Output-oriented rendering for Information Visualization

Matej Novotný
mnovotny@sccg.sk

Abstract

With rapid increase of data volumes in all the areas of research it becomes a crucial issue to be aware of the size of data and adapt the data processing techniques accordingly. This paper presents such innovations to information visualization and especially to parallel coordinates, a popular information visualization tool.
The presented contribution, called output-oriented rendering, treats large data with respect to the final visual output. The display is divided into smaller fractions and a special choice of data aggregation is used that reduces the data load and clarifies the view. The concept of output-oriented rendering is general and can become a part of many other visualization techniques. The resulting display reacts interactively even for large data and is able to display millions of data items without overplotting or losing important information.

1. Introduction

Information visualization represents an established set of techniques and approaches used to communicate abstract information in a graphical and usually interactive way. A well-developed information visualization display enhanced by effective interaction apparatus will provide invaluable insight into data for exploratory, analytical or presentational purposes. However, the visualization methods often face challenges that decrease visual clarity, hide important data or slow down the visualization below interactive level. One of the main causes of such behavior is large data.

While recent technical development enables us to render large numbers of graphical elements and transform the geometry using a wide variety of hardware-accelerated operations, the development of data acquisition and data storage technology counterbalances this improvement by giving rise to huge volumes of information. In addition to that, the human visual system with its limits and specific properties introduces another aspect to the competition between size of the data and economy of the visualization. It is not sufficient to rely solely on technical development, especially when considering the human aspect. A display can be capable of rendering millions of graphical items yet the overall information value of such display is diminished if the view is crowded, overplotted and hard to read. Therefore a more radical approach is necessary, that takes into account both computer limitations and human limitations. To achieve such improvement, let us consider two main resources of an information visualization display – information and interaction.

It is relatively straightforward that large data tends to exhaust both of these resources. An inefficient display filled with millions of items does not provide much information and it is hard to be interacted with because it responds slowly. While interaction, mostly due to the aspect of speed, is a resource that can be regained by technical improvement, information, on the other hand, is a value that is often increased only by a technological improvement. Matters become even more complicated when the mutual interconnection of information and interaction demands appears. For example, transparency is used to increase information value of a display. But it also introduces a severe performance hit and threatens interaction thus [FLOREK]. As another example, the speed of the visualization is often improved by a lower level of detail. This indeed does improve interaction but it also destroys certain information. These tradeoffs become very prominent in information visualization of large data. The introduced output-oriented rendering tries to preserve both information and interaction value and to improve each of them without causing much damage to the other.

2. Output-oriented Rendering

The basic philosophy of output-oriented rendering (and output-awareness in general) is to consider the actual visual effect of graphical elements and consider this attribute when deciding which of them have to be displayed at what level of detail or if they have to be displayed at all. This concept has been present in computer graphics since the very beginning. Geometry cropping, backface culling or BSP trees were designed to hide elements that produce no effect to the final display. Another example are the level of detail techniques in real-time rendering which replace large models that would have to be rendered to a small screen space by their simpler versions and thus they gain performance without losing visual quality.

Interpreting this approach to information visualization means to display only those data entries that will cause a relevant change of information value of the display. In screen space, it is achieved by updating only those parts of the view that are changed. In data space, the data entries are processed to filter out those entries that would provide little or no information gain within the resulting display.

A natural question in this context would be: How to assess the information gain of a certain element? There are several sophisticated methods of estimating the information value of a display but for sake of simplicity let us focus only on the actual visual difference caused by rendering an item. Even in this simplified case, the visual importance of an item depends on the actual rendering technique and on the specific properties of this technique. Nevertheless, at least one common rule can be abstracted that holds for all techniques. If an item is to be rendered to a certain area of the screen, its visual importance decreases with the density of this area (i.e. the number of items already occupying this area.) Considering the limitations of a computer display, the visual importance can even decrease down to zero if the area is “full” in terms of visualization. An example implementation of this approach can be seen in [FLOREK] where stencil test is used to avoid meaningless overplotting of fully opaque areas.

