DOI: 10.1111/cgf.12352

COMPUTER GRAPHICS *forum*
Volume 33 (2014), number 6 pp. 310–321

Visual Analysis of Trajectories in Multi-Dimensional State Spaces

S. Grottel[1], J. Heinrich[2], D. Weiskopf[3] and S. Gumhold[1]

[1]Computer Graphics and Visualization, TU Dresden, Dresden, Germany
{Sebastian.Grottel, stefan.gumhold}@tu-dresden.de
[2]CSIRO, Sydney, Australia
julian.heinrich@csiro.au
[3]Visualization Research Center, University of Stuttgart, Stuttgart, Germany
weiskopf@visus.uni-stuttgart.de

Abstract

Multi-dimensional data originate from many different sources and are relevant for many applications. One specific sub-type of such data is continuous trajectory data in multi-dimensional state spaces of complex systems. We adapt the concept of spatially continuous scatterplots and spatially continuous parallel coordinate plots to such trajectory data, leading to continuous-time scatterplots and continuous-time parallel coordinates. Together with a temporal heat map representation, we design coordinated views for visual analysis and interactive exploration. We demonstrate the usefulness of our visualization approach for three case studies that cover examples of complex dynamic systems: cyber-physical systems consisting of heterogeneous sensors and actuators networks (the collection of time-dependent sensor network data of an exemplary smart home environment), the dynamics of robot arm movement and motion characteristics of humanoids.

Keywords: visualization, continuous-time, parallel coordinate plots, scatterplots

ACM CCS: I.3.3 [Computer Graphics]: Picture/Image Generation Line and curve generation I.3.8 [Computer Graphics]: Applications Visualization

1. Introduction

Multi-dimensional trajectories and time-series play important roles in many applications. One source such data comes from are multi-dimensional state spaces. Value changes within such state spaces can be seen as continuous trajectories. Instead of treating such data as unstructured collection of sample points, trajectories, similar to time-series, represent different data values of one single entity—i.e. one system—at different points in time. In general, a data set is not limited to a single entity, but for the sake of simplicity we restrict our work to this case. Comparing the concept of trajectories with time-series, we want to define an important difference in the context of this work. Time-series, such as financial stock data, do not necessarily allow for interpolation as the underlying models might not fulfill the necessary preconditions. We understand trajectories as samples at different points in time along a continuous path within multi-dimensional spaces, thus allowing for interpolation between these samples as part of trajectory reconstruction.

Sources of multi-dimensional trajectory data are the state space of complex networks of sensors and actuators, e.g. used for environment automation. Such systems as well as the accompanying computation technologies pervade our daily life, often completely concealed. The paradigm of the *Internet of Things* [AIM10] follows from the idea of interconnecting elements, such as wired and wireless sensors and actuators, with intelligent communication infrastructure. One instance of this paradigm is the idea of *cyber-physical systems* (CPS) [Poo10] which relate such physical elements to their virtual counterparts. Although, each element acts only locally, i.e. sensing or manipulating the physical world through the intelligent interconnect, the mutual monitoring and controlling, the system as a whole can perform complex tasks safely and efficiently. Concrete implementations of these paradigms are contemporary smart structures, such as smart grid, smart factories and smart homes. Interactive visualization of states of such systems as a whole can aid understanding the complex interactions and the emerging system behaviour. Especially, the scenario of smart homes is highly challenging because of the extreme diversity of sensors and actuators, as well as because this scenario exhibits the most dynamics in terms of reconfiguration of the infrastructure, i.e. removing, repositioning and adding system elements [FSS13]. Among others, smart home systems are comprised of environmental sensors, such as temperature, ambient light, air humidity and barometric pressure. Actuators range from simple elements, such as motorized valves at heaters or window openers, to complex platforms such as service robots for transportation and interaction tasks.

Data in such an application scenario are highly heterogeneous and dynamic, e.g. different types or numbers of sensors may be introduced into the system. Therefore, our design focuses on a generic approach and a wide applicability to different data. We understand our visualization design as a first-step exploration tool, to gain overview of the data and some detail information, as precursor to further analysis. We create three visualizations based on this idea (cf. Figures 1 and 7): Continuous-time scatterplots (Figure 7, left), in particular continuous-time scatterplot matrices (CSPLOM), allow us to visually analyse correlations between dimensions and to select potentially interesting subsets of dimensions. Continuous-time parallel coordinates plots (CPCP) (Figure 1, bottom) represent the multi-dimensional data in a compact form providing a good overview. The novel key element of these plots is that they fully incorporate continuous temporal interpolation, reflecting the data characteristics of trajectories. We supplement these plots by a temporal heat map (THM, Figure 1, top) showing the temporal evolution of the data directly. In this way, the different kinds of plots provide complementary information to cover all aspects of the data. These visualizations are shown in coordinated views, with synchronized dimension axes (for CPCP and THM) and highlighting a selected point in time in all plots (cf. Figure 9).

Our main contribution is the theoretical foundation for continuous-time scatterplots and CPCP, leading to respective and practical visualization techniques. Based on physically motivated density and mass conservation, which has been discussed before for spatially continuous data only (cf. [BW08] and [HW09]), we aggregate the trajectory data over time. This approach is directly designed for trajectories in multi-dimensional spaces. We show the usefulness of our method with several case studies, using coordinated views of our continuous plots and THM, directly showing the evolution of the data.

2. Related Work

There are many approaches to the visualization of multi-dimensional data in general and to the visualization of time-dependent data in particular. Surveys and books exist that provide a good entry point into these topics, e.g. [WB94], [Kei02], [AMM*08], [AMST11]. In addition to generic concepts, application-specific solutions exist that provide specialized visual metaphors. With our work, we aim at generic multi-dimensional visualization of trajectory data.

Multi-dimensional data can be visualized following many different approaches [WB94], [Kei02]. Iconic representations such as Chernoff faces [Che73] map dimensions to attributes of a glyph. Others (e.g. the radviz method [HGM*97]) employ geometric transformations to the input data or represent data values as pixels [Kei02]. Further methods use axis-aligned 2D subspaces to visualize pairwise 2D projections of the data such as scatterplots and scatterplot matrices [Har75] (SPLOM). Keim *et al.* [KHD*10] combined the scatterplot concept with pixel-based visualization, to optimize the visual information by a trade-off between strict placement and overdraw. While scatterplots are typically used for the visualization of 2D data, a SPLOM represents all pairwise relations of a data set in a matrix of scatterplots. The most common use of a SPLOM is to provide an overview of the data and allow the user to navigate multi-dimensional data sets [EDF08].

Another method of pairwise 2D projections of the data is the use of parallel coordinates plots [Ins85] (PCP). In traditional PCPs, multi-dimensional points are mapped to polygonal lines intersecting a set of parallel axes that represent dimensions, allowing one to visually trace the values of a point in multi-dimensional space [KZZM12]. However, for large data sets, parallel coordinates tend to clutter the view due to potential heavy overplotting of lines. This has been addressed using various approaches; see [HW13] for a recent survey on parallel coordinates and techniques that address the issue of cluttering. One approach uses brushing and colours to emphasize data according to a given classification scheme. In this work, however, we do not have any *a priori* classification of the data. Alpha blending [LS09], [DHNB09] can be used to visually separate dense from sparse areas. However, alpha blending distorts the distribution of lines and is dependent on the drawing order. These problems are avoided by binning and other approaches based on counting lines [AdOL04], [MW91], [FKLI10]. The concept of line density is also present in other visualization applications, e.g. for stream line visualization of vector fields [KLG*13]: density functions of overlapping lines are accumulated by using additive blending. This follows the idea of density, but all lines are equally weighted, independent from the movement speed along these lines. Density images for scatter plots and PCPs (using Hough space transformation) can be employed to automatically assess the quality of the respective visualizations [TAE*11].

