data points directly to the screen space $|DS| << |A|$.

Let $MRO = \{mro_1, \cdots, mro_m\}$ be a decomposition of the data space into a set of multi-resolution objects. The goal of the multi-resolution visualization is to determine an useful mapping function f of the MRO objects to the display space DS that must satisfy three visual exploration goals:

1. **Relevance Preservation** The most important constraint is that the relevance of the visualization must be as high as possible: Formally, this can be expressed as
$$\forall (mro_i, l_i) \subseteq DS : \sum_{i=0}^{m} l_i \longrightarrow max$$
2. **Minimal Decomposition**
$$\forall (mro_i, l_i) \subseteq DS : \sum_{i=0}^{m} |mro_i| \longrightarrow max, max \leq x_{max} \cdot y_{max}$$
3. **Spacefilling**
$$\forall (mro_k, l_i) \subseteq DS : \left| \bigcup_{k=1}^{m} mro_k \right| \longrightarrow max, , max \leq x_{max} \cdot y_{max}$$

The third constraint demands that the data display space should be occupied as much as possible. Since we do not want to loose any information, it is also clear that every data point must be contained in at least one MRO of the display space.

3.4 MultiResolution Optimization Problem

While it is not hard to find a good solution for any of the three constraints taken individually for small and medium data sets A, they are difficult to optimize simultaneously for visualizing large and massive data sets A. Therefore, our goal is to find a good spacefilling trade-off between Relevance Preservation and the Minimal Decomposition constraint, formal defined as optimisation problem:

$$\alpha \cdot \sum_{i=0}^{N-1} l_i + \beta \cdot \sum_{i=0}^{N-1} |mro_i| + \gamma \cdot \left| \bigcup_{k=1}^{m} mro_k \right| \longrightarrow max$$
$$\forall i = \{0, \ldots, N-1\} : f(a_i) = (mro_k, l_i) \wedge \alpha, \beta, \gamma \in \mathbf{R}$$

4 Experimental Results

This section presents an application example of our relevance driven Multiresolution approach. In our prototyp implementation we integrated our technique into the CircleView system [10]. We provide an data example from the area of stock market analysis, which shows that our Multiresolution approach has the potential for making visualization techniques scalable, reduces information overload and thus provides better visualizations.

4.1 MultiResolution Circle View

We integrated the Multiresolution approach into the CircleView technique in order to analyse a historical stock market dataset containing S&P 500 stock prices. The basic idea of the CircleView visualization technique is to display the distances for the attributes as segments of a circle, similar to the CircleSegments technique[1]. If the data consists of k dimensional attributes, the circle is partitioned into k segments, each representing the distances for one attribute. Inside the segments, the distance values belonging to one attribute are arranged from the center of the circle to the outside in a subsegment layout. The size of the segments and subsegments can either be predefined or parameter dependent. Additionally the size of each subsegment can vary from pixel to segment size. CircleView supports ordering and clustering of segments and subsegments, user interaction as well as nearest neighbour searches between segments or within single segments. Therefore it allows intuitive comparison between single segments and subsegments as well as the identification of trends and exceptions.

Figure 3 presents the basic idea of MultiResolution CircleView, showing the stock prices of 240 stocks from the S&P 500 over 6 months, starting from January 2004 (outside of the circle) to June 2004 (center of the circle). Each segment represents a single stock and each subsegment the average closing stock price of a certain time period depending on the detail level. The number of data values that can be visualized without aggregation using CircleView is limited by the circle area $C_{area} = 2 * \pi * radius$. In massive datasets this border can easily be reached. That means, if the radius of our circle would be less than 120 pixels, it is impossible to show all 120 stock prices without occlusion. To handle this case using Multiresolution, we use the fact that from an analysts point of view it may

