ThermalPlot: Visualizing Multi-Attribute Time-Series Data Using a Thermal Metaphor

Holger Stitz, *Student Member, IEEE,* Samuel Gratzl, *Student Member, IEEE,*
Wolfgang Aigner, *Member, IEEE,* and Marc Streit, *Member, IEEE*

Abstract—Multi-attribute time-series data plays a vital role in many different domains, such as economics, sensor networks, and biology. An important task when making sense of such data is to provide users with an overview to identify items that show an interesting development over time, including both absolute and relative changes in multiple attributes simultaneously. However, this is not well supported by existing visualization techniques. To address this issue, we present *ThermalPlot*, a visualization technique that summarizes combinations of multiple attributes over time using an item's position, the most salient visual variable. More precisely, the x-position in the *ThermalPlot* is based on a user-defined degree-of-interest (DoI) function that combines multiple attributes over time. The y-position is determined by the relative change in the DoI value (ΔDoI) within a user-specified time window. Animating this mapping via a moving time window gives rise to circular movements of items over time—as in thermal systems. To help the user to identify important items that match user-defined temporal patterns and to increase the technique's scalability, we adapt the level of detail of the items' representation based on the DoI value. Furthermore, we present an interactive exploration environment for multi-attribute time-series data that ties together a carefully chosen set of visualizations, designed to support analysts in interacting with the *ThermalPlot* technique. We demonstrate the effectiveness of our technique by means of two usage scenarios that address the visual analysis of economic development data and of stock market data.

Index Terms—Time-dependent data, multi-attribute data, focus+context, semantic zooming.

✦

1 INTRODUCTION

UNDERSTANDING temporal developments of multi-attribute data is an essential task in many domains, such as economics, sensor networks, biology, or data journalism. Gaining new insights from such data can be challenging—even for a single multi-attribute item. However, the complexity increases significantly when scenarios comprise a collection of items, where each item comes with a set of multiple attributes that change over time. An important task when making sense of such data is to provide users with an overview for identifying items that show an interesting temporal development, including both absolute and relative changes of multiple attributes simultaneously. Two of the main challenges in this context are (I) integrating multiple heterogeneous attributes from a collection of items and make them comparable, and (II) showing multiple levels of temporal dynamics. Although a wide array of visualization techniques have been proposed for addressing both challenges, they often scale poorly to multiple attributes, a large set of items, or long time-series [2]. In contrast to visual approaches, automated alternatives for summarizing multi-attribute data, such as Principle Component Analysis (PCA) and Multidimensional Scaling (MDS), often do not produce projections that can be interpreted intuitively. To the best of our knowledge, no single approach exists that effectively handles both challenge (I) and (II).

In this paper, we aim to fill this gap by presenting the *ThermalPlot* visualization technique as our primary contribution.

ThermalPlot provides an overview of a collection of items, allowing analysts to quickly identify items that show an interesting development over time. The technique encodes time-dependent changes in attributes into an item's position, which is known to be the strongest visual variable for encoding quantitative data [25]. Changes can be derived effectively from the position in order to detect outliers, trends, and patterns. The position is based on a modular degree-of-interest (DoI) function which combines multiple attributes with adjustable weight. As a secondary contribution, we introduce an interactive exploration environment for multi-attribute time-series data that integrates a series of support views which enrich interaction with the *ThermalPlot* technique.

We introduce the *ThermalPlot* technique and its implementation using a publicly available data set from the *Organization for Economic Co-operation and Development* (OECD)[1]. The data set contains an extensive collection of attributes for all OECD member countries. To illustrate our technique, we chose *long-term*, and *short-term interest rates* collected on a monthly basis between January 2000 and July 2015. We show how Latvia managed to tackle the financial crisis in 2009 and how it developed afterwards. In addition, we demonstrate scalability and effectiveness in a stock data use case where a private investor makes informed stock investment decisions using the *ThermalPlot* system.

2 RELATED WORK

Due to the broad applicability of multi-attribute time-series data, a vast body of related work exists. For an extensive survey on the special characteristics of time in general, and a systematic discussion of available techniques, we refer the reader to the

• *Holger Stitz, Samuel Gratzl and Marc Streit are with Johannes Kepler University Linz.*
  E-mail: {holger.stitz, samuel.gratzl, marc.streit}@jku.at
• *Wolfgang Aigner is with St. Poelten University of Applied Sciences.*
  E-mail: wolfgang.aigner@fhstp.ac.at

Pre-print version

1. http://stats.oecd.org, data set downloaded on 2015-08-21.

book on time-oriented data visualization [2] and the corresponding online collection of available techniques[2].

Exploring complex real-world phenomena almost always requires taking into account a number of interrelated attributes along with their changes over time. To effectively address this goal, it is necessary to tackle two challenges, which we briefly mentioned in the introduction already: (I) the integration and comparison of multiple heterogeneous attributes for a collection of items, and (II) the extraction of temporal dynamics on multiple levels. In the following, we divide the related work along the lines of these challenges and discuss existing solutions together with their respective strengths and weaknesses.

2.1 Multi-Attribute Item Comparison

Addressing challenge (I)—integrating and comparing multiple heterogeneous attributes of an item collection—is difficult because the comparison should ideally work in both dimensions: across multiple attributes of a single item and across a single attribute of multiple items. Both can be addressed, for instance, by superimposing multiple curves in a line chart, by stacking multiple line or horizon charts [16], or by using other pixel-based techniques [3], [19]. However, these solutions often do not work for making comparisons in both directions simultaneously. In our approach, we summarize multiple-attributes using a modular DoI function and visualize the aggregated items based on their DoI values.

2.2 Visual Analysis of Temporal Dynamics

To investigate the dynamics of time-dependent data, referred to as challenge (II), an analyst needs to take into account temporal patterns on both a global (coarse temporal granularity) and a local scope (fine temporal granularity). The literature differentiates between two possible kinds of visual mappings for encoding time-dependent data: static (mapping time to space) and dynamic (mapping time to time) [2].

Mapping Time to Time

The main advantage of using animation to communicate temporal changes is its intuitiveness, as it is the way how we perceive changes in real world. Users might also find it more exciting and fun to use [29]. Examples for animated visualizations are the well-known *Gapminder Trendalyzer*[3] that uses bubble charts to plot demographic changes over time and animated scatter plots for stock market data [33].

While the use of animation can be a good design choice for presentation purposes, animation is known to be problematic for analysis tasks [21], [29]. Also, animation can be effective to encode smooth changes, however, users might get confused if the changes affect too many data items, if items do not move in synchrony, or if items reverse their tracks over time [29]. The reason for these limitations is our limited ability to follow, memorize, and compare information across time steps [9]. The problem can be alleviated to a certain extent by allowing users to pause, replay, and adjust animation speed. However, due to the shortcomings, we chose to not rely on animation as a primary encoding principle to summarize temporal changes in large item collections.

2. http://survey.timeviz.net
3. http://www.gapminder.org

Mapping Time to Space

The principle of mapping time to space utilizes position to encode change and temporal patterns. A wide array of techniques exist for investigating seasonal patterns and trends (e.g., *Cycle Plot* [8] and *GROOVE* [22]). However, they do not scale well to multiple attributes, a large set of items, or long time-series. While heatmaps and pixel-based techniques scale better, it can become difficult for users to compare temporal patterns and trends across items or attributes.

