Color Lines View: An Approach to Visualization of Families of Function Graphs

Krešimir Matković
VRVis Research Center
Matkovic@VRVis.at

Denis Gračanin
Virginia Tech
gracanin@vt.edu

Zoltán Konyha
VRVis Research Center
Konyha@VRVis.at

Helwig Hauser
University of Bergen
Helwig.Hauser@UiB.no

Abstract

Data sets often include information that can be represented as a mapping that describes how a dependent variable depends on an independent variable. Such a mapping, usually represented as a function graph, can be parameterized to provide a family of function graphs. The challenge is how to efficiently aggregate individual function graph views to represent the whole family and allow visual analysis and search for patterns. We propose a novel view, called the color lines view, which provides a two dimensional, rectangular view where each line represents a single function graph. The points on the line correspond to values of the independent variable. The point colors represent the value of the dependent variable. The lines, placed next to each other in parallel, show a family of function graphs. The color lines view offers sorting and brushing features which support visual analysis procedures that are difficult to perform with previously existing views.

Keywords— **dense pixel displays, color lines view, sorting, brushing, multiple views, aggregated views, visual analysis**

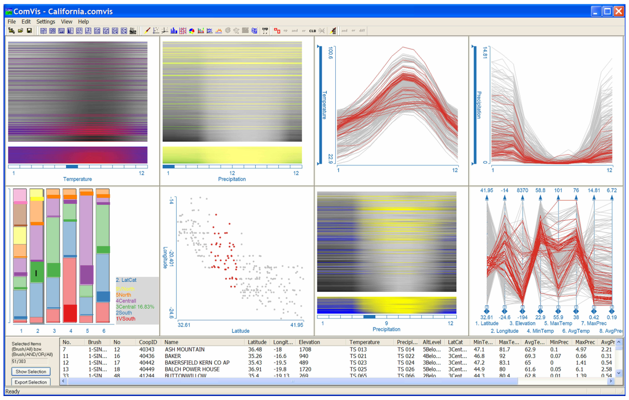
1 Introduction

Interactive visual analysis and information visualization often deal with multidimensional spaces. Each dimension in the multidimensional space usually corresponds to a scalar value. Problems from scientific domains (but not only those) sometimes follow a different data model where dimensions can be, in addition to scalar values, mappings or data series. We described such a model in [4]. In that model a dimension can have values that are mappings of one or more independent variables. In this paper we limit our discussion to the case where there is exactly one independent variable in such mappings. In that case the mapping can be represented as a function graph (curve). All function graphs that belong to the same dimension (all values for that dimension) constitute a family of function graphs. A family of function graphs can be visualized and analyzed using the curves view introduced in [4]. Although the curves view proved to be very powerful, sometimes it is not sufficient.

In this paper we introduce a novel view, the color lines view, for the visualization of a family of function graphs. It depicts each function graph (curve) as a straight line. The horizontal positions on the line correspond to values of the independent variable. The corresponding function graph value is indicated by the color of the point. In other words, the color of the line changes along the independent variable based on the value of the dependent variable.

Since there is a whole family of function graphs, there are many such lines that are drawn beside each other. The color lines view depicts the nature of function graphs and any clusters in the family very well. Since the color lines view is usually used in a multiple linked views setting, special attention must be paid to interaction features. The interaction features include various sorting and brushing possibilities. The combination of sorting and brushing makes the color lines view a very powerful technique for the visual exploration of families of function graphs.

The color lines view is integrated in an interactive visualization software called ComVis [6] that provides conventional views like 2D and 3D scatter plots, parallel coordinates, histograms, as well as a curves view for displaying function graphs. ComVis provides features like composite brushing, sorting and linking that are very useful for visual analysis (Figure 1).