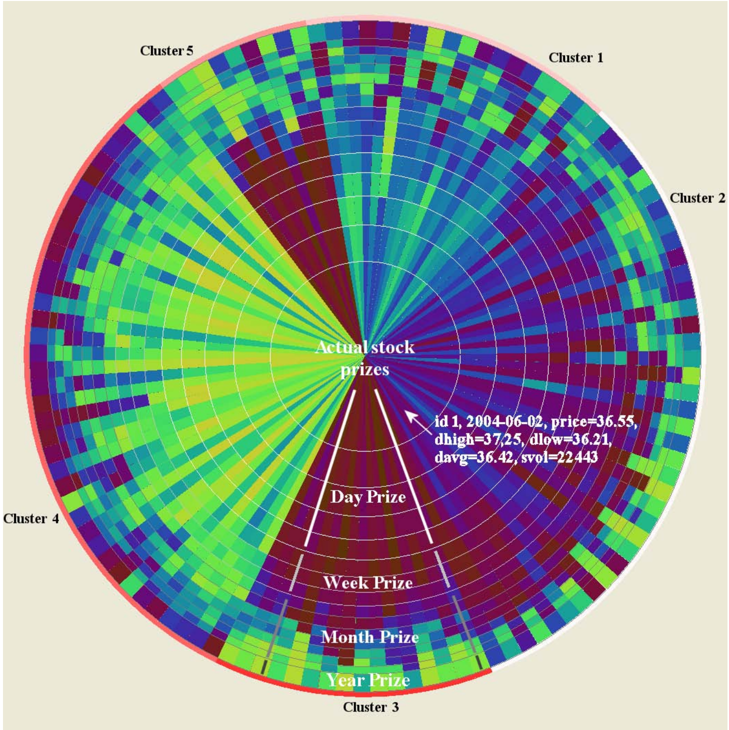

Figure 3: CircleView showing stock prices of 240 stocks from the S&P 500 over 6 months (120 business days). Each Segment represents a single stock and the subsegments represent stock prices. The most actual stock price are shown in the middle of the circle at full detail. For older stock prices the multiresolution approach presents average prices per week/month/year (outside of the circle).

be more interesting to analyse actual stock courses rather than historic courses. Therefore the basic idea is to show only actual stock prices at full detail and to present historic values as aggregated high level views. The relevance function is based on this assumption and therefore the relevance value of each data point (stock price) is determined by it's time stamp. The corresponding MRO Tree is shown in figure 5. In the presented example, the five latest day closing stock prices are shown at full detail by the innermost subsegments. The level of detail as well as the length of the subsegments decrease from the center to the outside of the circle. Historic stock prices are only presented as average values per week, per month or per year depending on the particular data as shown in figure 3. The older the data, the lower the level of detail. Therefore the user is able to access particular information on actual data instantly, e.g. by mouse interaction

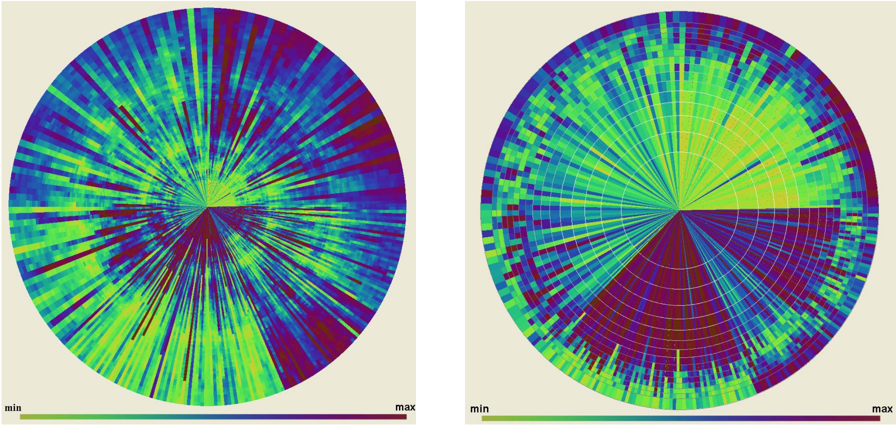

Figure 4: Comparison CircleView (left) vs. MultiResolution CircleView (right) showing 240 stocks from the S&P 500 over 6 months (120 business days). Due to the limited circle radius (screen space) only 100 days can be shown in the left figure, MultiResolution (right) allows instant access to the whole data set, but gives more importance to actual stock prices, shown by larger subsegments in the center of the circle. Clustering helps to identify similar groups of stocks.

and gets at the same time an overview of the whole dataset. Of course there is the possibility to perform drill down operations on items with lower resolution to get information on historic data on demand.

In Figure 4 the described data sample is visualized using k-means clustering with 5 clusters. Similar stocks are clearly revealed. It is easy to see that Multiresolution provides better results than standard CircleView in terms of revealing actual stock prices, but at the same time preserves the global trend. An analyst may easily select and explore the actual stock prices, which is a difficult task in standard CircleView presented figure 4 left. By selecting a week or month value, the user is able to drill down to the next lower resolution level day and week respectively. The fact that the level of detail depends on the importance of data values results in very flexible visualizations.

5 Conclusion and Future work

In this paper we presented an approach for relevance driven Multiresolution visualization of large datasets to increase the scalability of visualization techniques. The basic idea is to define a relevance for each data object with respect to a given relevance function, and to present highly relevant objects at full detail and less relevant objects at lower detail. We presented a formal definition of the