However, none of the above techniques assumes the multi-dimensional input data to be given on a continuous domain. This problem is addressed by Bachthaler and Weiskopf [BW08] providing an analytical solution to the computation of the corresponding density in scatterplots, which can also be applied to parallel coordinates [HW09]. The continuous model allows for the extraction of critical points [LT10], other features [LT11], the design of transfer functions [GXY11], [WZL*12] and can be computed efficiently [BW09], [HBW11]. Recently, Molchanov *et al.* [MFL13] have generalized the approach of the continuous data domain. Their data are given by topology-free, multi-dimensional particle data from SPH simulation and they present the method for generating continuous star splots. All the above approaches, however, are based on the continuous spatial data domain. We adopt the original approach [BW08], [HW09] and present a model for the computation of densities in both scatterplots and PCPs from a continuous trajectory that we obtain by interpolation in the temporal domain.

Our model is based on mass conservation and thereby reflects the discrete point-based rendering (for scatterplots) and line-based rendering (for parallel coordinates) in the limit process of increasing sampling rate and, thus, sampling quality. Noting that variation integrated over time is lost when using the traditional line-based approach, Johansson *et al.* [JLC07] describe one of the few previous approaches for parallel coordinates that explicitly take into account the continuity of time for such time-dependent multi-dimensional data. Their method used polygons with a fixed density for

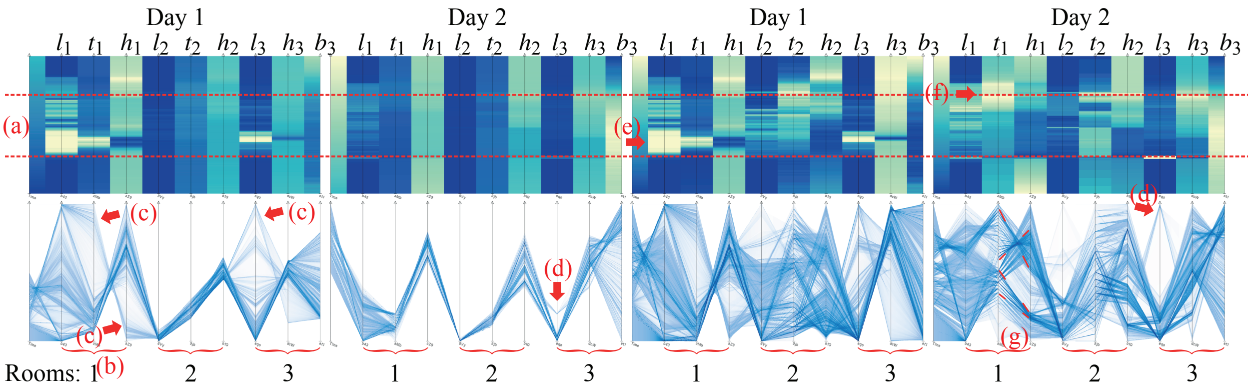

Figure 1: *Temporal heat map (THM) and continuous parallel coordinates plots (CPCP) of sensor data collected over two full days: l_i denotes ambient light sensors, t_i temperature sensors, h_i humidity sensors and b_i barometric pressure sensors. The left diagrams use shared scaling, making values across axes of the same type and across both days comparable. The right diagrams use independent scaling to show small changes. The annotations (a)–(g) mark interesting observations (cf. Section 4.1).*

consecutive points in time; accumulating densities then separates regions of high variation from regions of low variation. However, their approach does not visualize correlations between dimensions originating from the data behaviour along a time step, because they use a fixed-density model does not reflect the point–line duality between Cartesian and parallel coordinates and its strong effect on density. In contrast, our new model of continuous-time parallel coordinates fully includes those aspects; therefore, it is the first technique to deliver consistent results, regardless of the temporal sampling rate.

Wegenkittl *et al.* [WLG97] used extruded parallel coordinates to visualize multi-dimensional dynamical systems (note that the authors use the term *high-dimensional*, but use data sets comparable to our data in terms of numbers of dimensions). In their approach, the parallel coordinates system is moved along the z-axis to encode time, such that a 3D PCP is constructed. While their approach uses parallel coordinates for visualization, we do not rely on 3D rendering due to the additional ambiguity introduced by projection.

Other previous papers on PCPs for temporal data [BBP08], [tCMR07] differ from continuous-time PCPs because they use animation to show time [BBP08], to show differences between successive time points [BBP08], or they use frequency plots only along individual data axes but not between two axes [tCMR07]. However, the heat map representation of electroencephalography data [tCMR07] is a typical example of previous work on frequency plots for time-dependent multi-dimensional data. The design of our THM follows this known strategies, modifying specific aspects of the visual design to allow for easy integration in our multiple views setup.

There are many ways to visualize time-dependent data in general. While some traditional time-dependent visualizations, such as line graphs, use interpolation for values between adjacent points in time, they are either not suited for the visualization of multi-dimensional trajectories or they are explicitly designed to search for events occurring at a specific point in time. Hence, for the visualization of the overall correlation between multiple variables, i.e. integrated over time, such visualizations are not suited. Thinking of multiple trajectories as a multi-dimensional data set, axes might also represent one time-dependent variable each, such that one polygonal line is obtained for every point in time. This enables the visualization of relations between variables over *periods of time*. A similar approach can be used for air-traffic control [Ins01], [Ins09], parameter trajectories for facial dynamics [TFA*11] and the visualization of trends in PCPs [LS09]. However, none of these examples employ a density-based approach to the visualization of multi-dimensional time-oriented data.

3. Visualization Techniques

Our visualization is composed of coordinated views showing different representations of a multi-dimensional data set. The first one is a CSPLOM showing the pair-wise relations between dimensions. The second visualization is the highly related CPCP to show correlations among all dimensions. Both plots support each other nicely, because the CSPLOM allows the analyst to observe interesting dimension relations, which can be used to derive subsets and ordering of dimensions for the CPCP. Because of the density-based model, the CPCP present the value distribution on each dimension, allowing one to spot outliers or value clusters. The third visual representation is a THM. Both, the CSPLOM and the CPCP do not visualize time directly, due to the integration over time. The THM uses parallel dimension coordinates, such as the CPCP, but represents values by colour, as all axes represent time.

Our visualization system combines all three representations in the form of coordinated views: All views show the same data (or subsets of the same data). The THM and the CPCP can share the same dimension axes, i.e. using the same subset of dimensions and the same ordering. If this is the case, then both plots can be shown atop of each other and zooming and (horizontally) panning is synchronized in these views.

Another coordination feature, which applies to all views, is the synchronized selection of a point in time within the data. The whole trajectory can be viewed as *animation*, in which case this selected point in time is changed automatically. We highlight the currently selected point in time in the views using a line of user-defined colour (cf. Figure 9). Here, we use red lines because this colour is not used in the colour maps in the figures of this paper. The point in time is shown as dots in the CSPLOM. In the CPCP, the values of the currently selected point in time are highlighted by a corresponding poly-line, similar to the classical PCP generation. In the THM, a point in time is just a horizontal line.

To emphasize the direction of change over time, we extend the highlight of the point in time, following the idea of afterglow. While the benefit is minimal for the THM, the direction of change is an important information for the CSPLOM and the CPCP (cf. Figure 9). The length of the afterglow polygon in terms of past time steps can be freely defined by the user or disabled completely.

