Eurographics/ IEEE-VGTC Symposium on Visualization (2007)
Ken Museth, Torsten Möller, and Anders Ynnerman (Editors)

Depth Cues and Density in Temporal Parallel Coordinates

Jimmy Johansson[†], Patric Ljung[†] and Matthew Cooper[†]

[†]Norrköping Visualization and Interaction Studio, Linköping University, Sweden

Abstract

This paper introduces Temporal Density Parallel Coordinates (TDPC) and Depth Cue Parallel Coordinates (DCPC) which extend the standard 2D parallel coordinates technique to capture time-varying dynamics. The proposed techniques can be used to analyse temporal positions of data items as well as temporal positions of changes occurring using 2D displays. To represent temporal changes, polygons (instead of traditional lines) are rendered in parallel coordinates. The results presented show that rendering polygons is superior at revealing large temporal changes. Both TDPC and DCPC have been efficiently implemented on the GPU allowing the visualization of thousands of data items over thousands of time steps at interactive frame rates.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Display algorithms I.3.6 [Computer Graphics]: Interaction techniques

1. Introduction

Parallel coordinates [Ins85] is a technique used for analysing multivariate data on a 2D display. In parallel coordinates, the variables of each data item, $\boldsymbol{d}_m$, are mapped to parallel axes so that each multivariate data item is displayed as a polyline intersecting all axes. For data sets containing thousands of multivariate data items, the parallel coordinates display becomes cluttered so several techniques have been proposed to deal with this. See, for example, [FWR99, Hin87, WL97, JLJC05, NH06].

In time-varying multivariate data, there is an additional time dimension and the challenge of producing an informative visualization increases significantly. A time-varying multivariate data set is defined, in this context, as being one that contains one or several data items, $\boldsymbol{d}_m$, for each of a number of time steps, t_i, where each $\boldsymbol{d}_m$ contains values for at least two variables. This type of data can be found, for example, in the social science community where it is of interest to study how attributes from municipalities or countries relate to each other over long time periods. This can allow the user to discern how they are affected by, for example, taxes and laws.

Since the patterns in parallel coordinates are well-studied [Ins85, Weg90] and the technique is used in many disciplines it would be advantageous to also be able to use it for analysing time-varying multivariate data sets. This paper investigates *Temporal Parallel Coordinates* — 2D parallel coordinates containing an additional time dimension. Two approaches for visualizing time-varying multivariate data sets are introduced: (1) *Temporal Density Parallel Coordinates* (TDPC), which extends the use of density maps and transfer functions [JLJC05] to include a temporal window, and (2) *Depth Cue Parallel Coordinates* (DCPC), based on temporal binning, that employs perception-based colouring and concepts from volume rendering to reveal temporal positions.

The proposed techniques in this paper can be used to visualize large, time-varying multivariate data sets (thousands of data items and thousands of time steps) at interactive frame rates. Besides rendering lines to visualize actual values in a data set, the concept of visualizing changes between time steps is introduced. This is achieved by rendering polygons and is illustrated in figure 1 where a variable, x, changes value over time. Despite the completely different changes seen in figures 1(a) and 1(d), the resulting visualizations using lines in parallel coordinates (using the same variable on both axes) are identical, see figures 1(b) and 1(e). Visualizing the amount of change using polygons gives the distinctly different results illustrated in figures 1(c) and 1(f).

† {jimmy.johansson,plg,matt.cooper}@itn.liu.se

There is a well-known duality between Cartesian and parallel coordinates [Ins85, Weg90]. Points in Cartesian coordinates become lines in parallel coordinates and lines in parallel coordinates become points in Cartesian coordinates. There is also an analogy between lines in Cartesian coordinates and polygons in parallel coordinates (lines can be seen as an infinite number of points and polygons as an infinite number of lines). A visualization using polygons in parallel coordinates is therefore analogous to displaying the line segments between the points (figures 1(a),1(d)). As the distances between the points decrease, the line segments become shorter and consequently the visual result of using lines and polygons in parallel coordinates becomes increasingly similar. As the distances between the points increase, the line segments become more important in order to perceive how the points are connected. In parallel coordinates this suggests that using polygons would produce a visualization that better reveals large temporal changes in data, which is also supported by the results presented in this paper.