In contrast to animation, small multiples [35] and trajectories (aka traces) are better suited for analysis tasks performed on time-dependent data [29]. The small multiple representation can be a plot, a glyph [10], or any other visualization. While small multiples can be used to create an effective overview (e.g., *LiveRAC* [27]), they do not scale beyond a couple of dozen items or attributes. The sequence view in [31] is an example for a large small multiple matrix showing line charts in 66 columns and 83 rows. However, such large matrices only work in combination with automated detection and guidance mechanisms that point users to relevant parts.

Trajectories

Trajectories are another option to encode change over time. Trajectories are visual traces that appear when continuously plotting the temporal development of items in 2D (e.g., [28], [31]) or 3D [37]. The emerging paths can then be used to compare patterns across multiple items. However, multiple overlapping trajectories can result in visual clutter, reducing their effectiveness [28]. Clustering algorithms can alleviate the problem by aggregating trajectories based on some kind of similarity metric [31], [38]. Ziegler et al. [41], for example, cluster companies that belong to the same industry sector and then present the trajectories for each company in a small multiple view that is grouped by cluster. However, this solution comes with extra cognitive load for comparing the items. In the *ThermalPlot*, trajectories can be displayed on demand for selected items only in order to keep the visual clutter at a minimum.

DimpVis [20] is a technique where users directly interact with the trajectories for navigating in time. While the technique is very intuitive, it lacks direct support when items remain at one specific position for several time steps and it can get difficult to work with complex trajectories. Instead of allowing the user to directly interact with the trajectories, we decided to let users select a single time step via line charts in a detail view. The corresponding parts of the trajectories are then highlighted accordingly.

In summary, we could not identify an existing technique that allows users to address both challenges for a large collection of multi-attribute items.

3 USER TASKS

In visual data analysis, users frequently face open-ended, ill-defined tasks such as "find or detect something interesting". Particularly when dealing with multi-attribute data over time, such discovery tasks can become very cumbersome, as many attributes must be taken into account. To bring some clarity to such fuzzy analytical objectives, we identified five user tasks that must be supported and serve as a set of design requirements to be met.

T1: Monitor the development of multiple items in a certain time window. The user wants to get an overview of multiple items and monitor them simultaneously over a certain period of time.

T2: Select attributes and define their interestingness. Since not all data item attributes are equally interesting or important in specific use cases, the user needs to select one or multiple attributes and define how interesting each attribute is in relation to the other (DoI).
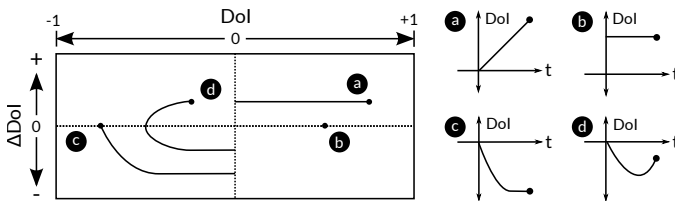
T3: Detect items that are (most) interesting. According to T2, the user wants to detect items that best fulfill the defined interestingness metric. Such items need to be made visually salient.

T4: Understand why the items are considered to be interesting. After discovering a set of interesting items, users need to understand why the system considered a specific item to be interesting.

T5: Monitor the development of a single item. Finally, it is not only important to know which attributes contributed to the interestingness of an item, but also to be able to analyze them in detail. This involves the need to investigate and compare the development of multiple attributes over time.

4 *ThermalPlot* VISUALIZATION CONCEPT

The fundamental idea underlying *ThermalPlot* is simple but effective: we map a user-specified degree-of-interest (DoI) value on the x-axis and the change in the DoI value (ΔDoI) on the y-axis, as illustrated in Figure 1. Plotting an attribute and its first derivative is not new (see for instance Figure 2 in [23] that plots the *stock price* vs. *price change*). However, instead of making static plots for single attributes, we create interactive visualizations that summarize the temporal development of multiple attributes (addressing task T1).

Fig. 1: *The DoI value is mapped to the x-axis and the ΔDoI to the y-axis. DoI values that change over time result in distinctive positions and trajectories of items in the ThermalPlot space. The DoI values of the examples are (a) linearly increasing, (b) constant, (c) decreasing and then constant, and (d) decreasing first and then increasing.*

In the *ThermalPlot* the DoI value is a weighted combination of one or multiple attributes over time, as explained in detail in the following section. Figure 2 shows a *ThermalPlot* visualization of the previously mentioned OECD data set. Depending on the usage scenario, the analyst defines a DoI function that results in high DoI values for one or multiple items of interest (addressing task T3). The ΔDoI is determined by the DoI change between the start (t_s) and end (t_e) of a user-defined time window (see Figure 2(b)). The items are then placed in the plot according to their DoI and ΔDoI values. This mapping results in a visualization where users can derive an aggregated summary of the items' developments over time from their positions in the plot. Latvia, for instance, was hit hard by the financial crisis in 2009, but showed a positive development in the *short-term interest rate* in the following years (compared to 2006). Thus, it is positioned in the upper left area (see Figure 2(a)). In contrast, Greece has a constant positive *short-term interest rate*, but is located in the lower left area because of

the ongoing negative development of the higher weighted *long-term interest rate* attribute.

Another consequence of this mapping are the distinctive movements of items within the plot over time that are reminiscent of a thermal system. Items whose DoI increases move from left to right (see Figure 1(a)), while items with decreasing DoI move from right to left (see Figure 1(c)). The faster the DoI changes, the higher the items will rise. Consequently, negative changes in the DoI values result in downward movements of items. Mapping the ΔDoI values to the y-axis naturally separates items with a positive trend from those with a negative trend over the chosen time window. Together, this behavior results in circular movements of items. The magnitude of changes in the multi-attribute data determines the size of the circular patterns, resulting in macro and micro movements. Figure 1 illustrates four example movements through the *ThermalPlot* space, together with the corresponding development of the DoI value over time. Note that this does not necessarily mean that the items are constantly changing their positions in the plot, as the position is only updated when new data arrives in live streaming scenarios. Depending on the frequency of the updates, this might only happen monthly, as in our OECD use case, or daily, as in the stock market use case.