No.	Brush	No	CoopID	Name	Latitude	Longit...	Elevation	Temperature	Precipi...	AltLevel	LatCat	MinTe...	MaxTe...	AvgTe...	MinPrec	MaxPrec	AvgPr
7	1-SIN...	12	40343	ASH MOUNTAIN	36.48	-18	1708	TS 013	TS 014	5Belo...	3Cent...	47.1	81.7	62.9	0.1	4.97	2.21
11	1-SIN...	16	40436	BAKER	35.26	-16.6	940	TS 021	TS 022	4Belo...	3Cent...	46.8	92	69.3	0.07	0.66	0.31
12	1-SIN...	17	40442	BAKERSFIELD KERN CO AP	35.43	-19.5	489	TS 023	TS 024	3Belo...	3Cent...	47.2	83.1	65	0	1.41	0.54
13	1-SIN...	18	40449	BALCH POWER HOUSE	36.91	-19.8	1720	TS 025	TS 026	5Belo...	3Cent...	44.9	80	61.6	0.05	6.1	2.58
33	1-SIN	48	41244	BUTTONWILLOW	35.4	-19.13	269	TS 065	TS 066	2Belo	3Cent	44.3	80.4	62.8	0.01	1.39	0.54

Figure 1: ComVis: an example of coordinated multiple views (color lines views, curves views, bar charts, scatterplot, parallel coordinates) and composite brushing.

2 Related Work

Previous work related to the color lines view can be classified into three fields: focus and context visualization, pixel based visualization and color mapping methods.

The table lens view [7] uses a focus+context (fisheye) technique for tabular information and displays important label information and multiple distal focal areas. It can visualize and help make sense of large tables using a graphical mapping scheme for depicting table contents by fusing symbolic and graphical representations into a single, user customizable coherent view.

The information mural view [2] provides a miniature version of the information space using visual attributes (gray-scale shading, intensity, color, and pixel size) and anti-aliased compression techniques. The view alleviates problems due to the information bandwidth constraints resulting from the limited number of pixels on the screen.

Dense pixel displays [3] simultaneously represent many data items by mapping each of them to a pixel and arranging the pixels suitably. The value of each dimension is visualized as a colored pixel. The pixels belonging to each dimension are grouped into adjacent areas. Visualizations resulting from changes in the arrangement and the coloring of the pixels provide detailed information on dependencies and patterns [3]. The proper arrangement of pixels is important to allow pattern discovery and support visual analysis. Data sets can have a natural ordering of elements (e.g. time-series data) or can be without ordering (e.g. categorical data) [3].

Mapping data values to color is a critical aspect of pixel-based methods [3]. Color gradients can provide more just noticeable differences than gray scale ones. A single optimal color scale may not be feasible, but a combination of several interchangeable color scales can provide a better perception of the information [5].

3 Proposed Approach

We use a data model where dimensions can be represented as mappings or data series (function graph) in addition to scalar values. A data set has m "input" dimensions (independent variables) and n "output" dimensions (dependent variables).

The independent variables (input dimensions) have scalar values and can be expressed as $\mathbf{x} = [x_1, \ldots, x_m] \in I$. Here I denotes the set of all possible combinations of independent variables values.

The dependent variables (output dimensions) are functions of the independent variables. When a value of a dependent variable k is a function graph (data series over t) $f_k(\mathbf{x}, t)$, it contains o data elements, one for each value of $t \in \{t_1, \ldots, t_o\}$. A family of function graphs is then a set of function graphs for each possible value of $\mathbf{x}$, $\{f_k(\mathbf{x}_i, t) | \forall \mathbf{x}_i \in I\}$.

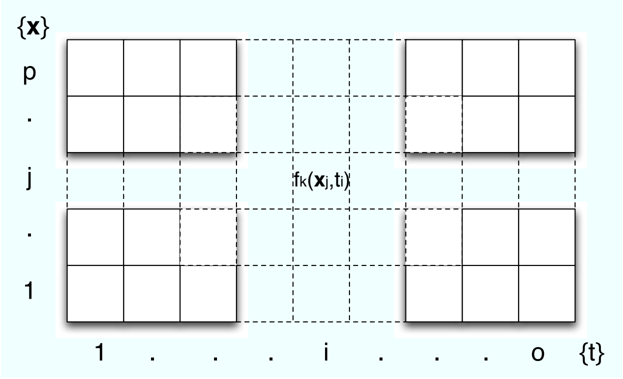


Figure 2: A rectangular pixel arrangement of $o \times p$ for a family of function graphs $f_k(\mathbf{x}, t)$.

The number of function graphs in a family (p) is equal to the number of all possible values of $\mathbf{x}$. Therefore, a family of function graphs for a dependent variable f_k contains p function graphs, each with o data points, a total of $o \times p$ data points. There are many data sets that fit in this data model. An example is provided in Section 4.