All techniques presented in this paper are based on textures and implemented on the GPU allowing interactive frame rates when analysing multivariate data items over thousands or even tens of thousands of time steps. The main contributions of this paper can be summarized as:

- The concept of visualizing the change between time steps in time-varying multivariate data sets using parallel coordinates.
- Temporal Density Parallel Coordinates (TDPC) that can be used to reveal discrete time steps and the amount of change occurring.
- Depth Cue Parallel Coordinates (DCPC) that can be used to reveal where in time data items are positioned as well as where in time changes occur.
- Efficient methods for interactive updates of temporal density maps and temporal binning.

The remainder of this paper is organized as follows. Section 2 presents related work in the area of parallel coordinates and visualization of time-varying data. Section 3 gives detailed descriptions of the proposed techniques. Implementation details are discussed in section 4. Results are presented in section 5 and conclusions in section 6.

2. Related Work

A number of attempts have been made to visualize time-varying multivariate data sets, see [MS03] for an excellent overview. Since this paper investigates temporal parallel coordinates, this section focuses on previous attempts to extend parallel coordinates to incorporate a time dimension.

To include a time dimension in parallel coordinates a single visualization for each time step could be used but for hundreds or thousands of time steps this approach is limited by the display area of a standard monitor, rendering each individual visualization too small to perceive structures. Superimposing parallel coordinates (one for each time step) could be done but would not reveal any temporal structure without further enhancement through, for example, an additional axis representing time [Eds03] or by using colour. The axes could also be used to represent different time steps for a single variable [Eds03]. For long time series these methods would be severely limited by cluttering. A trend graph, showing values for each variable over time, may be placed over the corresponding axis [ZLTS03]. This requires a trend graph for each variable for each data item and does not scale well with an increasing number of multivariate data items.

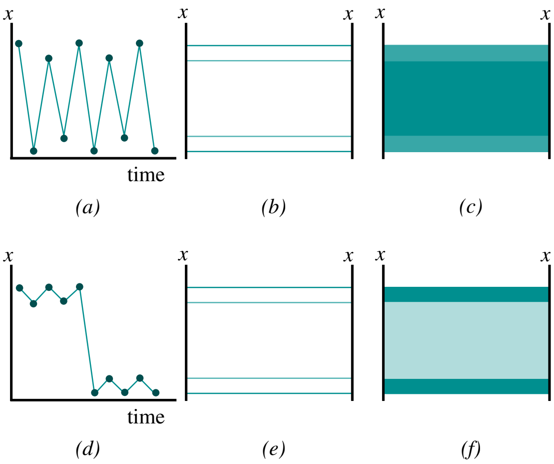


Figure 1: *Displaying the changes within two sequences of data values, (a) and (d), using polygons ((c) and (f)) rather than displaying the values at each time step using lines ((b) and (e)). Opacity is used to convey the number of overlapping changes occurring over time, from highly transparent (few overlaps) to fully opaque (many overlaps).*

It is possible to increase each axis in the parallel coordinates display to a two-dimensional space by using planes as axes [WLG97, RWK*06]. The additional dimension can be used to represent time. For large data sets, this extension suffers from cluttering artefacts and it is difficult to get an overview. A similar extension of parallel coordinates is extruded parallel coordinates [WLG97] where a number of bands or a surface can be used to represent how variables change over time. This approach is strongly limited in the number of data items that can be simultaneously displayed. The approach of unfolding parallel coordinates into a 3D display [FCI05] also suffers from cluttering when analysing large data sets. When using 3D displays with perspective projection, distances can be distorted and make it difficult to compare data items at different depths [NTPT96].

Instead of the traditional 2D axes configuration in parallel coordinates, a circular arrangement can be used to simultaneously visualize relationships between all variables and

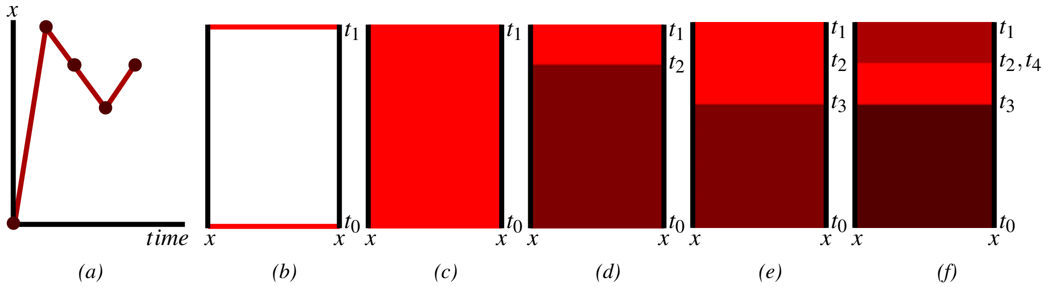


Figure 2: *Visualizing changes between consecutive time steps. As the value of x changes over time, new polygons are rendered to produce a density map.*

time. This configuration does not, however, reveal how variables are related over time. This can be done using a 2D display [TAS04] or a 3D display [JCJ05, JLJC06, TAS05].