The *ThermalPlot* concept is particularly powerful in two basic scenarios: (1) showing a static snapshot that summarizes the temporal development of items in a given time window. (2) presenting the current status of a live streaming data set, where item positions are slowly updated when new data comes in.
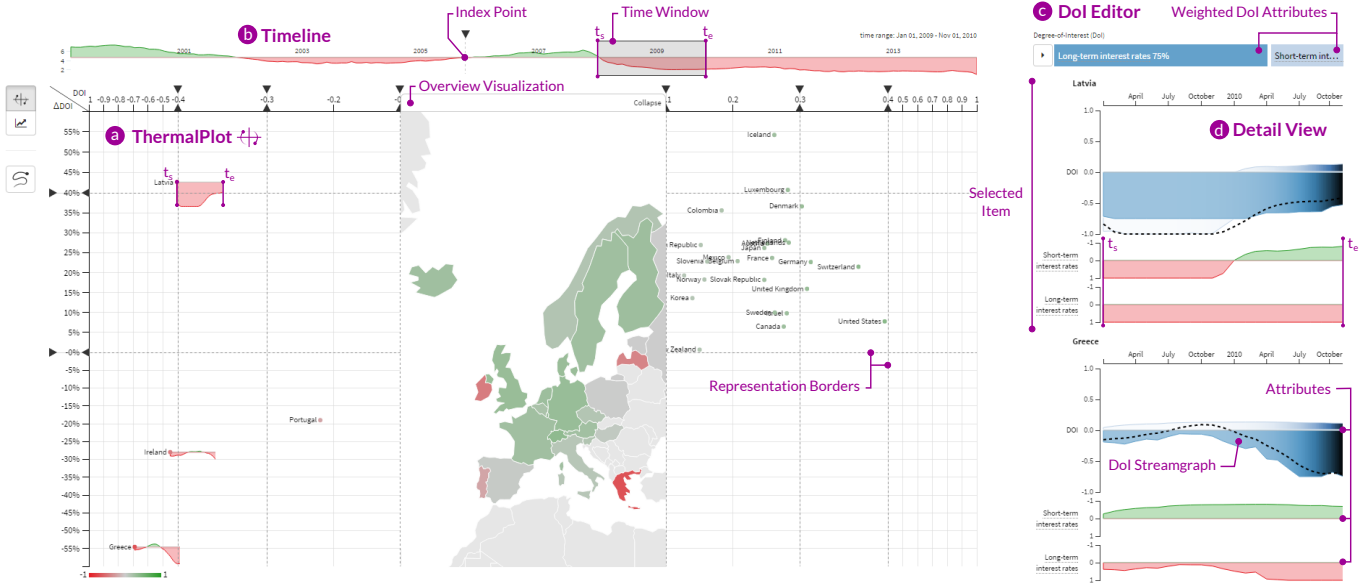
4.1 Modular Degree-of-Interest (DoI)

Using DoI functions to adapt the visual representation is a well-known approach, and has been applied in many different ways and contexts, for example, to explore trees [11], temporal data [7], large static graphs [36], dynamic graphs [1], and to the visual analysis of small interconnected biological networks [24]. Intuitively, the value that results from a DoI function should reflect how interesting a particular item is to the analyst. In the *ThermalPlot* an item's position at a specific time point directly corresponds to its DoI value and therefore represents its interestingness. Before we discuss the additional roles of the DoI function in the *ThermalPlot* besides positioning the items, we introduce the method by which DoI values can be calculated for time-series data.

In his fundamental work on generalized fisheye views [11], Furnas introduced the concept of DoI, where the function for calculating the DoI can be driven by various attributes. Which attributes define the importance of an item depends on the data set and task. The attributes that contribute to the DoI can be either static, such as the *founding year of the state*, or dynamically changing over time, such as the *interest rate* or the *gross domestic product (GDP)* of a country. *ThermalPlot* supports multi-attribute DoI functions, where for each time point individual attribute values are combined using a weighted sum, resulting in a single raw DoI value ($DoI_{raw}(t)$) (addressing task T2).

$$DoI_{raw}(t) = \sum_{i=1}^{n} w_i \times v_i(t) \mid \sum_{i=1}^{n} w_i = 1.$$

The weights applied can be defined freely by the user where n is the number of attributes that contribute to the raw DoI, w are the weights of the components, which sum up to one, and $v_i(t)$ is the attributes' value at time point t.

Fig. 2: *The ThermalPlot technique (a) integrated in the exploration environment for multi-attribute time-series data showing the development of countries within a user-specified time window (b). Item positions in the plot are based on the selected index point and the weighted DoI attributes configured via the DoI editor (c). The detail view (d) shows the development of the composed DoI over time as streamgraphs and line charts of single attributes for selected items. Latvia, for instance, was hit hard by the financial crisis in 2009, but showed a positive development in the short-term interest rate in the following years (compared to 2006). Consequently, it appears in the upper left area. In contrast, Greece appears in the lower left area, even though the short-term interest rate attribute shows a constant positive development. However, it is surpassed by the higher weighted long-term interest rate attribute, which results in an overall negative development of the country.*

To incorporate trends and temporal effects at a specific time point t, we apply an exponential smoothing strategy [13]. Exponential smoothing aggregates multiple previous time points by assigning decreasing weights over time. We use the *Holt-Winters exponential smoothing* method (also termed second-order exponential smoothing) [40], which is known to work well with data that contains trends or seasonal patterns.

$$DoI(t) = \alpha \times DoI_{raw}(t) + (1-\alpha) \times$$
$$(DoI_{raw}(t-1) + DoI_{trend}(t-1)) .$$

The α ($0 \leq \alpha \leq 1$) smoothing factor determines how many of the previous time steps influence the current value. An α value of 1 disables smoothing, while a value close to zero results in a strongly smoothed value. $DoI_{trend}(t)$ is an estimator for the trend of the time series, where β ($0 \leq \beta \leq 1$) is an additional smoothing factor for the trend similarly:

$$DoI_{trend}(t) = \beta \times (DoI(t) - DoI(t-1)) +$$
$$(1-\beta) \times DoI_{trend}(t-1).$$

Both parameters are configurable by the user. By default, we use $\alpha = 0.4$, $\beta = 0.4$ in the OECD use case.

An inherent problem of double exponential smoothing is finding an initial value for $DoI(0)$ and $DoI_{trend}(0)$. Depending on α and β, the influence of one specific time point $t-k$ on the current time point t can be computed. Thresholding the influence leads to a k which can be used as the starting point for computing the current time point t by setting $DoI(t-k) = DoI_{raw}(t-k)$ and $DoI_{trend}(t-k) = 0$. In our OECD scenario, for instance, we used $k = 12$, which corresponds to one year, as the granularity of k is defined in months. This results in a weight of time point $t-k$ of

just 0.00087 at the current time point t when using an alpha value of 0.4.

For calculating the ΔDoI, the user needs to specify the time window (ranging from t_s to t_e with $\Delta t = t_e - t_s$). The ΔDoI is then defined as:

$$\Delta DoI(t) = DoI(t) - DoI(t - \Delta t).$$

In the case of $t = t_e$, this leads to $\Delta DoI(t_e) = DoI(t_e) - DoI(t_s)$, i.e., the DoI's change within the selected time window.

Normalization

To obtain correct DoI values that result in valid positions of items in the *ThermalPlot* space, it is essential that the values of a specific item's attribute that contributes to the DoI can be directly compared across items in a meaningful way (e.g., the *population of a country*). In contrast, *interest rates* of a country are determined by many factors and their absolute values cannot be meaningfully compared. We address this problem by letting the user define an index time point t_{index} that serves as a reference point [4]. For the OECD data set, this reference point could be, for instance, a year before the financial crisis. Taking the change relative to the index point rather than absolute values enables comparison of the DoI values across items.

$$v_{rel}(t) = \frac{v(t) - v(t_{index})}{v(t_{index})}.$$

A related issue is that values across various attributes need to be in the same range before they can be incorporated into a combined DoI value. To address this problem, we allow users to specify a min/max value for each attribute, which we then use to normalize the values across attributes.
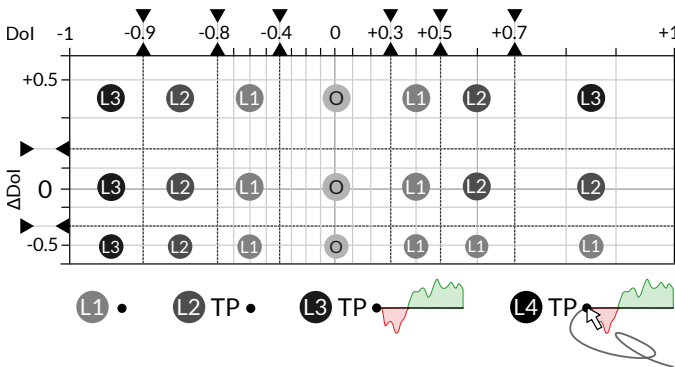
DoI Symmetry

DoI values are commonly defined in the range $[0,1]$, where 0 means not interesting at all and 1 highly relevant. However, in scenarios that use an index point for normalization, the DoI values can be both positive and negative. In the OECD use case, for instance, an analyst could be interested in the biggest losers of the financial crisis in 2009 (negative overall DoI) that showed an upward trend in the following year (positive ΔDoI). To support such scenarios, we define the DoI in the range $[-1,+1]$, as illustrated in Figure 1. In these symmetric DoI cases, 0 still means not interesting. However, we let users define whether negative, positive, or both DoI values are of interest (see Section 4.3 for details).