Figure 2 shows that those data points can be arranged in a rectangle. The color of the pixel at coordinates (i, j) corresponds to the value of $f_k(\mathbf{x}_j, t_i)$. This dense pixel representation provides a good overview of the family of function graph.

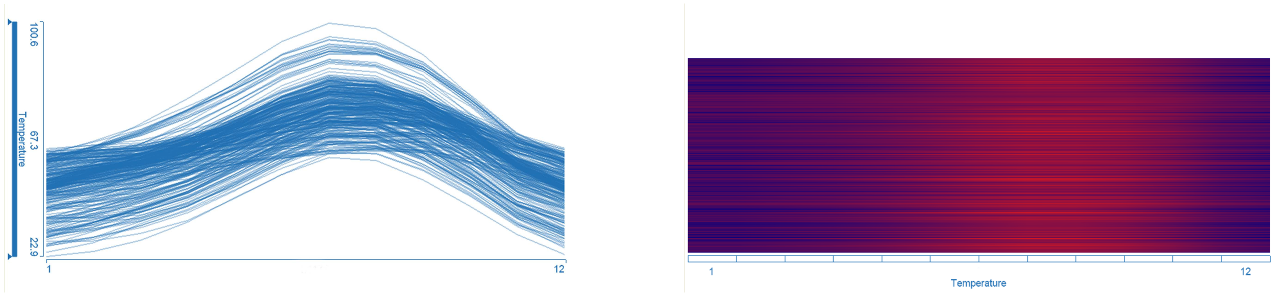
The visual analysis procedures, or search for patterns, involve transformations of the dense pixel display in a way that allows grouping of "similar" pixels while preserving the rectangular shape. We assume that there is a natural ordering of the values of t so the transformations should preserve that ordering and keep individual data series intact. Therefore, a transformation is limited to swapping rows within the rectangle. Given the initial ordering of rows, $1, \ldots, p$, the end result of a transformation is a permutation of those rows. The number of all possible permutations of p elements is $p!$, which is prohibitive for exhaustive exploration. Therefore, some guiding criteria are needed for selecting a permutation. We suggest internal sorting, external sorting and brushing criteria.

The internal sorting means that for a fixed value of t, t_i, we sort rows based on the values of their i-th pixels. The corresponding permutation π that is the result of the sorting of the values at the i-th pixel has the property:

$$f_k(\mathbf{x}_{\pi_1}, t_i) \leq \ldots \leq f_k(\mathbf{x}_{\pi_l}, t_i) \leq \ldots \leq f_k(\mathbf{x}_{\pi_p}, t_i)$$

where π_l is the row index with the l-th smallest value of f_k for the fixed t_i.

The external sorting does not use f_k but an external function, $g(\mathbf{x}, t_i)$ to determine the order of the rows such



(a) Temperature data in the curves view. There are 303 curves over 12 time points in the figure.

(b) The same data set in the color lines view. There are 303 lines and 12 segments in each line, following the arrangement in Figure 2. Blue indicates low, red indicates high temperatures.

Figure 3: Temperature data from 303 stations over a period of 12 months as a family of function graphs in (a) the curves view and (b) the color lines view. The x axis represents the months January through December in both views.

that

$$g(\mathbf{x}_{\pi_1}, t_i) \leq \ldots \leq g(\mathbf{x}_{\pi_l}, t_i) \leq \ldots \leq g(\mathbf{x}_{\pi_p}, t_i)$$

Brushing does not permute the rows, but marks a range of rows to be observed in subsequent permutations. The focus and the context are shown in different color schemes.

Typical visual analysis procedures include:

- Sort rows based on the i-th pixel values and observe sorting patterns for other pixel positions.
- Sort rows based on the i-th pixel values, brush a range of the sorted rows, perform another sort and observe the distribution pattern of the brushed rows.

4 Color Lines View

The color lines view is used to visualize and explore a family of function graphs. The main idea is to show each function graph (curve) in the family as a straight line that changes its color to indicate the function value. The color lines view shows the lines placed one above the other.

We will use a simple meteorological data set to introduce the view. The data set consists of data from 303 meteorological stations in California [1]. Various data including longitude, latitude, elevation and average temperature is given for each station. Monthly average temperatures are also given as a mapping or function over time. Note that this reduced data set consists of only 303 stations and covers only one year. It is used only for illustration purposes. All examples are given in order to explain the new color lines view.