Parallel coordinates has also been used for visualization of time-varying multichannel EEG data [tCMR07] where a min-max and density plot, describing extreme values and distributions of multivariate data items over time, is used in combination with parallel coordinates. This approach does not, however, reveal how variables correlate over time.

3. Visualizing Temporal Information

Using polygons to visualize changes between time steps in parallel coordinates is illustrated in figure 2 with an example using two axes. For simplicity the same variable, x is used on both axes. x has its minimum value at time step t_0 and its maximum value at time step t_1. Visualizing these two time steps by superimposing two parallel coordinates (one for each time step) gives two horizontal lines, see figure 2(b). Instead of drawing two lines, a polygon can be drawn to represent the change between two time steps (figure 2(c)). As the value of x continues to change over time, new polygons are added. At time steps t_2 and t_3 the value of x decreases and finally increases at time step t_4, see figures 2(d)–2(f). In this example, since the same variable is mapped to the two axes, the change has always been in the same direction, either positive or negative. For situations where there is both a positive and a negative change a twisted polygon is drawn, see figure 3.

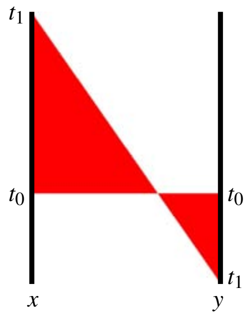


Figure 3: *A 'twisted polygon' is used to convey a simultaneous negative and positive change in two adjacent axes.*

3.1. Temporal Density Parallel Coordinates

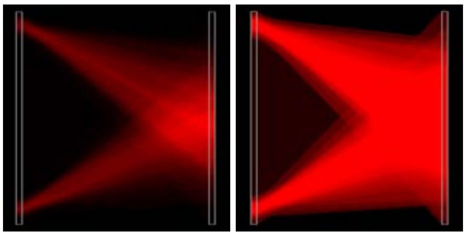
As illustrated in figures 2(c) – 2(f) the intensity of the polygons, representing the number of overlaps at each pixel, changes as more time steps are added. For a given pixel, k, a density value, ρ_k, can be defined as

$$\rho_k(t_a, t_b) = \sum_{i=a}^{b} \mathcal{C}(k, \Theta(t_i)) \tag{1}$$

where t_a and t_b represent the bounds of the visualized time period ($t_a < t_b$). $\mathcal{C}(k, \Theta(t_i))$ gives the total number of primitives intersecting pixel k. The set of M primitives representing M data items at time step t_i is referred to as $\Theta(t_i)$. In the case of using polygons, $\Theta(t_i)$ refers to the change between t_i and t_{i+1}. When rendering lines, $\Theta(t_i)$ refers to the discrete values of the M data items at time step t_i.

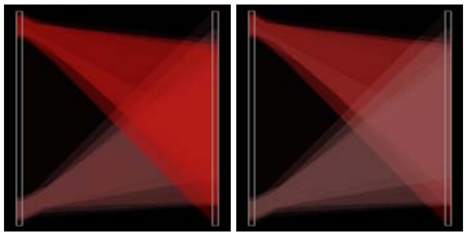
A common approach when analysing time-varying data sets is to concentrate the analysis on a subset of the data by means of a time window. Resizing and moving the time window reveals how data values change over time. When using this type of interaction on large data sets it is not feasible to re-render all data items for each new update of the visualization since the frame rate would be too low, resulting in tedious and time-consuming work for the user. Using equation 1 it is possible to make incremental updates as the bounds of the time window change. As a result interactivity is independent of the size of the time window, thus enabling analysis of thousands of time steps at interactive frame rates.