4.2 Clutter Reduction Strategies

As in any scatterplot representation, a high density of marks in a certain region of the plot can result in occlusion problems and visual clutter. To increase the scalability of the *ThermalPlot* technique regarding the number of items, we apply a two-fold strategy that combines semantic zooming [12] with optional orthogonal stretching of scales [30].

Semantic zooming

We use semantic zooming to adapt the level of detail of an item to its DoI or ΔDoI value (addressing task T3). In our approach it is possible to define **representation borders**, that cause the representation to change when crossed. The borders can be defined either statically for a specific scenario or interactively by a user. Representation borders can be defined for both axes, resulting in a grid in which the level of detail can be specified for each grid cell (see Figure 3). Hence, every cell can also be seen as a rectangular semantic lens [34]. For instance, if a user is interested in items that are located in the sectors on the upper right as well as those contained in the first grid column on the left in the *ThermalPlot*, she can increase the level of detail for these particular regions of the plot.

Fig. 3: *DoI values between [-1,1] are mapped to the x-axis and ΔDoI values with a variable range to the y-axis. The visual space can be geometrically distorted by manipulating the DoI value associated with a representation border. The level of detail in each region of the plot can be configured individually. Examples of the pre-defined levels of detail L1 to L4 are shown at the bottom. Note that L4 is reserved for selections only and O indicates an embedded overview visualization (see Section 4.3).*

What information the various levels of detail show depends again on the usage scenario. The design space ranges from a single

pixel to multivariate data glyphs [39], and even embedded full-fledged visualizations. In our prototype we support four levels of detail, as illustrated in Figure 3, that are defined incrementally. This means that the representation of each level contains all visual elements from the previous level, plus new elements for encoding further details.

L1: Colored Mark. As a mark we chose a circle that can be colored by either a static attribute value or a temporally changing attribute that is aggregated to a single value.

L2: L1 + Label. The L2 representation extends the mark from L1 by the item's textual label.

L3: L2 + Line Chart. In addition to the mark and label, we show a full line chart with the temporal development of a single attribute for the selected time window.

L4: L3 + Trajectory. In L4 we add a trajectory that describes the path an item takes through the *ThermalPlot* space over time.

Orthogonal Stretching

In addition to semantic zooming, we use orthogonal stretching [30] as a second measure to reduce the visual clutter. Which parts of the scale should be stretched or compressed can either be statically defined for a specific setup or dynamically set by the user. In theory, the representation borders are independent of the handles for the orthogonal stretching. However, for the sake of simplicity in the interaction, we use the same interactive borders for controlling both. Users can move a representation border by dragging the small triangle that points away from the plot, while the triangle pointing towards the plot is the handle for distorting the space, as illustrated in Figure 3. Integrating a fisheye lens would also be an option. However, as the items' position plays a central role in our method, we decided to refrain from applying non-linear distortion techniques.
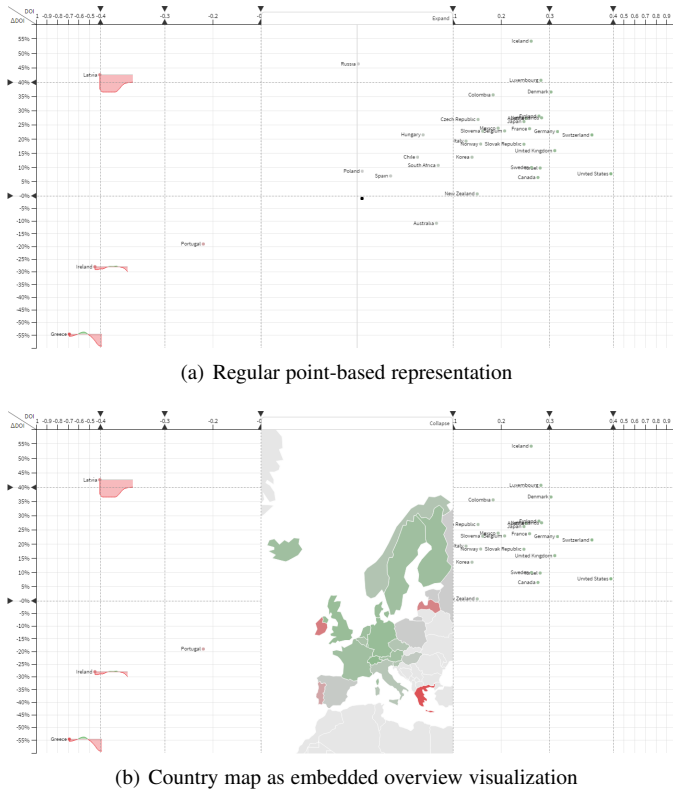
4.3 Integrated overview visualization

Items with a zero or low DoI value are by definition of no or low interest to the user. So instead of representing them by their default point-based representation, as shown in Figure 4(a), we optionally use the space to visualize all items as marks inside an embedded overview visualization (see Figure 4(b) or Figure 10). The fixed layout of this visualization provides a meaningful structure that users can employ for quickly locating items. Depending on the DoI value range, the overview visualization can be either embedded in the center of *ThermalPlot* or attached on the left or right side. To flexibly tailor the *ThermalPlot* space to the task at hand, users can freely move and resize the overview representation via drag-and-drop.

What visualization technique fits best, depends on the domain problem and data. In our OECD example, the European countries are shown in map form (see Figure 4(b)). For stock data the companies can be visualized as a *Map of the Market* [26] (see Figure 10). To provide further contextual information, static attributes associated with the items can be additionally encoded in the overview visualization. In our OECD use case, for instance, the color of each country depends on its current DoI value. By default, we apply a diverging color scale where red is mapped to $DoI = -1$, gray to $DoI = 0$, and green to $DoI = +1$. Color-blind users can switch to suitable alternative mappings. As all items that are hidden by the overview visualization are of little or no interest, we desaturate them in the treemap, to make items with a high DoI value more salient.