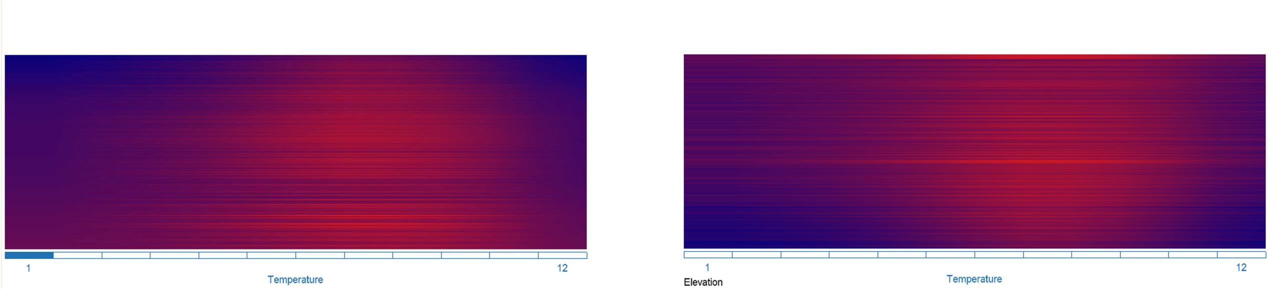
This data set matches the data model described in Section 3. There is one independent variable (input dimension), the station id ($m = 1$) with 303 different values ($p = 303$). The dependent variables (output dimensions) include longitude, latitude, elevation, and temperature ($n = 4$). Longitude latitude, and elevation are scalars while temperature values are data series of 12 points, one sample per month ($t \in \{1, \ldots, 12\}$, $o = 12$.

There is one temperature function graph for each station. The set of all temperature function graphs (303 curves) constitutes a family of function graphs. Figure 3 shows the temperature graphs in two different views. The overall trend, cold winters and warm summers can clearly be seen in both views but the curves view is a clear winner for such an overview.

If the user needs a deeper insight into the data, a simple overview provided in Figure 3(b) is not sufficient. However, if the user can discover patterns in the data that are less obvious in other visualizations, then he or she gets additional insight. The drawing order of lines obviously influences the patterns that appear. Moreover, if we provide an intuitive way of changing the order, we can support visual analysis.

The basic interaction with the view is sorting the lines (Figure 4). The user can simply click a location on the x axis (independent variable, time in our example) and lines will be sorted and drawn according to the values of the dependent variable at this position. Figure 4(a) shows that if the user selects January, the lines with low January temperatures are drawn at the top and the lines with high January temperatures are drawn at the bottom. We can see similar color patterns for January and December, indicating that those temperatures are correlated, as expected. There are many lines at different rows that are blue in the middle, which means low summer temperatures. Therefore, we can see that stations which are warm in January are not necessarily warm in June or July.

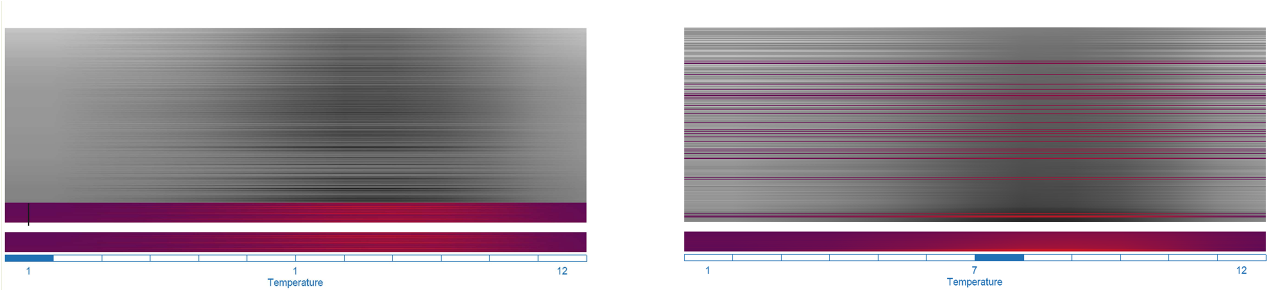
Since this family of function graphs belongs to a multidimensional space we can sort it by some other dimension. Let us examine what happens if we sort temperature lines according to the elevation of the stations. High elevations are drawn in the bottom and low elevations in the top in



(a) The color lines view is sorted by January temperatures. This is indicated by the blue rectangle in the bottom left corner. The colors are remarkably varied in the middle, which means there is little direct correlation between January and summer temperatures.