For large data sets, ρ_k will typically range from values close to zero to values of several thousands. Such a large range of density values cannot be directly displayed on a standard monitor. Thus, it is desirable to use Transfer Functions (TFs) to emphasize different parts of the density range. A TF is used to map density values to colours and opaci-



(a) (b)

Figure 4: *TDPC is used to explore the density over time. (a) a TF (γ = 1) is used to map density values to opacity. (b) a different TF (γ = 0.4) is used to emphasize low density regions.*



(a) (b)

Figure 5: *Results of using DCPC. A single hue is used and saturation and value are decreased for older changes. In (a) the data is viewed from the end of the time period. In (b) it is viewed from the beginning.*

ties [JLJC05]. A basic definition of a TF, $\mathcal{T}$, is

$$\langle \boldsymbol{c}, \alpha \rangle = \mathcal{T}(\tilde{\rho}) \tag{2}$$

where $\boldsymbol{c}$ is a colour, α is an opacity and $\tilde{\rho} = \rho / \max(\rho_k)$.

Setting a TF to control the opacity ($\alpha = \tilde{\rho}$) of a single hue, $\boldsymbol{C}$, according to

$$\mathcal{T}(\tilde{\rho}) = \langle \boldsymbol{C}, \tilde{\rho}^{\gamma} \rangle \tag{3}$$

gives the results illustrated in figure 4. In this figure polygons are used as primitives and the data set has a single data item including two variables over 200 time steps.

3.2. Depth Cue Parallel Coordinates

DCPC extends TDPC by also incorporating temporal information. To convey this information, a sense of depth needs to be represented in the display such that, for example, recent data in the examined time period are perceived as being in the foreground and older data as being in the background. An effective means by which this can be achieved is through a volumetric representation, compositing the data into the 2D display. The beginning of a time window is defined as t_a (old values) and the end as t_b (new values).

For the DCPC technique, ρ_k is partitioned into B bins, $\rho_k^1, \rho_k^2, \ldots, \rho_k^B$, to produce a 'volume'. The partitioning is made such that the number of time steps in the bins are as balanced as possible. The time interval $[t_a, t_b]$ is thus divided into B subintervals, $[t_a^j, t_b^j]$, and ρ_k^j is defined as $\rho_k(t_a^j, t_b^j)$ in equation 1, where $j = 1, \ldots, B$.

To convey the position of data in time, the HSV colour space is used with a single hue with a decrease of saturation and value for bins containing older time steps. This has the effect that new data are given a bright colour and older data appear grey and 'washed-out'. The per-pixel mapping of density, ρ, to opacity, α, is calculated as

$$\alpha = 1 - e^{-\rho / \zeta\lambda} \tag{4}$$

ζ refers to the number of data items multiplied by the number of time steps in bin j, thus it varies with each bin. λ is a global scaling factor used to adjust opacity.

The volume rendering integral is applied using front-to-back compositing [EHK*06]. For each pixel, the composited colour, $\boldsymbol{c}_{\text{tot}}$, and opacity, α_{tot}, are calculated as

$$\begin{aligned} \boldsymbol{c}_{\text{tot}} &\leftarrow \boldsymbol{c}_{\text{tot}} + (1 - \alpha_{\text{tot}})\alpha_j \boldsymbol{c}_j \\ \alpha_{\text{tot}} &\leftarrow \alpha_{\text{tot}} + (1 - \alpha_{\text{tot}})\alpha_j \end{aligned} \tag{5}$$

where $\boldsymbol{c}_j$ is the bin colour and α_j is the per-pixel opacity for bin j. It is possible to view a time period from both the end and the beginning. Viewing it from the end puts new data in the foreground (figure 5(a)). Viewing it from the beginning places older data in the foreground (figure 5(b)). Regardless of view direction, new values are always painted with the highest possible value and saturation. Originally it was intended to use colour change as aerial perspective but it has been found that the ability to maintain a consistent relationship between colour and time was more useful and that the natural link between vibrant colours and newness and washed-out colours and age was more significant.

3.3. Interacting with Large Data Sets

When moving a time window (such that only information between time steps t_a and t_b, where $t_a < t_b$, is included in the visualization) forward in time using TDPC, only two updates are necessary for each new frame; to remove information for time step t_a and to add information for time step t_{b+1}. For DCPC the same process needs to be applied for each bin. The texture-based approach, described below, makes it possible to combine these many operations into only a small number of rendering passes.

4. Implementation

All of the techniques presented in this paper have been implemented using C++, OpenGL, Frame Buffer Objects (FBOs) and the OpenGL Shading Language (GLSL). The implementation has been carried out on a laptop with a 2 GHz Intel processor, 1 GB of RAM and an NVIDIA GeForce Go 6800 graphics card. The graphics card only supports 16-bit floating point textures and blending. This allows the representation of integer values ranging from -2047 to 2048 in each RGBA texture component and the smallest number that can be added without exceeding the precision limit is 1.0. FBOs are used in order to render directly to textures, thus avoiding texture copying.