We implemented the rendering of our plots in C++ with OpenGL using GLSL shaders to harvest the processing power of programmable GPUs and to reach interactive frame rates. Most elements of the different plots are directly evaluated in the fragment shader stage, e.g. the density information for the CSPLOMs and CPCPs is computed by additive blending in float frame buffers. A full view of a data set from the motion capture database (cf. Section 4.3), containing 62 degrees of freedom and more than 2 700 time steps, renders at more than 90 FPS at a high resolution of 3500 × 1000 pixel (GPU: NVidia GeForce GTX 680).

3.1. Continuous-time scatterplots

We adopt the model of continuous scatterplots [BW08] and continuous parallel coordinates [HW09] for multi-dimensional trajectories. Bachthaler and Weiskopf [BW08] describe a generic mathematical model for the transformation of a density s in the *spatial domain* to a density σ in the *data domain* under the assumption of mass conservation (i.e. the total mass does not change under the transformation):

$$\int s(t)\mathrm{d}t = \int \sigma(\xi)\mathrm{d}^2\xi. \tag{1}$$

Based on continuous scatterplots, Heinrich and Weiskopf extend this model to compute a point-wise density φ in the 2D *parallel-coordinates domain* [HW09], which is the subject of Section 3.2

While the model of continuous scatterplots and continuous parallel coordinates is generic, previous work focused on the transformation of data from a 3D spatial domain to the data domain and the parallel-coordinates domain. To cover the problem of trajectory visualization, we treat the *temporal* domain of the time-series analogous to a 1D *spatial* domain from which we compute densities in scatterplots and parallel coordinates.

With t describing time, let $\tau(t) = (\tau_1(t), \tau_2(t), ..., \tau_m(t))^\mathrm{T}$ denote an m-dimensional trajectory in the temporal domain. We start with trajectories sampled at discrete time points t_j. Figure 2 (top) illustrates $\tau(t)$ in the discrete setting for all domains. Note that for every time point, the m attached data values form a single m-dimensional point in the data domain (or a single line in the parallel-coordinates domain, cf. the red points or lines in Figure 2).

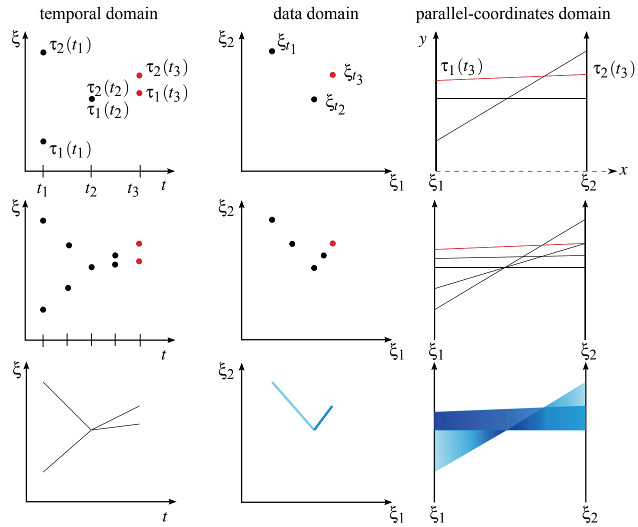

Figure 2: *Representation of multi-dimensional trajectory data in the temporal domain (left), the data domain (centre) and the parallel-coordinates domain (right). For every time point t_i, m attributes are mapped to points $\xi_{t_i} = \tau(t_i)$ in a scatterplot and polylines in parallel coordinates. Attributes or dimensions τ_i are represented as axes ξ_i. Resampling between t_j and t_{j+1} with linear interpolation results in lines with constant density in the scatterplot (middle), whereas the vertical density of lines in parallel coordinates changes with respect to the horizontal position x. Our density model (bottom) correctly reflects this density.*

The second row of Figure 2 shows the same data with values sampled at three additional points in time (obtained by linear interpolation). The corresponding points in the data domain are mapped to a curve (or a line for linear interpolation) intersecting $\xi_{t_1} = (\tau_1(t_1), \tau_2(t_1))^\mathrm{T}$ and $\xi_{t_2} = (\tau_1(t_2), \tau_2(t_2))^\mathrm{T}$. It is important to note, however, that the underlying model supports any other interpolation scheme as well. For our data, we found that linear interpolation is sufficient to reveal interesting patterns in the aggregated views, while keeping the mathematical model and implementation simple. This is mostly due to the high number of samples in our application, such that the difference to plots created with higher-order interpolation schemes becomes negligible.

Since τ is a mapping from a 1D temporal domain to a 2D data domain, the support for the resulting density σ in the scatterplot is a 1D curve C. To compute σ, we split the 2D integration domain of (1) in two parts: (i) along the resulting curve C and (ii) the perpendicular space $C_\perp$ around that curve:

$$\int \sigma(\xi)\mathrm{d}^2\xi = \int_C \left[\int_{C_\perp} \delta_\perp(\tilde{\xi})\mathrm{d}\tilde{\xi}\right] \hat{\sigma}(\hat{\xi})\mathrm{d}\hat{\xi}, \tag{2}$$

where $\delta_\perp$ is a 1D delta function (more precisely, a delta distribution) on $C_\perp$. This generalization is required because the support of σ is just a 1D curve within the 2D scatterplot, i.e. a null set; any integration over a function with the support of a null set vanishes.

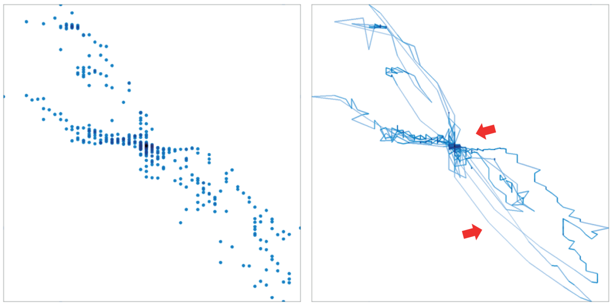

Figure 3: *Traditional density-base scatterplot using discrete points (left) compared to a continuous-time scatterplot (right) integrated over time (trajectory). Only with the continuous-time scatterplot, the path of interpolated values is visible through the continuous representation (lower red arrow) and regions with high density clearly denote prevalent values (red arrow at centre).*

With (2), however, the 2D spatial integration yields finite values. Therefore, the 1D density $\delta_\perp$ plays the crucial role in constructing the continuous-time scatterplot. For background reading on distributions as generalized functions, we refer to the textbook by Griffel [Gri03].

By employing the transformation theorem for integrals to the outer integral of (2), mass conservation for the time interval $[t_j, t_{j+1}]$ can be stated as

$$\int_{t_j}^{t_{j+1}} \hat{\sigma}(\mathbf{r}_j(t)) \left|\mathbf{r}'_j(t)\right| \mathrm{d}t = \int_{t_j}^{t_{j+1}} s(t)\mathrm{d}t, \qquad (3)$$

where $\hat{\sigma}$ is the density in the data domain on the curve C_j and $\mathbf{r}_j : [t_j, t_{j+1}] \longrightarrow C_j$ is a parametrization of that curve. With linear interpolation, C_j is the line defined by ξ_{t_j} and $\xi_{t_{j+1}}$ and we only need to integrate the segment between those points and solve (2) for $\hat{\sigma}$ to obtain the density contribution of this line in the data domain:

$$\hat{\sigma}_j = \frac{s_j}{||\xi_{t_{j+1}} - \xi_{t_j}||}, \qquad (4)$$

with the assumption of constant density s_j in the time interval $[t_j, t_{j+1}]$. Note that $s(t)$ describes the relative importance of samples in the temporal domain and thus can be set to $s(t) = 1$ for uniform sampling of time points. However, the model is generic and any other density distribution can be used instead. The bottom row of Figure 2 illustrates the density representation in all domains.