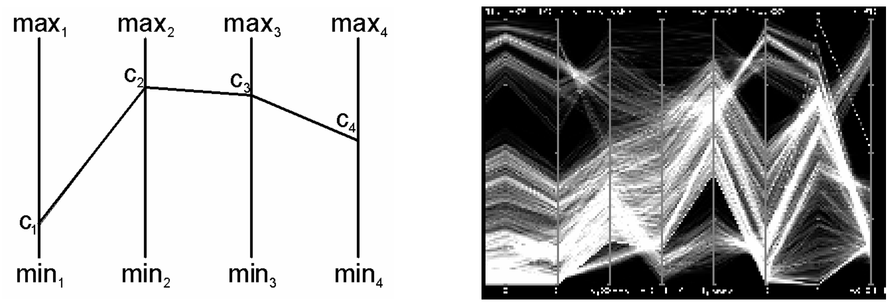
Since information visualization displays are rarely static, another rule for an output-oriented visualization is to preserve the graphical information between succeeding steps in interaction. Only those sub-areas of the display that actually change need to be re-rendered. With a little help from graphical hardware, this rule would result in a display divided into multiple high precision textures placed as layers one above another. By fractioning the display the changes triggered by interaction can be localized to a much smaller area and the rest of the view can remain untouched. Moreover, sophisticated transformations of the display can be achieved by performing only pixel-based operations on the stored output [JOHANSSON].

3. Output-oriented Abstraction

The principles of output-oriented rendering can be applied to data abstraction as well. Abstraction is used in information visualization to decrease the number of displayed elements and to simplify the view in order to increase its clarity. If we consider the visual importance of the individual elements we can create abstraction that has a largely reduced number of elements and still produces the same visual output. The benefits of such abstraction are obvious.

4. Implementation in Parallel Coordinates

The approach presented in this paper will be demonstrated on parallel coordinates. As introduced by [INSELBERG], this popular information visualization technique is suitable for high-dimensional data. It is especially useful for observing multidimensional structures and trends which don't show up so easily in displays of lower dimension capacity. However, due to the nature of this technique, the final display easily gets crowded. Moreover, since multiple (and often semi-transparent) lines are rendered for each data entry, it is also very performance-demanding in large data visualizations.

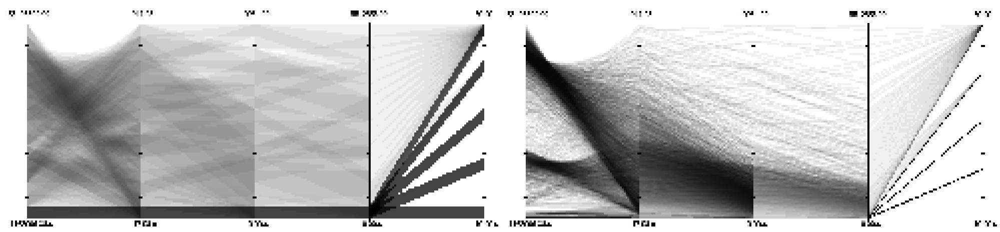


Parallel coordinates: in contrast to the Cartesian coordinate system, the axes in parallel coordinates are placed parallel and equidistant. An n-dimensional point $C(c_1,c_2,\ldots c_n)$ is drawn as a poly-line connecting the positions of $c_1,c_2\ldots c_n$ on their respective axes. An example of a real world application of parallel coordinates is shown in the picture on the right.

The following modifications were proposed and successfully implemented:

- View is divided into layers. Each layer represents a certain layer in the data – a subset of the original data sharing a certain conceptual property.
- Layers are divided into segments. Each segment is an area between two adjacent axes of the parallel coordinates plot.
- Data is binned before rendering. This part is explained in a separate section later in this paper.