For TDPC, primitives are rendered to colour channels of a number of RGBA textures. Content is added by means of additive blending and removed using subtractive blending. In order to ensure a correct density map a normalization is performed at each visualization update by calculating $\tilde{\rho} = \rho/\max(\rho_k)$. A fragment shader is used to composite the contents of the textures used into the final rendered image and for applying TFs.

For DCPC time steps are distributed with respect to time into B colour channels. In this particular implementation, B is set to eight which corresponds to two RGBA textures. This has proven to provide a sufficient number of distinct saturation and value levels in order to discriminate the position of data in time and still provide an interactive visualization. For some applications this number may be smaller or larger. Having an exactly correct density map in DCPC is not an absolute requirement in order to perceive the position of data in time. Each bin is therefore normalized by its total number of data items, a significantly faster operation compared to finding the maximum density value. To fine-tune the normalization a user-controlled constant (λ in equation 4) is added, which can be changed by means of a slider. The volume rendering integral is calculated in a fragment shader to composite the contents of all colour channels into the final image. When moving a time window, the bins need to be updated. Moving content from one colour channel to another is done in a single pass. Each texture is connected to an FBO and moving content between two FBOs requires an extra rendering pass. In general the maximum number of rendering passes required is $U = B + \Omega$, where Ω is the number of textures used.

If two variables mapped to adjacent axes both have a positive or negative change then a quadrilateral is rendered (figure 2(c)). For cases where there is a positive change in one variable and a negative change in another, a quadrilateral cannot be used since non-convex quadrilaterals cannot be guaranteed to work correctly in hardware rendering. In this case a quadrilateral is rendered as two triangles (figure 3).

The parallel coordinates axes are, for both TDPC and DCPC, expanded horizontally in order to better reveal changes occurring within individual variables. In this way, data items and changes between time steps are also seen, not only their relationships to each other. This can be seen in figures 6(c)–6(g).

5. Results

The first example in this section illustrates the use of TDPC and DCPC when analysing a large, time-varying multivariate data set. The second example shows how the visual results of rendering lines and polygons are affected by the size of temporal changes.

5.1. Interactive Analysis of Temporal Information

In section 1 a time-varying multivariate data set was defined as being one containing one or several data items, $\boldsymbol{d}_m$, for each of a number of time steps, t_i, where each $\boldsymbol{d}_m$ contains values for at least two variables. A synthetic data set, following this definition, was used for an initial assessment. The data set contains 100 data items, each containing values for six variables at each of 500 time steps. A synthetic data set was used since clearly defined patterns can be constructed and visually examined. Real-world examples of such a data set are, for example, multichannel EEG data, social science data, energy data and climate data.

For DCPC and TDPC two RGBA textures of size 512×256 were used. For DCPC eight bins were used. The visualization, together with a number of interaction sliders, was rendered in a display window having a resolution of 1024×1024.

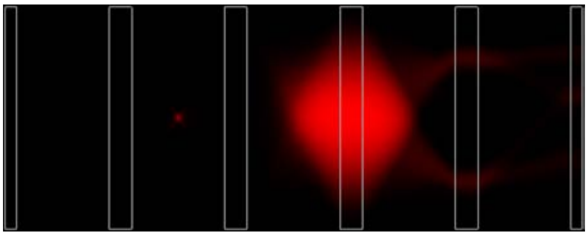
Standard parallel coordinates can be used to visualize time-varying multivariate data sets by superimposing all parallel coordinates for all time steps. An additional time axis can be added (figure 6(a)) and used to select a subset of the time period (figure 6(b)). For a large, time-varying multivariate data set this does not, however, reveal any temporal structure due to cluttering artefacts.

In TDPC a density map is generated so TFs can be used to investigate both high and low density regions. By focusing on high density regions it can be seen in figure 6(c) that the third axis from the right and a point between the second and third axes from the left show the highest density values. The variable mapped to the third axis from the right contains the highest density values and appears to have a Gaussian distribution, therefore it can be expected to contain only noise. Using a TF to focus on low density regions reveals additional structures. Strong correlations are seen between the second and third axes from the left and some possible clusters between the two rightmost axes (figure 6(d)).