Due to the linear model of (1), the overall density is then obtain by scattering all time intervals to the data domain and summing over all density contributions at the respective points. A comparison of a discrete and continuous-time scatterplot using (4) is given in Figure 3. The classical density-based scatterplots (left) can use point splats (box function). Sample accumulation is thus visible independent from the actual pixel resolution of the visualization. Continuous-time scatterplots (right) still show these density variations. Furthermore, the path of interpolated values is visible—an information that is easily lost in traditional scatterplots. Our method visualizes the resulting scatterplot for the limit process of continuously increasing the temporal sampling. The visualization is thus consistent with our further representations of CPCP and THM. Additionally, the temporal behaviour of the data, i.e. connections of consecutive sample points, is also visible.

Since a single scatterplot visualizes only two out of m dimensions, we added a scatterplot matrix (Figure 7, left) to our system in order to present all pairwise correlations in a single visualization. This is especially useful for interactively selecting dimensions of interest and defining the order of axes in parallel coordinates.

3.2. Continuous-time parallel coordinates

In parallel coordinates, data dimensions ξ_i are represented by vertical axes such that every point $\xi_{\mathbf{t_j}} = \tau(t_j)$ of the trajectory generates a poly-line. As illustrated in Figure 5, a discrete mapping may lead to the same visual encoding for different trajectories. Assuming interpolation between points in time, however, we can use our density model to disambiguate those similar cases.

Similar to the previous section, we employ a mass conservation model for PCPs [HW09] to compute a point-wise density $\varphi(x, y)$ in the parallel-coordinates domain. Here, the model is based on the observation that *line density* is determined by counting lines along the vertical y-coordinate of the PCP. As a consequence, for every horizontal location x in the parallel-coordinates domain, the integral of the density $\varphi(x, y)$ over y has to result in the same mass. Without loss of generality, we defined the density formula $\varphi_j(x, y)$ in the time interval $[t_j, t_{j+1}]$ and construct the overall density function as

$$\varphi(x, y) = \sum_j \varphi_j(x, y). \qquad (5)$$

Similar to (1), mass conservation can be stated as

$$\forall x : \int_{y_j}^{y_{j+1}} \varphi(x, y)|\mathrm{d}y| = \int_{t_j}^{t_{j+1}} s(t)\mathrm{d}t, \qquad (6)$$

with y_j being the y-coordinate of the poly-line generated by $\tau(t_j)$ at horizontal location x. In parallel coordinates, the y-coordinate can be computed as

$$y(t, x) = (1 - x)\tau_1(t) + x\tau_2(t), \qquad (7)$$

with $x \in [0, 1]$ between axes representing τ_1 and τ_2. To compute $\varphi_j(x, y)$ from $y(t, x)$ and to guarantee mass conservation, the derivative of y with respect to t is required:

$$\frac{\partial}{\partial t} y(t, x)|_{[t_j, t_{j+1}]} = \frac{\Delta y_j(x)}{\Delta t_j} = \frac{y(t_{j+1}, x) - y(t_j, x)}{t_{j+1} - t_j}. \qquad (8)$$

Using this derivative for variable substitution in (6)

$$\forall x : \int_{t_j}^{t_{j+1}} \varphi(x, y)\frac{|\Delta y_j(x)|}{\Delta t_j} dt = \int_{t_j}^{t_{j+1}} s(t)dt, \qquad (9)$$

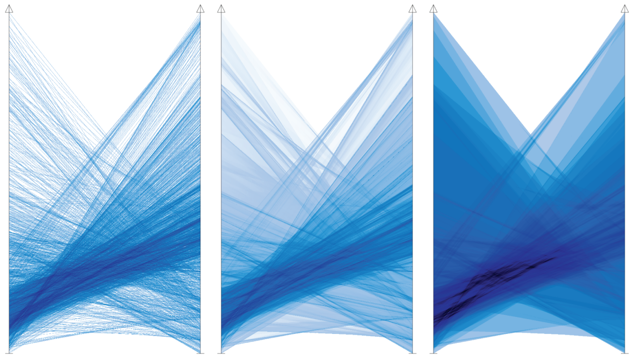

Figure 4: *The resulting images from different approaches to PCP rendering for two example dimensions. Left: the classical line-based approach with a highly sampled trajectory (2700 samples), centre: the CPCP using our model described in Section 3.2, right: the CPCP as presented by Johansson [JLC07]. In the limit process of sampling more and more lines using linear interpolation, the traditional plot converges to our continuous model in the centre.*

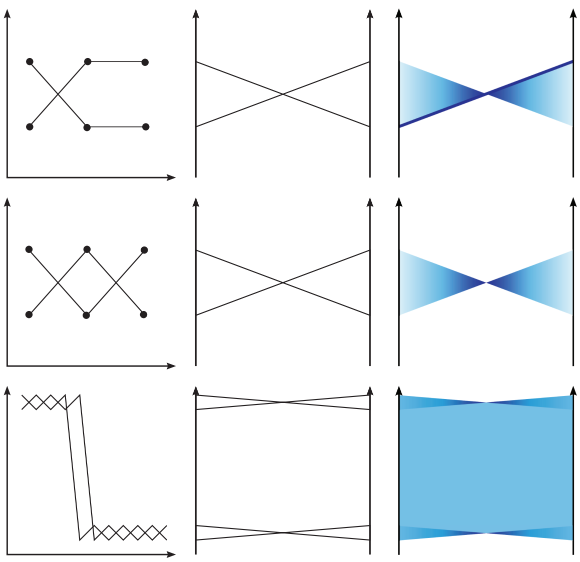

Figure 5: *Sketch of ambiguous cases for discrete parallel coordinates. The top and middle time-series are mapped to the same visual encoding in discrete parallel coordinates (centre column). Using a density representation, these cases are disambiguated (right column). CPCPs further allow us to visualize local and global patterns simultaneously, as the bottom example illustrates. Here, two local negative correlations and one global positive correlation are contained in the time-series.*

finally yields the formula for the density in the CPCP:

$$\varphi_j(x, y) = \frac{\Delta t_j}{|\Delta y_j(x)|} s(t). \tag{10}$$

A comparison between a classical PCP and our CPCP is shown in Figure 4. The data set shown contains more than 2 700 samples along

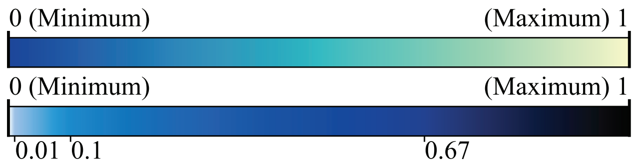

Figure 6: *The colour maps used in all of our visualization screen shots. Top: the colour map for the THM (from ColorBrewer [HB03]: sequential YlGnBu). Bottom: the colour map for the PCP. Due to the normalization in the density-based approach, the maximum values are only reached in very few regions. To be able to distinguish more values, additional interpolation colours are needed at low density (left).*

the trajectory. Note that the classical PCP and our model yield very similar results, as the discrete version approximates continuous-time parallel coordinates in the limit of an infinite number of samples [HW09].