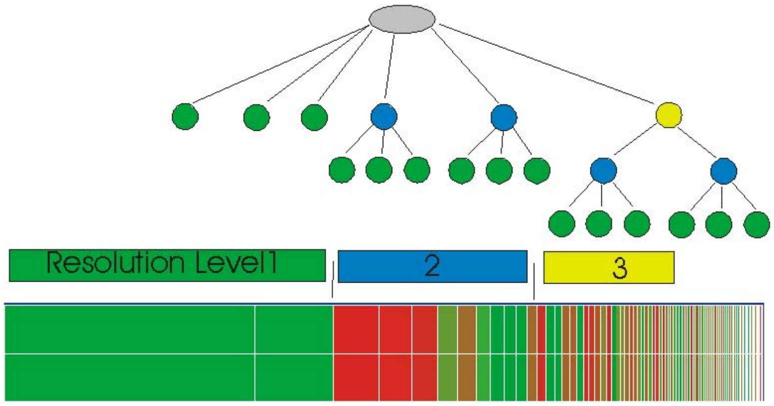

Figure 5: MRO Tree for the stock market example: Actual stock prices are single MRO's, historic ones are accumulated based on their timestamps (relevance) to minute, hour, day,. . . .

problem as well as an application example. Further research is necessary to use our technique as general fully automatic extension for visualization techniques, but our initial experiments have shown that the Multiresolution approach has the potential to increase scalability of visualization techniques in terms of data-set size and dimensionality, in order to keep pace with the increasing size of today's data sets. Future work will include the improvement of the prototype, the implementation of more general relevance functions and the application to other visualization techniques and example data sets.

References

1. M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Visualization '96, Hot Topic Session, San Francisco, CA*, 1996.
2. D.V. Beard and J.Q. Walker. Navigational techniques to improve the display of large two-dimensional spaces. *Behaviour and Information Technology*, 9(6):451–466, 1990.
3. E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: The see-through interface. In *Proc. SIGGRAPH '93, Anaheim, CA*, pages 73–80, 1993.
4. S. Eick and A. Karr. Visual scalability. *In J. of Computational and Graphical Statistics*, 1(11):22–43, 2002.
5. Lori A. Freitag and Raymond M. Loy. Adaptive, multiresolution visualization of large data sets using a distributed memory octree. In *Supercomputing '99: Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM)*, page 60, 1999.
6. Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *VISUALIZATION '99: Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, Washington, DC, USA, 1999. IEEE Computer Society.

7. G. Furnas. Generalized fisheye views. In *Proc. Human Factors in Computing Systems CHI 86 Conf., Boston, MA*, pages 18–23, 1986.
8. Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *VIS '90: Proceedings of the 1st conference on Visualization '90*, pages 361–378, 1990.
9. D. Jerding and J. Stasko. The information mural: A technique for displaying and navigating large information spaces. In *Proc. IEEE Symposium on Information Visualization*, pages 43–50, 1995.
10. D. A. Keim, J. Schneidewind, and M. Sips. CircleView: a new approach for visualizing time-related multidimensional data sets. In *Proceedings of the working conference on Advanced visual interfaces AVI, Gallipoli, Italy*, pages 179–182, 2004.
11. Daniel A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.
12. Eleftherios E. Koutsofios, Stephen C. North, Russell Truscott, and Daniel A. Keim. Visualizing large-scale telecommunication networks and services (case study). In *VIS '99: Proceedings of the conference on Visualization '99*, pages 457–461, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.
13. Ying K. Leung, Robert Spence, and Mark D. Apperley. Applying bifocal displays to topological maps. *Int. J. Hum.-Comput. Interact.*, 7(1):79–98, 1995.
14. J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In *Proc. Human Factors in Computing Systems CHI '91 Conf., New Orleans, LA*, pages 173–179, 1991.
15. Themistoklis Palpanas, Michail Vlachos, Eamonn Keogh, Dimitrios Gunopulos, and Wagner Truppel. Online amnesic approximation of streaming time series. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*, page 338. IEEE, 2004.
16. Catherine Plaisant, Jesse Grosjean, and Benjamin B. Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 57, Washington, DC, USA, 2002. IEEE Computer Society.
17. R. Rao and S. K. Card. The table lens: Merging graphical and symbolic representation in an interactive focus+context visualization for tabular information. In *Proc. Human Factors in Computing Systems CHI 94 Conf., Boston, MA*, pages 318–322, 1994.
18. Manojit Sarkar and Marc H. Brown. Graphical fisheye views of graphs. In Penny Bauersfeld, John Bennett, and Gene Lynch, editors, *Human Factors in Computing Systems, CHI'92 Conference Proceedings: Striking A Balance*, pages 83–91. ACM Press, Mai 1992.
19. B. Shneiderman. The eye have it: A task by data type taxonomy for information visualizations. In *Proc. 1996 IEEE Conference on Visual Languages*, pages 336–343, 1996.
20. E. Wegman. Huge data sets and the frontiers of computational feasibility, 1995.
21. Pak Chung Wong and R. Daniel Bergeron. Multiresolution multidimensional wavelet brushing. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, pages 141–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.
22. Pak Chung Wong, A. H. Crabb, and R. D. Bergeron. Dual multiresolution hyperslice for multivariate data visualization. In *INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, page 74, Washington, DC, USA, 1996. IEEE Computer Society.
23. J. Yang, M. O. Ward, E. A. Rundensteiner, and S. Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 19–28. Eurographics Association, 2003.