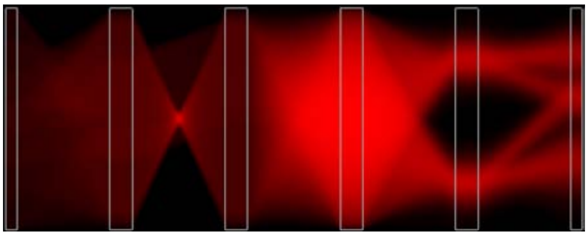
DCPC can be used to reveal information about where in time changes occur. It is possible to view the time period both from the end and the beginning without having to completely re-render the data. The only required change is the order of bin compositing and colouring in the shader and hence, switching between the two viewing directions is done instantaneously. By analysing both viewing directions a number of temporal patterns can be discerned, see figures 6(e) and 6(f). First, the leftmost variable shows a clear periodic pattern with a decreasing amplitude as time progresses.

(a) Standard parallel coordinates with an additional time dimension (leftmost axis) is used to analyse the entire time period.

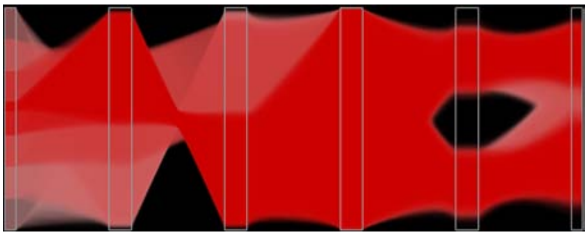
(b) Standard parallel coordinates using a time axis as in (a) but now only showing a subset of the time period.



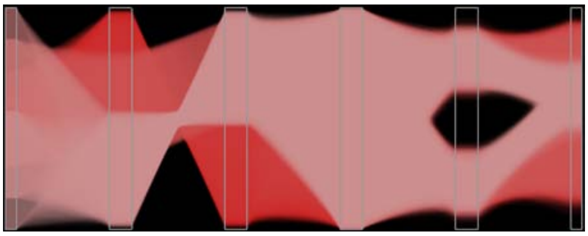
(c) TDPC, rendering polygons, is used to investigate where the maximum amount of change, resulting in the highest densities, occur.



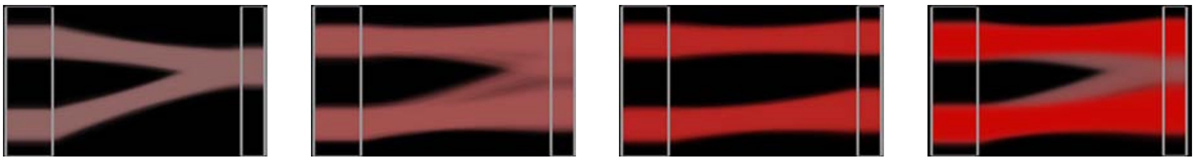
(d) TDPC, rendering polygons, is used to emphasize low density regions, revealing additional structures.



(e) DCPC, rendering polygons, is used to view the entire time period. Changes occurring at recent time steps (at the end of the time period) are placed in the foreground and rendered using maximum saturation and value.



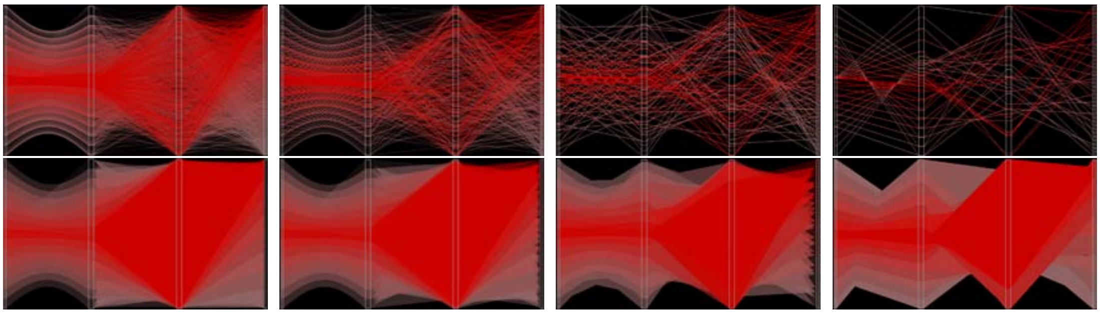
(f) DCPC, rendering polygons, is used to view the time period from the opposite direction compared to (e). The focus is now on changes occurring in the beginning of the time period which are rendered with the lowest saturation and value.



(g) The same time period as in (b) is analysed using DCPC. The three leftmost images show the content of three of the eight bins used while the rightmost image shows the compositing of all eight bins.