(a) Regular point-based representation

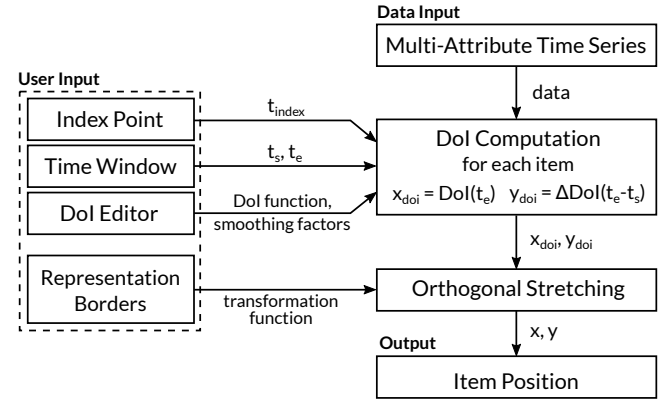(b) Country map as embedded overview visualization

Fig. 4: *Different options for visualizing items with a low or zero DoI value. In (a) the items scattered around the vertical center line are shown as points. (b) shows a country map embedded inside the ThermalPlot that gives a meaningful structure to items.*

As the overview contains all items with a DoI of zero or close to zero, items seamlessly enter the *ThermalPlot* space with an increased DoI value. The DoI threshold that defines up to which value items remain inside the overview visualization can be interactively specified by the user. Analysts can switch to the regular point-based representation by hiding the overview visualization (see Figure 4(a)). Animated transitions [17] allow users to visually follow items during the switch operation.

5 INTERACTIVE EXPLORATION ENVIRONMENT FOR MULTI-ATTRIBUTE TIME-SERIES DATA

To put the *ThermalPlot* method to practical use, a couple of support visualizations are required. As illustrated in Figure 2, the overall multi-coordinated view setup consists of four linked components: (a) the *ThermalPlot* as the heart of our system, (b) the timeline, (c) the DoI editor, and (d) the detail view. The **timeline** showing the full time window for which data is available lets the user set an index point t_{index} and a time window $t_s - t_e$ that specifies the data which then serves as an input to *ThermalPlot*. The **DoI editor** is the interface for composing the DoI function by means of combining and weighting multiple attributes. Figure 5 illustrates the inputs and the data flow between the different components. The last component of the setup is a **detail view** that presents the development of multiple attributes over time for the current item selection.

To cover the requirements of a wide range of different usage scenarios, the defined set of views in the *ThermalPlot* setup can

Fig. 5: *Illustration of the data flow along with inputs and outputs. The multi-attribute time series and the user input are combined in the DoI and ΔDoI computation, which are then transformed by the representation borders. The final output is the items' x- and y-position in the ThermalPlot space.*

be extended with special-purpose views that are tailored to the domain problem. For instance, when dealing with large item collections, it can be useful to add an interface for browsing and filtering items by static attributes or weighted combinations thereof [14] (see Section 6).

5.1 Interaction with the *ThermalPlot*

In the prototype implementation, multiple ways of selecting items exist. Users can directly select single items in the plot by clicking on their respective representations. The items' representation in the overview visualization depends on the used visualization technique (e.g., countries on a map and blocks in a treemap), while the *ThermalPlot* always uses the point, which is available at all levels of detail. As an alternative, we provide a lasso selection for flexibly defining arbitrary regions of interest. Instead of completely removing non-selected items from the overview visualization and the *ThermalPlot* itself, we decrease their opacity to keep them as contextual information.

5.2 Timeline

The timeline serves two purposes. First, it provides the user with an overview of the time window for which the multi-attribute data is available (addressing task T1). Second, it is the interface for setting the index point (see Section 4) and the time window which defines the data upon which the DoI over time is calculated. Both the time window and the index point can be moved interactively using drag-and-drop. The size of the time window can also be changed dynamically. A change in one of these settings triggers an update of the *ThermalPlot* where the data is streamed from the server. During data transfer, we indicate the progress by a gradually filled up background of the time window widget in the *ThermalPlot* and the detail view shows the portion of the data that has already been transmitted.

For scenarios where the item collection can be represented by a surrogate attribute that summarizes all items, we show a line chart in the background of the timeline. To indicate that the position of the index point defines the global point of reference for the DoI calculation, we set the baseline of the line chart according to the value of the summary attribute at the corresponding time point.

We then color all parts of the curve that are above the baseline in green and the parts below in red, as can be seen in Figure 2. For stock market data, for instance, we plot the *Standard & Poor's 500* (*S&P 500*) index point over time. If no meaningful summary attribute is available, as in our OECD data set, we define a default attribute that is used for the timeline (e.g., the average *short-term interest rate*).

5.3 Detail View

The *ThermalPlot* technique summarizes effectively the development of an item collection based on the aggregated multi-attribute DoI. Additionally, it is essential that users understand what contributes to the DoI value of items over time (task T4) and are able to drill down to the actual time-series data of the different attributes that contribute to the aggregated DoI value (task T5). In the detail view, we therefore show a streamgraph [6] for every selected item encoding the DoI value and its weighted components over time together with the full line charts for a predefined set of attributes that are associated with the currently selected items (addressing tasks T4 and T5). Both chart types show the data within the chosen time window $t_s - t_e$. If data is missing for specific time points, we interpolate them linearly based on their nearest valid neighboring data points.
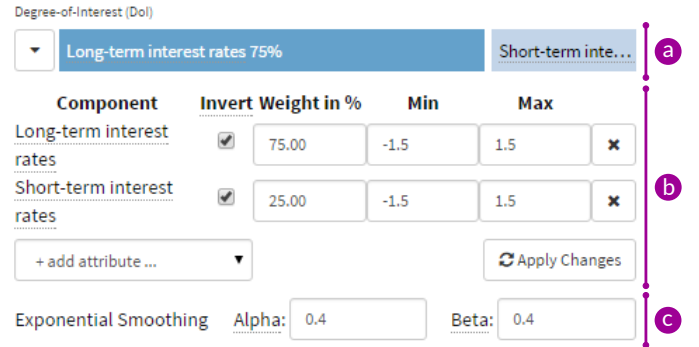
5.4 DoI Visualizations

We added three building blocks to the setup that are designed to allow the user to specify and understand the composed DoI function: the **DoI editor** as an interface for interactively configuring the DoI function; **DoI streamgraphs** for visualizing the contribution of each attribute for a specific item; and **trajectories** for showing the path an item takes through the *ThermalPlot* space.

DoI Editor

Users can interactively define and manipulate the multi-attribute DoI via an integrated DoI editor, as can be seen in Figure 6 (addressing task T2). The editor can operate at two different levels of detail. The overview, which is shown by default, presents the currently set weighting of individual components as a stacked bar. Using drag-and-drop, users can directly manipulate the weights of individual components. The weight of a component is redundantly encoded in the length of the bar and its saturation, and additionally shown in a text label. After switching to the detail mode, users can change the DoI formula by adding and removing attributes, setting their value range, and by optimizing parameters of the exponential smoothing (see Section 4.1). Additionally, the user can invert the meaning of an attribute, which is particularly useful when positive values have a negative meaning associated (e.g., the lower a country's *interest rate*, the better).

DoI Streamgraph

To support the user in understanding the effects the DoI function has on the items in the *ThermalPlot*, we add item-specific streamgraphs to the detail view (addressing task T4). The streamgraph visualizes how much each attribute contributes to the aggregated DoI value over time. Figure 7 shows an item's streamgraph for the DoI settings defined in Figure 6. In the streamgraph, every contributing attribute is represented as a stream whose height is proportional to the weighted attribute value $w_i \times v_i(t)$. See Section 4.1 for further details on the computation of multiple-attribute DoI values. In the example in Figure 7, the *short-term*

Fig. 6: *Multi-attribute DoI editor with a stacked bar that encodes the weighting of individual components (a). Using the interface below, the user can add components, invert the semantic, and set the range for each attribute (b). Smoothing parameters are applied globally (c).*

interest rate attribute is always positive, while the *long-term interest rate* is mostly negative. In cases where multiple attributes contain positive and negative values, the individual streams can cross each other. The color of each stream corresponds to the color in the DoI editor bar.