The usefulness of a visualization using density-based PCPs depends on the selected colour map. In our implementation, the colour map can be freely and interactively defined by the user. This allows the user to choose a colour map optimal for representing the features of the corresponding data set being visualized. For our presentation, we use a colour map generated by ColorBrewer [HB03]. Figure 6 shows the colour maps we used for our visualizations in this work. The lower map is used for all PCPs (except for the sketches in Figures 2 and 5). The colour map is inspired by the density of ink in hand-drawn plots, thus interpolating from white (minimum density) over blue to black (maximum density). From (10) follows that the density depends reciprocally on the difference in height between the lines corresponding to two adjacent samples. We chose a compromise between avoiding numerical problems and preserving the disambiguity feature of CPCPs by restricting the density to a maximal value and by choosing our non-linear colour map. This second colour map can also be freely defined and adjusted by the user.

3.3. Temporal heat map

The third view of our visualization is a THM, based on the well-known heat map concept [WF09], which represents data values by colour (often based on cool-warm shading). In our implementation, this representations shares the idea of parallel vertical axes for each dimension with PCP. However, unlike the PCP the vertical direction always depicts time. The values in each dimension along these axes are shown by colour using the user-defined colour map (cf. Figure 6). We chose a colour map from ColorBrewer that is suitable for printing and colour blind users. However, any other colour map can be used to better adjust to specific data. The rationale for the design of the THM is to compensate the lack of explicit information about time in the CSPLOMs and CPCPs. In the THM, the values for each dimension for each point in time can be seen.

The explicit representation of time allows one to quickly grasp temporal evolution of the data. For example, it is possible to distinguish between fast and slow changes based on the size of the area of a colour change (cf. e.g. Figure 1). Easily distinguishable colours even allow us to qualitatively compare values across several dimensional axes (cf. e.g. Figure 10). While the value distribution over the whole data set is explicitly visible in the CPCPs, the information of value changes behaviour is explicitly shown in this THM. Thus, all three visualization complement each other.

4. Case Studies

CPS can basically consist of two distinct types of elements: sensors, monitoring the physical world, and actuators, manipulating real objects. The number and type of these elements are highly heterogeneous and dynamic throughout the system, e.g. sensors and actuators may be added, removed or repositioned. We investigated three test cases. The first one mimics a smart home by an experimental installation of environmental sensors. We integrate concepts of service robots for our further test cases. While the ultimate goal of humanoid, multi-purpose service robots is not implemented, we use installations on smaller scale to investigate individual elements. For evaluation in the context of this paper, we use a robotic arm, equipped with force sensors at the joint motors (cf. Figure 7). However, we do not want to limit our visualization to concrete setups. To look at data beyond our currently available hardware installations, we chose to use motion capture data from human subjects as third test case towards the concepts of humanoid service robots.

4.1. Sensor network data

For our research project, we set up a CPS consisting of several wireless sensors distributed in several rooms of our lab, modelling a smart home environment. These sensors are automatically integrated and managed by a middle ware [FSS13]. To evaluate our visualization, we used data from nine sensors in three rooms collected over 48 h (cf. Figure 1). Each room was equipped with one ambient light sensor (l_1, l_2 and l_3) and one humidity sensor (h_1, h_2 and h_3). Additionally, two rooms were equipped with temperature sensors (t_1 and t_2), while the third room was equipped with a sensor for barometric pressure (b_3). All three sensors in each room were bundled and placed at the same location. Rooms 1 and 3 have windows facing east, while room 2 has windows facing west. The sensors' data were sampled once per second, filtered (eliminating noise), and aggregated to one value per minute, resulting in 1 440 samples per day. Data from some sensors show resolution problems, e.g. t_1 and t_2, which is clearly visible in the CPCP, as shown in the right diagrams in Figure 1. To provide a frame of reference, the dashed lines (a) roughly indicate the times for sunrise and sunset.

The diagrams on the left of Figure 1 use the same scaling for axes of the same type for both days, making all values comparable. Here, two interesting observations are marked: The temperature t_1 and humidity h_1 in room 1 show strong negative correlation, visible both in the CPCP and THM (b). This seems obvious, because humidity should decrease in a room when the temperature increases. However, this correlation is coupled to extreme values on the respective axes only (c) which occurred only in rooms 1 and 3. The corresponding peak values are visible in the THM early after sunrise. An explanation is that early that day the sun was shining brightly into the rooms, which is also indicated by the peak values of ambient light (e), raising temperature and lowering humidity. The effect is weaker in room 3 because here shutters of the windows were partially closed. A similar event occurred at the second day but is only visible by small value peaks (d). This is most likely because on the second day the sky was clouded. Therefore, the values of the light sensors do not show peak values as prominent as on the first day. Instead, unexpectedly high values for light and temperature can be seen late on the second day (f), which can be explained by artificial lighting.

In general, correlations between different sensors in the rooms are often not evident, except for the aforementioned event. For example, correlations among the sensors in room 2 are not clear at all, looking at the left diagrams. Using independent scalings for each dimension (right diagrams), the CPCPs show correlations for parts of the data. However, t_1 and h_1 exhibit positive (lower lines) as well as negative (upper lines) correlations visible by bundles in the CPCP (g). Similar bundles can be seen between the dimensions of the sensors of the other rooms, but were not highlighted to not overload the diagram with annotations.

These are some observations and interpretations of the combined visualizations of these data sets. Of course, more events are present within the data and are visible, e.g. minor peak value in l_2 each evening, corresponding to sunlight from the sun at low elevations shining into the rooms, or that the humidities of the rooms 2 and 3 are more closely related than the humidity of room 1. Actually, rooms 2 and 3 are located at the north-end of the corridor, rather close to each other, while room 1 is located at the south end of the corridor.

4.2. Robot arm data

CPS can also contain actuators manipulating the physical world. One of the most complex examples are robotic actuators. In our scenario, we experiment with an robot arm Jaco [jac]. This robot arm has six motorized joints and a three-finger hand. In the simple test case we evaluate, the arm lifts up a bottle of water at one position and puts it down at new position. Eliminating the need for extra sensors, the whole movement was trained beforehand. The visualization of the resulting data can be seen in Figure 7.

The intuitive graspable values (apart from trajectory time) are the positions of the robot hand lifting the bottle, i.e. *PosX*, *PosY* and *PosZ*. However, to understand the strain for the arm from a robotics point of view, the important values are given in relation to the motorized joints. Thus, as a first step, the coordinates of the hand are correlated to the joint positions using a CSPLOM (cf. Figure 7, left). Figure 7(a) shows a strong correlation between the x position and the joint number one. The y position component is positively correlated to the second joint (Figure 7b). Most importantly, the z position, which is the elevation of the hand and characterizes the lifting of the bottle, is mainly correlated to the second joint as well (Figure 7c).