Figure 6: *TDPC and DCPC ((c)–(g)) are used to interactively visualize a large, time-varying multivariate data set containing six variables and 100 data items at each of 500 time steps. The amount of change as well as where in time they occur are revealed.*

Second, the relationship between the second and third axes from the left shows a clear negative correlation in the end and in the beginning of the time period, with a positive correlation in between. Third, the variable mapped to the third axis from the right was previously suspected to contain noise. It is now seen that it contains large changes at both the beginning and end of the time period and, by applying a smaller time window, it can be confirmed that these large changes are present throughout the time period and this variable does, indeed, contain noise. Last, the relationship between the two rightmost axes shows two 2D clusters. Over time the clusters separate and are at the end of the time period separable in both dimensions. A close-up view of the clusters is shown in figure 6(g) (same time period as in figure 6(b)). The three leftmost images show the contents of three of the eight bins



(a) 2000 time steps. (b) 500 time steps. (c) 100 time steps. (d) 40 time steps.

Figure 7: *The effect of rendering lines (top row) versus polygons (bottom row) when the size of changes between time steps increase. For small changes the visual difference between rendering lines and polygons is small (figure 7(a)). For extremely large temporal changes a polygon-based rendering is advantageous since it better preserves structures (figure 7(d)).*

Table 1: *Frame rates of DCPC and TDPC for an idle state and when interacting with the visualization by means of moving a time window. The data set used contains six variables and 100 data items at each of 500 time steps.*

	DCPC	TDPC
Polygons	39 fps	86 fps
Lines	128 fps	128 fps
Idle	200 fps	490 fps

used and the rightmost image shows the full compositing of all eight bins.

The frame rates of DCPC and TDPC for an idle state and when interacting with the visualization by means of moving a time window are presented in table 1. In an idle state no interaction is performed, only the final compositing in the shader.

5.2. Polygons and Lines when Analysing Large Changes

Visualizing changes between time steps, rather than values at each time step, is further explored in this section. It was previously discussed in section 1 that rendering lines in parallel coordinates is analogous to rendering points in Cartesian coordinates and rendering polygons in parallel coordinates is analogous to rendering the line segments between the points.

It is here shown how the visual results of rendering lines and polygons are affected by the size of temporal changes and the difference is illustrated by an example data set describing the change of a single multivariate data item. To simulate large temporal changes, a data set is sub-sampled. Visualizing the entire data set containing 2000 time steps (having very small changes occurring between consecutive time steps) produces similar results either rendering lines or polygons, see figure 7(a). Increasing the size of the temporal changes by only using 25 percent of the time steps affects the line rendering substantially more than the polygon rendering. It is, however, still possible to discern all patterns using both techniques (figure 7(b)). When visualizing as few as five percent of the time steps, the data set now containing very large changes between the time steps, it is more difficult to perceive the periodic change between the two leftmost axes when rendering lines. The polygon-based rendering clearly reveals this (figure 7(c)). Using even fewer time steps (two percent) produces a data set containing extreme temporal changes and it is impossible to distinguish the original patterns using a line rendering. The polygon-based rendering is also affected but retains the main characteristics of the data set, see figure 7(d).

6. Conclusions and Future Work

In this paper the visualization of time-varying multivariate data sets in parallel coordinates has been explored. We have presented the concept of visualizing changes between time steps which is achieved by rendering polygons rather than lines. Our results show that the polygon-based approach is superior to line rendering when analysing data sets containing large temporal changes.

Temporal Density Parallel Coordinates (TDPC) and *Depth Cue Parallel Coordinates* (DCPC) have been introduced. TDPC can be used to interactively reveal the density of a specified time period and is based on a density map that can be efficiently updated. DCPC reveals where in time actual data values or changes occur. It is based on a temporal binning that can be efficiently updated, and employs perception-based colouring and concepts from volume rendering. Both TDPC and DCPC have been implemented on the GPU, allowing interactive visualization of thousands of data items over thousands of time steps, something not possible in previous efforts with parallel coordinates. When interacting with the visualizations by means of a time window, the interactivity is independent of the size of the window.

An application area of particular interest is system identification [Lju99] where results from preliminary model building are studied by the user in order to tune and test the fit of a model structure. For future work it would be interesting to study how the model validation process can be aided by the use of interactive visualization allowing simultaneous analysis of the dependencies between large numbers of variables and time.

Acknowledgements

This work has been funded by the Swedish Foundation for Strategic Research, grant A3 02:116.