Because of the layering principle, streamgraphs can by definition represent only positive values. However, to be able to encode both components with a negative and with a positive impact on the DoI, we decided to use a two-part representation: The parts above and below the baseline show the contributions of the attributes that exhibits positive and negative changes relative to the index point, respectively. Hence, the raw DoI value is the difference between the stream heights above and below the zero line.

The dotted line above the streamgraph representation indicates the smoothed DoI value for each time point. The smoothing effect is clearly visible when comparing the dotted line to the raw streamgraph values, which show small fluctuations of the attribute values.
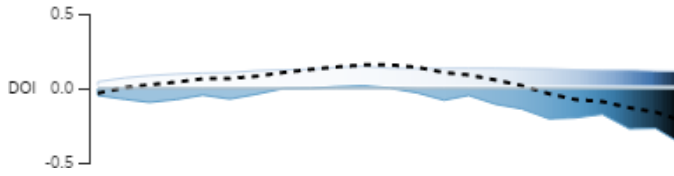
We additionally indicate how much individual time points contribute to the final smoothed DoI value by applying a horizontal color gradient: the darker the color, the higher the impact on the overall DoI.

In many real-world scenarios, missing data plays a role. An example is the closed stock market on holidays and weekends, where all associated attributes have no values for these days. Depending on the usage scenario, there are many ways of dealing with this problem. The missing values can be handled either in the data space, by applying forecasting or interpolation strategies, or in the view space, by clearly marking the missing data, for instance.

In *ThermalPlot* we apply a nearest neighbor interpolation to fill missing data values (see Section 5.3). However, it is essential that users are aware of the strategies applied and their consequences, as these can influence the results of multi-attribute DoI functions. To address this issue, we explicitly indicate interpolated values marked by a hatching pattern (see detail view in Figure 13).
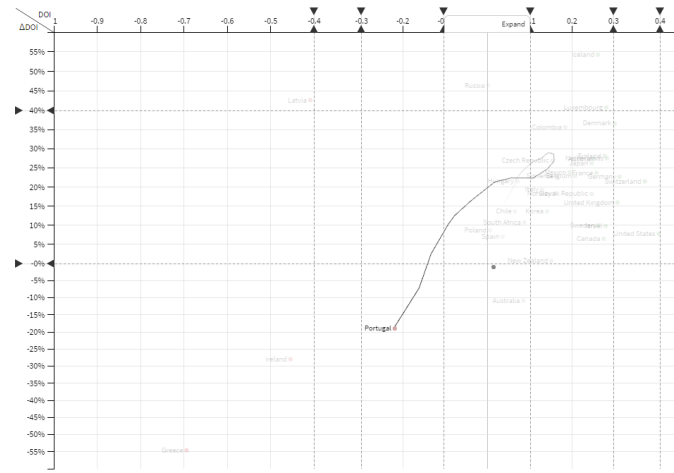
Trajectories

With the highest level of detail (see **L4** in Section 4.2) we add trajectories to the item representation in the *ThermalPlot*. A trajectory represents the item's path within the selected time window. The example in Figure 8 matches the DoI streamgraph

Fig. 7: *Streamgraph visualizing the contribution of each DoI component at each time point within the time window. Streams below the zero line indicate negative contributions, while the dotted black line indicates the smoothed DoI value.*

presented in Figure 7. The trajectories' opacity decreases over time (e.g., as in [29]), to allow correct interpretation of parts of trajectories in which the DoI remains relatively constant. In addition, the trajectories give analysts a static view of different thermal effects, such as loops and periods of rest, without the need for animation.

Fig. 8: *Trajectory of a selected item in the ThermalPlot describing the path an item takes through the ThermalPlot space over time. The transparency along the trajectory encodes the item's temporal development.*

Although the simultaneous display of trajectories of multiple items can easily clutter the representation (see Figure 9), it facilitates spotting items that behave very differently from the rest. However, to keep the visual clutter at a minimum, trajectories are disabled by default and shown for items that have been actively selected only.
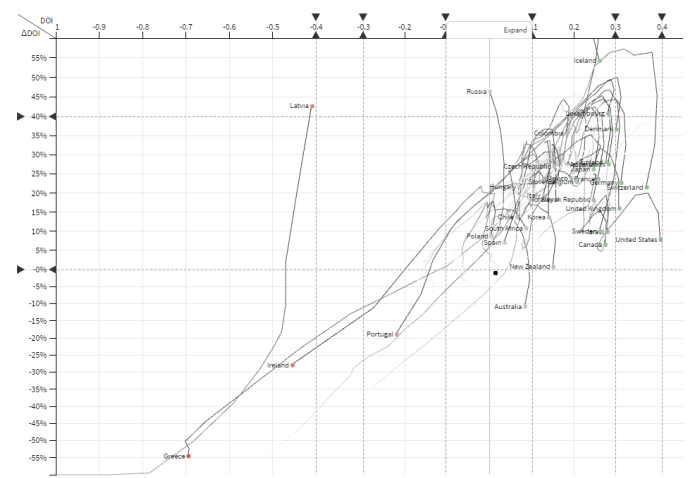
5.5 Implementation

The *ThermalPlot* prototype has a client-server architecture. The server component is based on the recently published *CloudGazer* infrastructure [32], which supports streaming of time-dependent data for multiple items and attributes. By using different data connectors, we can access either live data or data already collected from a database. The web-client uses AngularJS[4] for the overall management. Visualizations are implemented in D3 [5][5]. An interactive version of the *ThermalPlot* environment and a demonstration video is available on the accompanying website[6].

---

4. http://angularjs.org; used Angular v1.2.6
5. http://d3js.org; used D3 v3.5.6
6. http://thermalplot.pipes-vs-dams.at

Fig. 9: *Showing trajectories of multiple items simultaneously allows users to identify items that behave differently from the rest. However, as trajectories soon result in a cluttered representation, they are turned off by default and shown for selected items only.*

6 USE CASE

We demonstrate the value and utility of the *ThermalPlot* technique by applying it to a stock market data set that includes multiple attributes such as *trade volume*, *open/close price*, and *daily low/high* for all companies traded in the *Standard & Poor's 500* (*S&P 500*) index. We gathered the daily data via the *Yahoo Finance API*[7].

The use case is based on multiple analysis sessions with a consultant who invests parts of his private savings in stocks. Our domain expert is 32 years old and works for a company that specializes in economics and innovation policy consulting. The first phase of the analysis took place in mid-July 2015 using the *thinking aloud* method. The second phase was conducted in August 2015 using self-reporting with an analysis diary.

The expert usually checks his personal portfolio once or twice a week with the goal of making informed investment decisions. His current decision-making process is based on a combined investigation of the companies' recent developments on the stock market and quarterly published performance metrics (e.g., *dividend yield* and *earnings per share (EPS) growth*) that are available for all traded companies.