To inspect the joint data (right half of the right diagrams in Figure 7) dimensions of relative position change and forces acting on each joint motor are visualized. These values are expected to be positively correlated, as the forces are measured via the motor's currents, which, in turn, are changed to control the motion of

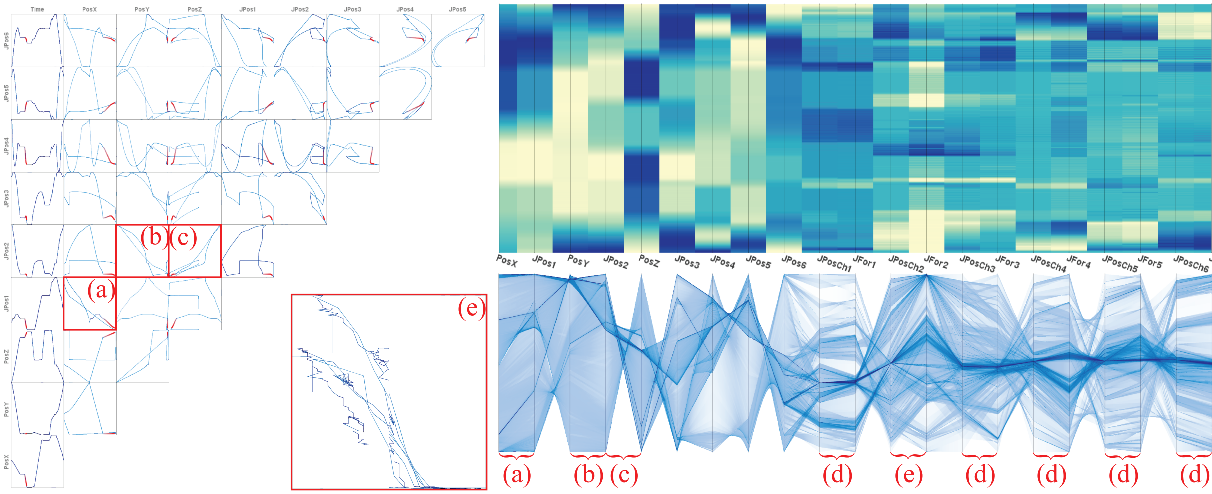

Figure 7: *Visualizations of a trajectory of a robot arm. Left: CSPLOM; top right: THM, bottom right: CPCP. Annotations (a), (b) and (c) show correlations between robot hand coordinates and robot arm joint angles. Annotations (d) and (e) show expected positive correlations between angle changes of the arm joints and forces acting upon these joints. At joint 2 (e), a more complex behaviour is visible (cf. Section 4).*

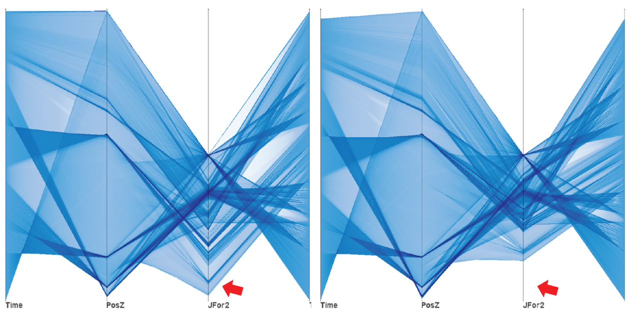

Figure 8: *CPCP of the three dimensions of data set of the robot arm lifting a water bottle (displayed dimensions: time, elevation of hand, force acting on joint 2, time). The visualization of the data set on the right shows an aberration from the other data sets as the forces are not reaching the expected minimum peak.*

the arm. The positive correlations are clearly visible for most joints (Figure 7d) with the exception of the second joint, which shows some disturbances (Figure 7e). Most prominently, some values of *JFor2* seem to be shifted below the usual values (also visible in the corresponding continuous-time scatterplot). Knowing the scenario, this joint is responsible for the force for lifting the bottle during the middle part of the trajectory, as additional work to carrying the weight of the arm itself.

Figure 8 shows the CPCPs of data sets of two trail runs of this scenario, only showing the z position and the force on the second joint, as these dimensions relate to the lifting of the bottle. While the data sets of most trail runs are identical to the data shown in the left diagram, the data set in the right diagram shows an aberration in the force dimension. The negative peak in force when the bottle is lifted up is much higher than expected. This means that the weight the arm is lifting is less than usual. In this specific trail run the bottle was simply missing and the joint was only lifting the weight of the arm itself.

Table 1: *Selected data sets [CMU] for our additional visualization experiments.*

Subject	Trail	Description
1	1	Forward jumps, turn around
7	1	Walk
39	13	Walk

4.3. Humanoid motion capture data

To extend our experiments from the robot arm to data comparable to a humanoid service robot, we use human motion capture data from the Carnegie Mellon University Motion Capture Database [CMU]. Apart from our application scenario, such data is of interest for visual exploration, e.g. Bernard *et al.* presented a visual data base exploration application [BWK*13]. Similar to the scenario of the robot arm, we are interested in the angular joint position values and forces, as these are relevant from a robotics point of view. Table 1 summarizes the data sets we use.

Figure 9 shows three important dimensions of the right leg—head of femur, knee and ankle—of the data sets 1.1 (jumping forward) and 7.1 (normal walking). The jumping motion in 1.1 (top left) generates clear negative correlations between both pairs of dimensions. For the walking motion in 7.1 (top right), correlations are not clear. Highlighting the point in time of the animation with afterglow and viewing the data set as animation changes in the correlations become visible: negative correlations (bottom left) change a few time steps later into positive correlations (bottom right) within a single step.

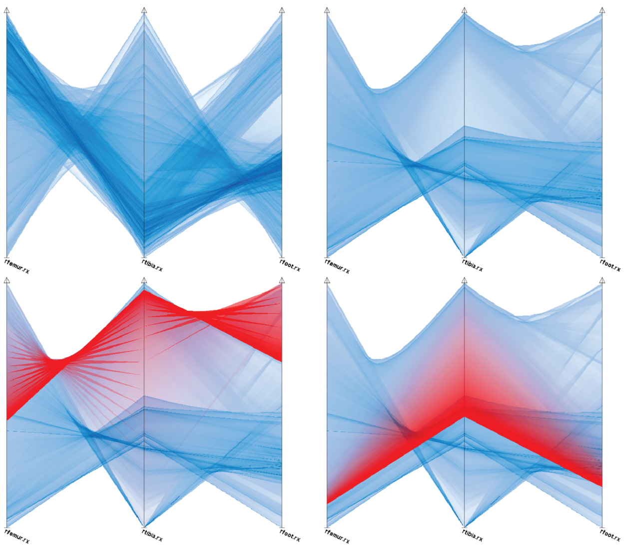

Figure 9: *CPCPs of the three main dimensions of the right leg of data sets 1.1 (top left) and 7.1 (top right). The jumping in 1.1 exhibit strong negative correlation among these dimensions, whereas the correlation is not clear for the walking in 7.1. Viewing the animation with point-in-time afterglow highlighting reveals negative correlation (bottom left) as well as positive correlation (bottom right).*

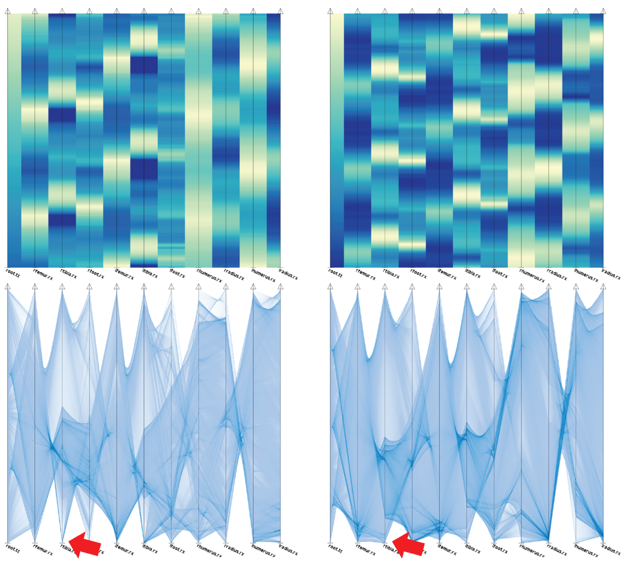

Figure 10: *Two walking data sets (7.1 left; 39.13 right) in THM and CPCPs.*

This indicates the temporal behaviour when rolling the foot through the step while walking.