References

[Eds03] EDSALL R. M.: The parallel coordinate plot in action: Design and use for geographic visualization. *Computational Statistics and Data Analysis 43*, 4 (2003), 605–619.

[EHK*06] ENGEL K., HADWIGER M., KNISS J. M., REZK-SALAMA C., WEISKOPF D.: *Real-Time Volume Graphics*. A K Peters, Ltd., 2006.

[FCI05] FANEA E., CARPENDALE S., ISENBERG T.: An interactive 3D integration of parallel coordinates and star glyphs. In *Proceedings IEEE Symposium on Information Visualization* (2005), pp. 149–156.

[FWR99] FUA Y.-H., WARD M. O., RUNDENSTEINER E. A.: Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings IEEE Visualization* (1999), pp. 43–50.

[Hin87] HINTERBERGER H.: *Data Density: A Powerful Abstraction to Manage and Analyze Multivariate Data*. PhD thesis, Informatik-Dissertationen ETH, Zurich, 1987. No. 4.

[Ins85] INSELBERG A.: The plane with parallel coordinates. *The Visual Computer 1* (1985), 69–91.

[JCJ05] JOHANSSON J., COOPER M., JERN M.: 3-dimensional display for clustered multi-relational parallel coordinates. In *Proceedings 9th IEEE International Conference on Information Visualization* (2005), pp. 188–193.

[JLJC05] JOHANSSON J., LJUNG P., JERN M., COOPER M.: Revealing structure within clustered parallel coordinates displays. In *Proceedings IEEE Symposium on Information Visualization* (2005), pp. 125–132.

[JLJC06] JOHANSSON J., LJUNG P., JERN M., COOPER M.: Revealing structure in visualizations of dense 2D and 3D parallel coordinates. *Information Visualization 5*, 2 (2006), 125–136.

[Lju99] LJUNG L.: *System Identification: Theory for the User*, 2:e ed. Prentice-Hall, 1999.

[MS03] MÜLLER W., SCHUMANN H.: Visualization methods for time-dependent data - an overview. In *Proceedings 35th Conference on Winter Simulation* (2003), pp. 737–745.

[NH06] NOVOTNÝ M., HAUSER H.: Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 893–900.

[NTPT96] NORMAN J. F., TODD J. T., PEROTTI V. J., TITTLE J. S.: The visual perception of three-dimensional length. *Journal of Experimental Psychology: Human Perception and Performance 22*, 1 (1996), 173–186.

[RWK*06] RÜBEL O., WEBER G. H., KERÄNEN S. V. E., FOWLKES C. C., HENDRIKS C. L. L., SIMIRENKO L., SHAH N. Y., EISEN M. B., BIGGIN M. D., HAGEN H., SUDAR D., MALIK J., KNOWLES D. W., HAMANN B.: PointCloudXplore: Visual analysis of 3D gene expression data using physical views and parallel coordinates. In *Proceedings Eurographics /IEEE-VGTC Symposium on Visualization* (2006), pp. 203–210.

[TAS04] TOMINSKI C., ABELLO J., SCHUMANN H.: Axes-based visualizations with radial layouts. In *Proceedings ACM Symposium on Applied Computing* (2004), pp. 1242–1247.

[TAS05] TOMINSKI C., ABELLO J., SCHUMANN H.: Interactive poster: 3D axes-based visualizations for time series data. In *Poster Proceedings Compendium, IEEE Symposium on Information Visualization* (2005), pp. 49–50.

[tCMR07] TEN CAAT M., MAURITS N. M., ROERDINK J. B. T. M.: Design and evaluation of tiled parallel coordinate visualization of multichannel EEG data. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (2007), 70–79.

[Weg90] WEGMAN E. J.: Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association 85*, 411 (1990), 664–675.

[WL97] WEGMAN E. J., LUO Q.: High dimensional clustering using parallel coordinates and the grand tour. *Computing Science and Statistics 28* (1997), 352–360.

[WLG97] WEGENKITTL R., LÖFFELMANN H., GRÖLLER E.: Visualizing the behavior of higher dimensional dynamical systems. In *Proceedings IEEE Visualization* (1997), pp. 119–125, 533.

[ZLTS03] ZHAO K., LIU B., TIRPAK T. M., SCHALLER A.: Detecting patterns of change using enhanced parallel coordinates visualization. In *Proceedings Third IEEE International Conference on Data Mining* (2003), pp. 747–750.