In his current workflow, he starts the analysis by investigating the static performance metrics from quarterly reports. As we have collaborated with him in a previous project, he already uses an interactive *LineUp* ranking visualization [14] that allows him to create a weighted combination of multiple static metrics to identify companies that could be underrated and thus interesting investment targets. He starts by going through the top-ranked companies and checks their recent development on the stock market by visiting online portals such as *Yahoo Finance*[8] or *finviz*[9]. However, collecting and matching this information from various sources is a tedious and time-consuming process.

ThermalPlot provides the expert with a big picture that summarizes the recent developments on the market (task T1). In order for *ThermalPlot* to show meaningful positions for the companies, the expert selects a time window and defines the DoI function. For

---

7. http://developer.yahoo.com/yql
8. http://finance.yahoo.com
9. http://finviz.com

Fig. 10: *ThermalPlot visualizing the recent development of all 500 companies from the S&P 500 index, according to the respective DoI computation. The treemap is used as an integrated overview visualization and shows all companies listed in the S&P 500 index. The annotations show the interpretation of our user for the different corner areas. Companies located in areas (a) and (c) are potential investment candidates.*

the stock market data, the timeline shows the overall development of the S&P 500 index from the beginning of 2015 to the last trading day—in this case August 14, 2015. He sets the time window to include all data from the last two months of trading, which in turn triggers loading of the corresponding data from the server, as shown in Figure 10. To make the prices comparable across companies, he sets the index point to the beginning of 2015 (see Section 4.1). Consequently, all companies with a considerable positive development since the start of the year appear to the right of the treemap, whereas companies with a negative development are positioned to the left (see Figure 10). For the DoI computation he chooses a weighted combination of the dynamic *closing price* (75%) and the static attributes *EPS growth* (10%), which indicates a positive business development, *return on equity* (10%), which indicates competitiveness from the shareholders' perspective, and *dividend yield* (5%), which indicates the "shareholder friendliness" (see the upper right corner of Figure 11 and addressing task T2).
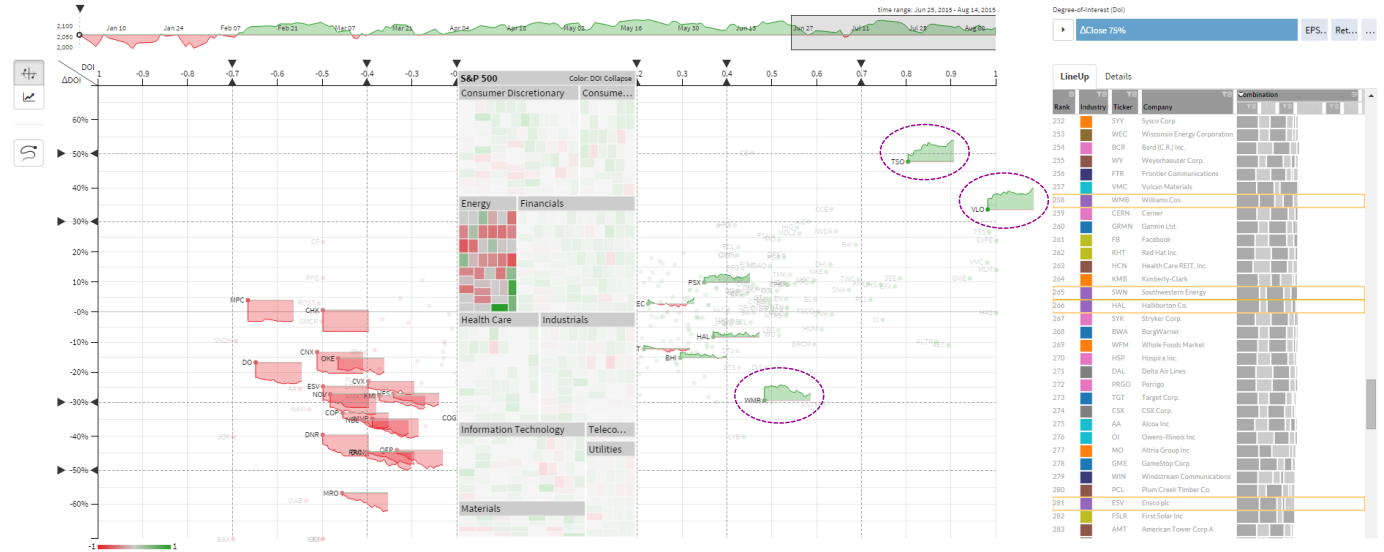
In the context of the stock market use case, the four corner areas have special meaning for the expert, as shown in Figure 10. The upper right area (b) contains strong companies with positive long- and short-term development. Consequently, this area would be the obvious investment choice, but comes with the risk of containing already overrated companies that could go down soon. Area (d) at the lower left contains companies with an ongoing negative development that would pose high-risk investment targets. Our expert is particularly interested in the remaining two areas: The lower right area (c) contains companies that developed well in the beginning of the selected time window, but show a

recent negative trend. The expert assumes that this could be only a minor short-term effect for a number of stocks and that their stock prices could rise again soon. The last area, on the upper left (a), contains companies with a negative development in the beginning but a recent positive trend. This could be caused by recent good news or reports, indicating a turnaround.

As an overview visualization, a centered treemap that covers the area defined by DoI values ranging from -0.2 to 0.2 is used. Companies in the treemap are grouped by industry sector according to the *Global Industry Classification Standard* (GICS) taxonomy[10]. The blocks representing the companies in the treemap are colored according to their DoI (see Section 4.3).

The expert starts the analysis by looking at the distribution of companies in the different areas of *ThermalPlot*. While the upper left area (a) is empty, the lower right area (c) contains three companies that he wants to investigate further: *WMB*, *LYB*, and *ANTM*. After selecting *WMB*, he sees that the company is part of the energy sector. Consulting the treemap, the majority of companies from the energy sector have performed poorly since the start of the year. As he closely follows the news and stock reports, this confirms what he already knows—there was a lot of pressure on the energy sector in the first half of the year. To take a closer look, he selects all companies in the energy sector by clicking the label of the sector in the treemap, as shown in Figure 11. While the companies in the lower left area are of no interest to the investor, he looks at *WMB*, which is a clear outlier located at the upper border of the lower right area (c), and investigates its

10. http://www.msci.com/products/indexes/sector/gics

Fig. 11: *Selecting the energy sector in the treemap highlights the stocks with the maximum detail level in the plot accordingly. At first glance, TSO and VLO are the obvious choice, but they might already be overrated according to our expert's own definition. Instead he investigates WMB that is located in his areas of interest (see Figure 10 (c)). According to the poor LineUp rank (to the right) and the loss in volume shown in the detail view (see Figure 13), he discards this company too.*
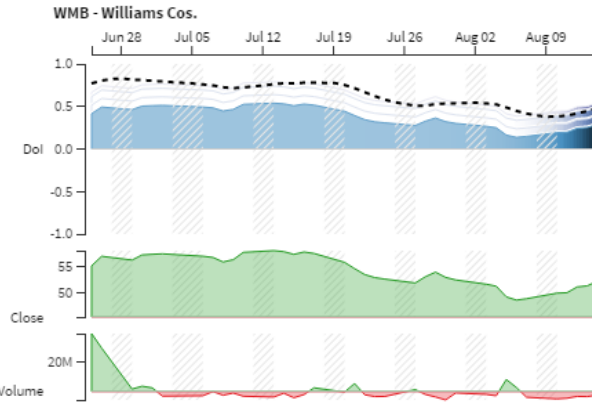