(b) The color lines view is sorted by elevation. Cold winters correspond to high elevations, but hot summers do not correspond to low elevation.

Figure 4: Sorting the lines by January temperatures (internal sorting) and by elevation (external sorting). The data is same as in Figure 3(b).



(a) High January temperatures are brushed by the black vertical line near the left edge of the view. The brushed lines are shown in the split view under the main one, too.

(b) The lines are sorted by temperature in July. The brushed set is quite scattered in the context, but the split view still allows its detailed exploration.

Figure 5: Brushing and sorting in the color lines view.

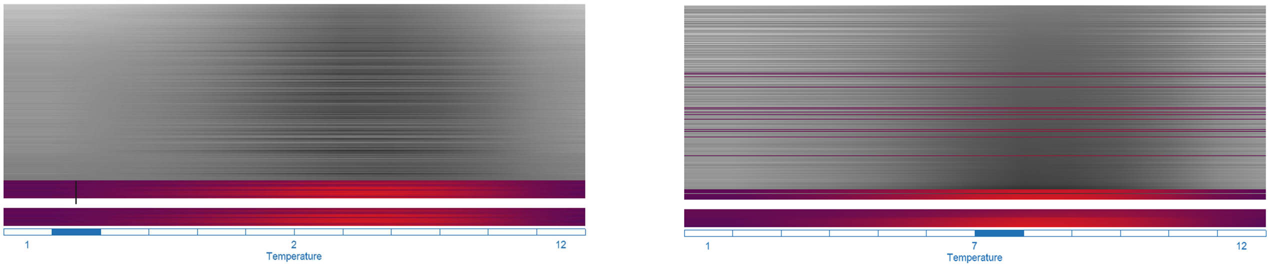
Figure 4(b). We can see that cold winters (dark blue at left and right) correspond to high elevations, as expected. On the contrary, hot summers (red in the middle) can be seen at many different elevations. The highest peak temperature is actually near the middle. The correlation between elevation and high summer temperatures is less direct.

In order to support more advanced analysis we have embedded the color lines view in a multiple linked views visualization system (Figure 1). The color lines view can display focus+context based on brushing in other views. A different (gray scale) color gradient is used for the context. Furthermore, the view is split. The brushed lines are drawn separately under the view so that patterns in brushed lines can be explored in more detail. They are drawn in their original context as well (Figure 5).

Lines in the color lines view can be brushed by simply drawing a line across them. Remember that each line represents one item in the data set. Therefore, whole lines are selected and the set of selected lines is independent from the actual horizontal location of the brush. Actually, the brush line is displayed as a vertical line at a horizontal coordinate corresponding to the sorting position. For example, if the view is sorted by January and a brush is created at any horizontal position then it will be drawn in the left of the view. This is useful, because the sorting can be changed later, but the position of the brush indicates the sorting applied at the moment of brushing.

The highest January temperatures are selected in Figure 5(a) and then the view is sorted by July (Figure 5(b)). The lines in the focus set are sorted in the lower part. We can also see those lines in the context where they are spread almost all over the view. This means that warm January does not necessarily mean warm July in our data set.