We, therefore, compare data set 7.1 to a second walking data set (39.13). Figure 10 shows both data sets in comparison with THM and CPCPs. One observation is that subject 39 has considerably less distinctive areas on the *tibia* dimensions (knee angles; 2nd and 5th dimensions from left). In the CPCPs this is clearly visible by differences in the density distributions on these axes (arrows). Subject 7 has rather strong spikes towards the minimum value, while subject 39 shows a more uniformly distribution along the whole axes. On explanation is that subject 39 does not stretch his knees completely while walking.

5. Conclusion and Future Work

The goal of our visualization design is to aid the visual analysis and exploration of multi-dimensional state-space trajectories. To this end, we do not restrict our design to specific setup but instead use general visualization components in coordinated views. These visualizations are THM, CSPLOM and CPCP. For the CSPLOM and CPCP, we have introduced a model for data integration over time and mass conservation. In our result section, we have discussed different case studies, for a model CPS as well as motion capture data. We have presented observations based on the different visualization and derived explanations. These observations are examples of what can be discovered via a visual analysis process using the chosen representations.

The CSPLOM is ideal to spot correlations between dimensions (cf. Section 4.2), which can then be analysed in detail, e.g. via CPCP (cf. Section 4.3). It also visualizes the value changes by connecting consecutive sample points. Except for highlighting a selected point in time and using an afterglow metaphor to visualize the dynamics of the trajectory, information about time, especially the concrete point-in-time of the individual sample points, is lost in these two diagrams. The THM addresses this problem by explicitly displaying values over time (cf. Section 4.1). Each of these representations has advantages and disadvantages. Their combination in coordinated views allows the user to compensate the weak points of the individual visualizations. With the presented observations (cf. Section 4), we were able to better understand relations of different aspects of our experimental CPS. For example, the complexity of correlation of different sensors was surprising (cf. (g) in Figure 1) and interesting for designing automatic event detection algorithms [FWS13].

We only understand our visualization design as first and generic exploratory analysis step. For many applications, specific visualizations convey the relevant information much better, e.g. for motion data (cf. Sections 4.2 and 4.3) explicit visualization in the spatial context of the skeleton is much easier to understand and interpret (cf. [BWK*13]). We will continue using and extending our visualization as part of our research project. The most important issue we want to investigate is the utility of (semi-)automatic filtering and visualization configuration. One issue is finding *good* parameter settings for the visualizations itself automatically [SSK06], e.g. colour maps or ordering of axes for the CPCP suitable for specific tasks. Especially, visualizing long trajectories the CSLPOM and CPCPs accumulate much data with different interpretation necessity. Thus, (semi-)automatic selection of temporal intervals, while still maintaining the overview of the whole trajectory, is not trivially implemented with our approach. A further idea is filtering a data set before visualizing it, based on data-mining algorithms. This filtering might be applied in terms of selecting subsets of dimensions, as well as in terms of adaptive temporal sub-sampling of the input data [TLS12]. Finally, we will investigate further possibilities for the user interface and usability, such as enhanced selection and linking and brushing, to optimized the interactive exploration.

Acknowledgements

This work was partially funded by the European Social Fund (ESF) project #100098171 and by the German Research Foundation (DFG) as part of the Collaborative Research Centre SFB 716 (project D.5).

References

[AdOL04] Artero A. O., de Oliveira M. C. F., Levkowitz H.: Uncovering clusters in crowded parallel coordinates visualizations. In *Proceedings of IEEE Symposium on Information Visualization* (2004), 81–88. http://dx.doi.org/10.1109/INFVIS.2004.68 doi:10.1109/INFVIS.2004.68.

[AIM10] Atzori L., Iera A., Morabito G.: The internet of things: A survey. *Computer Networks 54*, 15 (2010), 2787–2805. http://dx.doi.org/10.1016/j.comnet.2010.05.010 doi:10.1016/j.comnet.2010.05.010.

[AMM*08] Aigner W., Miksch S., Muller W., Schumann H., Tominski C.: Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics* 14, (TVCG, IEEE, Los Alamitos, CA, USA, 2008) pp. 47–60. http://dx.doi.org/10.1109/TVCG.2007.70415 doi:10.1109/TVCG.2007.70415.

[AMST11] Aigner W., Miksch S., Schumann H., Tominski C.: *Visualization of Time-Oriented Data. Human-Computer Interaction Series*. Springer, 2011. http://dx.doi.org/10.1007/978-0-85729-079-3 doi:10.1007/978-0-85729-079-3.

[BBP08] Blaas J., Botha C., Post F.: Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1436–1451. http://dx.doi.org/10.1109/TVCG.2008.131 doi:10.1109/TVCG.2008.131.

[BW08] Bachthaler S., Weiskopf D.: Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1428–1435. http://dx.doi.org/10.1109/TVCG.2008.119 doi:10.1109/TVCG.2008.119.

[BW09] Bachthaler S., Weiskopf D.: Efficient and adaptive rendering of 2-d continuous scatterplots. *Computer Graphics Forum 28*, 3 (2009), 743–750. http://dx.doi.org/10.1111/j.1467-8659.2009.01478.x doi:10.1111/j.1467-8659.2009.01478.x.

[BWK*13] Bernard J., Wilhelm N., Kruger B., May T., Schreck T., Kohlhammer J.: Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (Dec 2013), 2257–2266. http://dx.doi.org/10.1109/TVCG.2013.178 doi:10.1109/TVCG.2013.178.

[Che73] Chernoff H.: The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association 68*, 342 (1973), 361–368. http://dx.doi.org/10.2307/2284077 doi:10.2307/2284077.

[CMU] Cmu graphics lab motion capture database. http://mocap.cs.cmu.edu/. Last accessed on 13 May 2014.

[DHNB09] Dietzsch J., Heinrich J., Nieselt K., Bartz D.: SpRay: A visual analytics approach for gene expression data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, IEEE Computer Society Press, (Atlantic City, 2009) pp. 179–186. http://dx.doi.org/10.1109/VAST.2009.5333911 doi:10.1109/VAST.2009.5333911.

[EDF08] Elmqvist N., Dragicevic P., Fekete J. D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1539–1148. http://dx.doi.org/10.1109/TVCG.2008.153 doi:10.1109/TVCG.2008.153.

[FKLI10] Feng D., Kwock L., Lee Y., Taylor R. M.: Matching visual saliency to confidence in plots of uncertain data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 980–989.

[FSS13] Franke M., Seidl C., Schlegel T.: A seamless integration, semantic middleware for cyber-physical systems. In *ICNSC' 13: Proceedings of the IEEE International Conference on Networking, Sensing and Control* (2013), IEEE, pp. 627–632.

[FWS13] Franke M., Wuttig A., Schlegel T.: Enabling virtual, cloud-based sensors in assisted living environments. In *Proceedings of the 5th IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (2013), IEEE, pp. 513–518. doi:10.1109/cloudcom.2013.74.

[Gri03] Griffel D.: *Applied Functional Analysis*. Dover Books on Mathematics. Dover Publications, Mineola, NY, 2003.

[GXY11] Guo H., Xiao H., Yuan X.: Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates. In *Proceedings of the IEEE Pacific Visualization Symposium* (2011), pp. 19–26. URL: http://vis.pku.edu.cn/research/publication/pacificvis11-hdtf-small.pdf.

[Har75] Hartigan J.: Printer graphics for clustering. *Journal of Statistical Computation and Simulation 4*, 3 (1975), 187–213. http://dx.doi.org/10.1080/00949657508810123 doi:10.1080/00949657508810123.