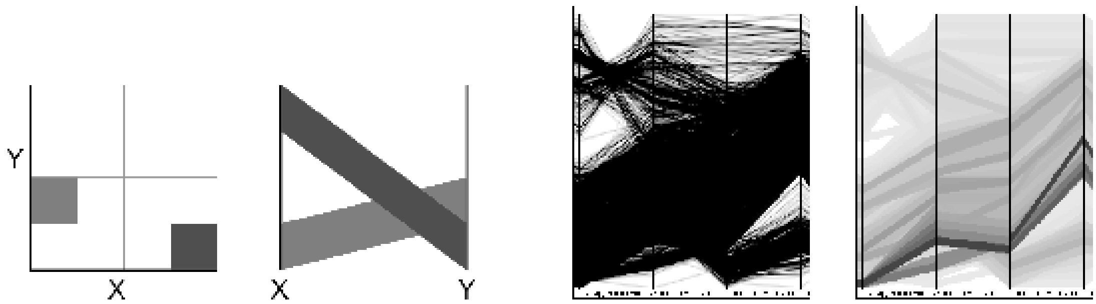
The resulting display is thus divided into *layers* × *segments* parts which localize particular changes to a small portion of the screen. The most frequent changes in parallel coordinates include changes of the mapping on individual axes and brushing. The mapping changes are easily localized so that at most two segments need to be updated for a single change. The brushing (selecting a certain subset of data), which usually affects the whole view is localized to a separate layer. Thus changes of the focus do not trigger update of the whole view.



Left picture shows an output oriented parallel coordinates view binned using 8×8 bins. The right shows the same using 64×64 bins. Note the precision of the visual output that is virtually the same as if line-based parallel coordinates were used.

5. Binning

Generally, binning is a process that converts data to a frequency-based representation by dividing the data space into a set of multidimensional intervals – called bins – and assigning to every bin an occupancy value which determines the number of data records that belong to the bin [SILVERMAN]. However, the exponential growth of the total number of bins would cause enormous memory demands when truly multidimensional data is considered. But due to our output-oriented approach, we are not affected by this curse of dimensionality as we apply binning not to the original ***n***-dimensional data but to the 2D visualization space.



Two-dimensional intervals (squares in first picture) are rendered as parallelograms in parallel coordinates (second picture). Third picture shows an example of parallel coordinates rendered using lines and the fourth picture shows the same plot displayed using bins. Note the changes in the clarity of the view and the revealed distribution of density

For each pair of adjacent axes, representing a pair of dimensions in the data, we bin the particular two-dimensional subspace into $b \times b$ bins. The resulting set of bins (for one pair of axes) forms a so called *bin map* and can be thought of as a 2D histogram of the distribution of all line segments between the two axes [NOVOTNY].

The bins are rendered as parallelograms connecting particular intervals on each axis. Due to its scalability, binning can either be used to provide a quick and cheap overview of the data or it can fully replace the line-based rendering without loss of information. Moreover, it provides an even better visualization than the original line-based rendering since it no longed depends on the size of the original data and thus various transformations can be applied interactively. The visual clarity is also improved, since the transparency of the elements – which visually encodes the density – can be transformed at no cost and many non-linear transfer functions are available. Such transfer functions are important for observation of large data sets where a simple linear mapping between density and opacity usually results in fully overdrawn display with few opacity values between 0 and 1.

6. Results

The display presented in this paper proves the validity of the concept of output-oriented rendering. It provides examples of how an output-oriented visualization can easily cope with large data, clean up overcrowded displays and provide fast interaction while still maintaining certain information value. In addition to that, the presented approach is scalable and in the case of parallel coordinate it even provides better information value with much less data entries than a display which renders all data entries.

The concept of output-oriented rendering can easily be applied to many other information visualization views – to scatterplots, starplots, network visualizations etc.

7. Acknowledgments

The author would like to thank Helwig Hauser (University of Bergen), Rober Kosara (University of North Caroline at Charlotte) and Harald Piringer (VRVis Research Center, Vienna) as the people whose ideas are a significant part of this project.

This research was partially financially supported by the European Social Fund.

8. References

[FLOREK] M. Florek and M. Novotný. Interactive Information Visualization using Graphics Hardware. Poster in *Spring Conference on Computer Graphics 2006*, pages 65-68, ISSN 1335-5694

[INSELBERG] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizingmultidimensional geometry. In *Proc. of IEEE Visualization*, pages 361–378, 1990.

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

[NOVOTNY] M. Novotný and H. Hauser. Outlier-Preserving Focus+Context Visualization in Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics* ,vol. 12, no. 5, pp. 893-900, September-October, 2006.

[SILVERMAN] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, 1986.