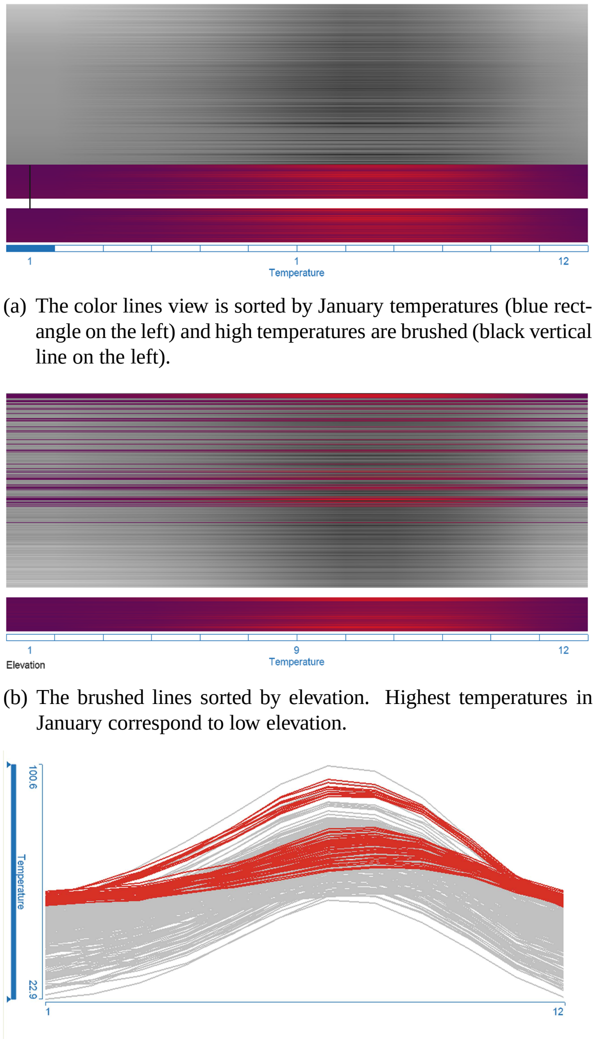
Figure 6 shows the same procedure for February. We sort the lines by values in February, brush warm temperatures and sort the lines according to July. Figure 6(a) shows the view with the highest February temperature selected. We sort the view for July (Figure 6(b)). The color lines view shows a peak for the July values. Interestingly, warm February does mean warm July at most stations!



(a) High temperatures in February are brushed.

(b) The lines are sorted by temperatures in July. Most of the brushed set stays clustered together.

Figure 6: Brushing and sorting based on the February data. Compare this to Figure 5.

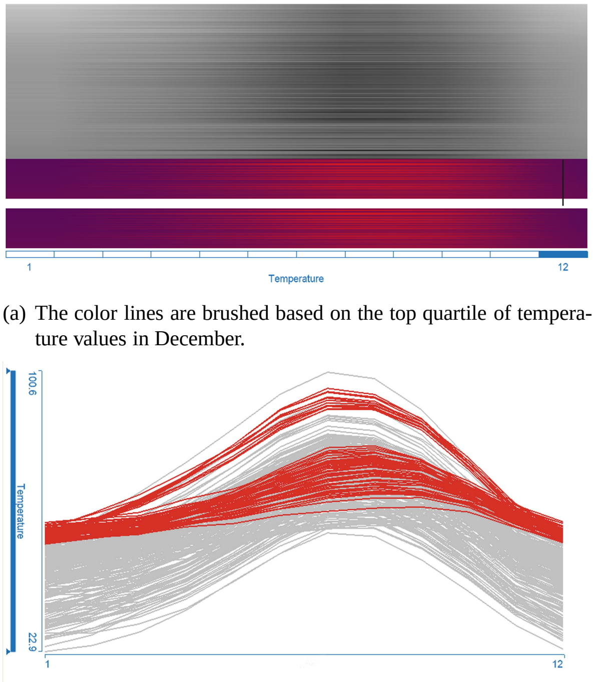


(a) The color lines view is sorted by January temperatures (blue rectangle on the left) and high temperatures are brushed (black vertical line on the left).

(b) The brushed lines sorted by elevation. Highest temperatures in January correspond to low elevation.

(c) The corresponding curves view shows how curves with high temperature values in January (and December) diverge in temperatures for summer months.

Figure 7: Brushing and external sorting based on the January data and elevation.



(a) The color lines are brushed based on the top quartile of temperature values in December.