[HB03] Harrower M., Brewer C. A.: Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal 40*, 1 (2003), 27–37. http://dx.doi.org/10.1179/000870403235002042 doi:10.1179/000870403235002042.

[HBW11] Heinrich J., Bachthaler S., Weiskopf D.: Progressive splatting of continuous scatterplots and parallel coordinates. *Computer Graphics Forum 30*, 3 (2011), 653–662.

[HGM*97] HOFFMAN P., GRINSTEIN G., MARX K., GROSSE I., STANLEY E.: DNA visual and analytic data mining. In *Proceedings of Visualization '97* (1997), pp. 437–441. http://dx.doi.org/10.1109/VISUAL.1997.663916 doi:10.1109/VISUAL.1997.663916.

[HW09] HEINRICH J., WEISKOPF D.: Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1531–1538. http://dx.doi.org/10.1109/TVCG.2009.131 doi:10.1109/TVCG.2009.131.

[HW13] HEINRICH J., WEISKOPF D.: State of the art of parallel coordinates. In *STAR Proceedings of Eurographics 2013* (2013), Eurographics Association, Girona, Spain, pp. 95–116.

[Ins85] INSELBERG A.: The plane with parallel coordinates. *The Visual Computer 1*, 4 (1985), 69–91. http://dx.doi.org/10.1007/BF01898350 doi:10.1007/BF01898350.

[Ins01] INSELBERG A.: Conflict detection and planar resolution for air traffic control. In *Proceedings of IEEE Intelligent Transportation Systems* (2001), pp. 1200 –1205. http://dx.doi.org/10.1109/ITSC.2001.948834 doi:10.1109/ITSC.2001.948834.

[Ins09] INSELBERG A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, Heidelberg, 2009.

[jac] Jaco Robot Arm from Kinova Robotics. http://kinovarobotics.com/products/jaco-research-edition/. Last accessed on 13 May 2014.

[JLC07] JOHANSSON J., LJUNG P., COOPER M.: Depth cues and density in temporal parallel coordinates. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization* (2007), vol. *7*, pp. 35–42.

[Kei02] KEIM D.: Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (2002), 1–8. http://dx.doi.org/10.1109/2945.981847 doi:10.1109/2945.981847.

[KHD*10] KEIM D. A., HAO M. C., DAYAL U., JANETZKO H., BAK P.: Generalized scatter plots. *Information Visualization 9*, 4 (2010), 301–311. http://dx.doi.org/10.1057/ivs.2009.34 doi:10.1057/ivs.2009.34.

[KLG*13] KUHN A., LINDOW N., GÜNTHER T., WIEBEL A., THEISEL H., HEGE H.-C.: Trajectory density projection for vector field visualization. *EuroVis - Short Papers 2013* (2013), 31–35.

[KZZM12] KUANG X., ZHANG H., ZHAO S., MCGUFFIN M. J.: Tracing tuples across dimensions: A comparison of scatterplots and parallel coordinate plots. *Computer Graphics Forum31*, 3 (2012), 1365–1374. http://dx.doi.org/10.1111/j.1467-8659.2012.03129.x doi:10.1111/j.1467-8659.2012.03129.x.

[LS09] LEE T.-Y., SHEN H.-W.: Visualization and exploration of temporal trend relationships in multivariate time-varying data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1359 –1366. http://dx.doi.org/10.1109/TVCG.2009.200 doi:10.1109/TVCG.2009.200.

[LT10] LEHMANN D., THEISEL H.: Discontinuities in continuous scatter plots. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1291–1300. http://dx.doi.org/10.1109/TVCG.2010.146 doi:10.1109/TVCG.2010.146.

[LT11] LEHMANN D., THEISEL H.: Features in continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1912–1921. http://dx.doi.org/10.1109/TVCG.2011.200 doi:10.1109/TVCG.2011.200.

[MFL13] MOLCHANOV V., FOFONOV A., LINSEN L.: Continuous representation of projected attribute spaces of multifields over any spatial sampling. *Computer Graphics Forum 32*, 3pt3 (2013), 301–310. http://dx.doi.org/10.1111/cgf.12117 doi:10.1111/cgf.12117.

[MW91] MILLER J. J., WEGMAN E. J.: Construction of line densities for parallel coordinate plots. In *Computing and Graphics in Statistics*. vol. *36*, Springer, New York, 1991, pp. 107–123.

[Poo10] POOVENDRAN R.: Cyber-physical systems: Close encounters between two parallel worlds. *Proceedings of the IEEE 98*, 8 (2010), 1363–1366. http://dx.doi.org/10.1109/JPROC.2010.2050377 doi:10.1109/JPROC.2010.2050377.

[SSK06] SCHNEIDEWIND J., SIPS M., KEIM D.: Pixnostics: Towards Measuring the Value of Visualization. In *Proceedings of the IEEE Symposium On Visual Analytics Science And Technology* (2006), pp. 199–206. http://dx.doi.org/10.1109/VAST.2006.261423 doi:10.1109/VAST.2006.261423.

[TAE*11] TATU A., ALBUQUERQUE G., EISEMANN M., BAK P., THEISEL H., MAGNOR M. A., KEIM D. A.: Automated analytical methods to support visual exploration of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 17*, 5 (2011), 584–597.

[tCMR07] TEN CAAT M., MAURITS N. M., ROERDINK J. B.: Design and evaluation of tiled parallel coordinate visualization of multichannel EEG data. *IEEE Transactions on Visualization and Computer Graphics 13*, 1 (2007), 70–79. http://dx.doi.org/10.1109/TVCG.2007.9 doi:10.1109/TVCG.2007.9.

[TFA*11] TAM G. K. L., FANG H., AUBREY A. J., GRANT P. W., ROSIN P. L., MARSHALL D., CHEN M.: Visualization of time-series data in parameter space for understanding facial dynamics. *Computer Graphics Forum 30*, 3 (2011), 901–910. http://dx.doi.org/10.1111/j.1467-8659.2011.01939.x doi:10.1111/j.1467-8659.2011.01939.x.

[TLS12] TONG X., LEE T.-Y., SHEN H.-W.: Salient time steps selection from large scale time-varying data sets with dynamic time warping. In *LDAV' 12: Proceedings of 2012 IEEE Symposium on Large Data Analysis and Visualization*, (2012), pp. 49–56. http://dx.doi.org/10.1109/LDAV.2012.6378975 doi:10.1109/LDAV.2012.6378975.

[WB94] WONG P. C., BERGERON R. D.: 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews,* *Methodologies, and Techniques* (1994), G. M. Nielson, H. Hagen and H. Mueller (Eds.), Dagstuhl, Germany, pp. 3–33.

[WF09] WILKINSON L., FRIENDLY M.: The history of the cluster heat map. *The American Statistician 63*, 2 (2009), 179–184.

[WLG97] WEGENKITTL R., LOFFELMANN H., GROLLER E.: Visualizing the behaviour of higher dimensional dynamical systems. In *Proceedings of Visualization '97* (1997), IEEE, pp. 119–125. http://dx.doi.org/10.1109/VISUAL.1997.663867 doi:10.1109/VISUAL.1997.663867.

[WZL*12] WANG Y., ZHANG J., LEHMANN D. J., THEISEL H., CHI X.: Automating transfer function design with valley cell-based clustering of 2d density plots. *Computer Graphics Forum 31*, 3pt4 (2012), 1295–1304. http://dx.doi.org/10.1111/j.1467-8659.2012.03122.x doi:10.1111/j.1467-8659.2012.03122.x.

Supporting Information

Additional Supporting Information may be found in the online version of this article at the publisher's web site:

Video S1