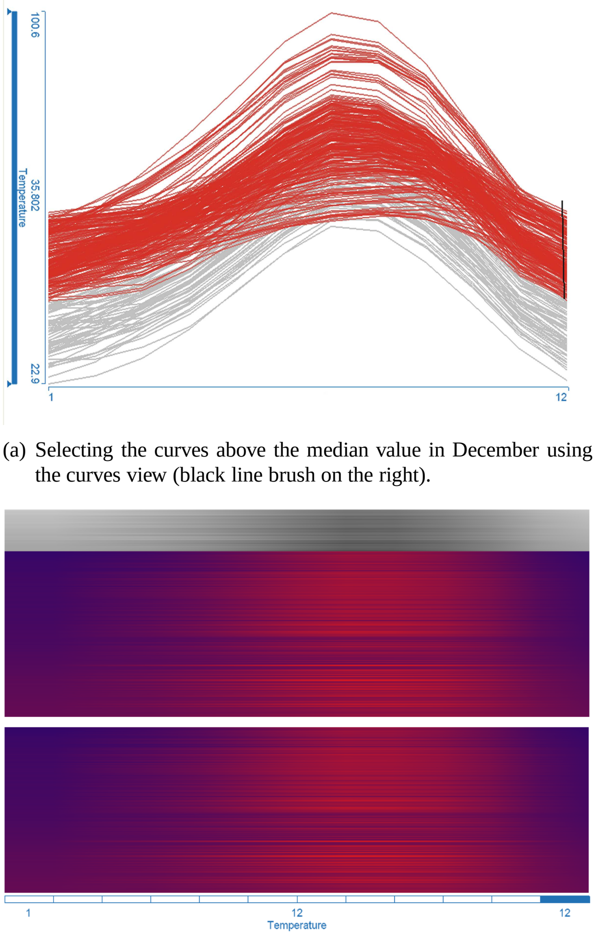
(b) The same items are highlighted in the corresponding curves view. The percentage of highlighted curves is less obvious due to overlapping.

Figure 8: The color lines view is more suitable for brushing a specific percentage of the total number of curves.

Figure 7 illustrates brushing combined with the external sorting. The highest January temperatures are selected in Figure 7(a). The view is sorted by elevation in Figure 7(b). We can see that high temperatures in January correspond to low elevation.

One of the advantages of the color lines view compared to the curves view is the better control over brushing (Figure 8). In the color lines view, we can see precisely what percentage of the data set is brushed. Figure 8(a) shows a brush in the color lines view that selects the top quartile of the temperature values. Its length is exactly one fourth of the height of the rectangle. It is not possible to determine the exact percentage of curves brushed in the corresponding curves view (Figure 8(b)) due to the overlapping of curves.

Figure 9 illustrates a different point. After selecting the top half of the temperature values in the curves view in Figure 9(a), we discover in the linked color lines view (Figure 9(b)) that significantly more than one half of the curves are brushed. Therefore, the curves view is better for brushing based on median or similar range values.



(a) Selecting the curves above the median value in December using the curves view (black line brush on the right).

(b) The corresponding color lines view shows that significantly more than half of curves is selected.

Figure 9: The curves view is more suitable for the brushing of curves within a range of values.

Conclusions

Exploring patterns in families of function graphs is a common task in many application domains. We have introduced a new aggregated view, the color lines view, for the visualization of families of function graphs. The color lines view can display patterns and clusters in the data set that are difficult to show with other, “traditional” views. The sorting and brushing make it especially suited to the comparison of patterns in a family of function graphs at various values of the independent variable. These features of the color lines view provide support for complex visual analysis procedures.

Future work will proceed in two directions. We will examine aggregation methods for data sets where p and o are substantially larger than the number of pixels on the display. We will also explore the use of large ($>$ 10 megapixels) displays and related usability issues.

Acknowledgements

We thank Audrey Cahill of AVL-List GmbH for her valuable comments. Parts of this work have been carried out at the Center for HCI at Virginia Tech (`http://www.hci.vt.edu/`) and in the scope of applied and basic research at the VRVis Research Center which is funded by an Austrian governmental research program called K plus (`http://www.kplus.at/`).

References

[1] World data center for meteorology. `http://www.ncdc.noaa.gov/oa/wmo/wdcamet.html`.

[2] Dean F. Jerding and John T. Stasko. The information mural: A technique for displaying and navigating large information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):257–271, July-September 1998.

[3] Daniel A. Keim. Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, January–March 2000.

[4] Zoltán Konyha, Krešimir Matković, Denis Gračanin, Mario Jelović, and Helwig Hauser. Interactive visual analysis of families of function graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1373–1385, November/December 2006.

[5] Haim Levkowitz and Gabor T. Herman. Color scales for image data. *IEEE Computer Graphics and Applications*, 12(1):72–80, January 1992.

[6] Krešimir Matković, Mario Jelović, Josip Jurić, Zoltan Konyha, and Denis Gračanin. Interactive visual analysis and exploration of injection systems simulations. In *Proceedings of the IEEE Visualization 2005 Conference*, pages 391–398, October23–28 2005.

[7] Tichomir Tenev and Ramana Rao. Managing multiple focal levels in table lens. In *Proceedings of the IEEE Symposium on Information Visualization 1997*, pages 59–63,122, 20–21October 1997.