Fig. 12: *The expert selects the companies LYB and ANTM that are located in area (c) in Figure 10. He presumes that the stock price for companies in this area could raise again soon. He discards ANTM, due to the poor LineUp rank (> 100) and would rather invest in LYB, that can be found under the top 20 companies.*

performance by inspecting the combined streamgraph (task T4) and the individual attributes in the detail view (see Figure 13 and addressing task T5). He recognizes an odd peak in the *volume* line chart in late June. After a quick Internet search, he is able to attribute the peak to a dividend payout announcement. After finally checking the company's rank in the LineUp visualization, where it is ranked in the lower half (> 250 of 500), he decides to look for other investment options. Still focusing on the energy sector, the two outliers in the upper right area (b) grab his attention: *TSO* and *VLO*. However, as companies in this area may already be overrated according to his own definition, he decides to not invest in the energy sector and clear the selection. He continues by checking the development of *LYB* and *ANTM*, which are the

other two companies from his original selection (see Figure 12). A further look in *LineUp* reveals that *ANTM* does not rank highly (> 100) and does not meet his static performance metric criterion. However, *LYB* is among the top 20 companies according to the ranking and therefore seems to be a potentially lucrative investment. Using *ThermalPlot*, the expert was able to effectively identify one stock worth investing in from a large collection.

Informal User Feedback

The fact that the expert is now using *ThermalPlot* on a regular basis demonstrates that he deems the tool as a valuable addition to his current stock investment decision making process. He also mentioned that the new analysis workflow is much more elegant

Fig. 13: *The detail view shows the DoI streamgraph and line chart for the different attributes for the selected company WMB. The hatching pattern in the streamgraph and line charts encode missing values caused by weekends and holidays.*

than his former approach and that the technique does an excellent job in summarizing the current developments on the market.

However, he also noted that, as a post-analysis follow-up step, he still needs to check external sources and recent news to collect more evidence for backing up the investment decision he made based on *ThermalPlot*. Consequently, *ThermalPlot* cannot serve as a comprehensive tool for decision-making, but shows its value for monitoring the market and quickly identifying potential investment candidates. He further mentioned that it would make sense to extend the time window to include the last two years, as short-term fluctuations are not relevant to a private investor who is rather interested in long-term investments. He also noted that the trajectories are hard to interpret.

In the feedback sessions we discovered that it is difficult for the user to configure the DoI and understand the implications of the weighting and the exponential smoothing. To address these issues, an advanced editor for constructing the DoI function and further techniques for combining multi-attribute time-series data must be developed.

7 Discussion

Scalability

One of the most critical factors when it comes to judging the value of a novel visualization technique is its scalability. In this regard, the *ThermalPlot* suffers from the same occlusion problems as scatterplots. This can be particularly problematic for items that are shown in a higher level of detail, containing labels, line charts, and trajectories. We address the problem by giving users the possibility to change the level of detail combined with the option to spread and compress the visual space, as discussed in Section 4.2. The embedded and linked overview visualization also supports quick identification and selection of items. However, ultimately it is the user's responsibility to resolve visual clutter in densely populated areas of the plot. As future work, it seems promising to integrate strategies that automatically adapt the scales and representation borders. This way, the system could adapt to the current situation without active interaction by the user. This could be particularly valuable for live streaming scenarios, where items move through the *ThermalPlot* space, to provide users with an overview that enables them to quickly grasp the overall status of the items and how they behave over time. To reduce the visual clutter in

certain regions by adapting the layout and level of detail, it would be necessary to quantify the overlap of items in the respective regions of the plot. Although this adaptive behavior has potential to significantly reduce visual clutter, it is prone to get distracting or even confusing to users.

Still, the proposed measures would not prevent the overlap of items that have the exact same position, i.e., the same DoI and ΔDoI values. A possible countermeasure is to aggregate a group of individual items to a single surrogate item. A disadvantage is that users could then assume that all aggregated items have the exact same temporal development. However, due to the multi-attribute DoI function, the contribution that attributes have to the final DoI can vary over time. The same argument holds for trajectories, which could be considerably different due to the fact that only limited previous time points are taken into account for computing the final DoI (see Section 4.1 on exponential smoothing). To address these issues, advanced aggregating techniques for time-series data need to be developed. The integration of adaptive behavior and aggregation techniques are both interesting topics for future research, albeit beyond the scope of this paper.
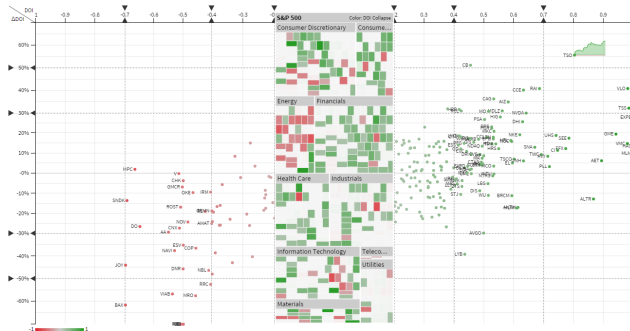
ThermalPlot vs. Line Chart

One of the most common visualization techniques for time-series data is a regular line chart with time mapped to the x-axis and values (e.g., DoI values) mapped to the y-axis (see Figure 14(b)). Line charts scale to only a small number of items and a small time window for data with high variance [18]. Following the trend of one particular line in a large collection is only possible using interaction, i.e., by selecting an item or following a line with the cursor. In contrast, *ThermalPlot* provides an overview that scales to many items. The performance of an item can be identified by its position (see Figure 14(a)). However, this technique comes with the disadvantage that time is encoded implicitly.

In order to select the same items in *ThermalPlot* and a line chart, the user needs to apply different interactions. Selecting all items within a specific DoI value range in the line chart corresponds to selecting a vertical slice in *ThermalPlot* (see Figure 15(a)). Selecting all items with a specific ΔDoI value requires to set an angular brush [15] in a line chart, whereas in *ThermalPlot* the selection is a horizontal slice of the plot (see Figure 15(b)). Selecting items within a specific DoI and ΔDoI range requires a combination of region select and angular brush in a line chart, whereas in *ThermalPlot* it can be achieved by making a rectangular selection (see Figure 15(c)). A lasso selection in *ThermalPlot* is even more flexible, as it allows the user to select items within an arbitrary region. To achieve the same selection in a line chart, however, a more complex series of interactions is required.
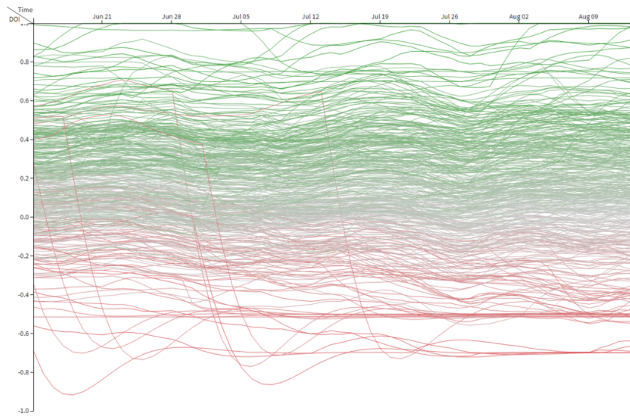
Trajectories

Trajectories provide an easy and static way for showing an item's position for all time points within the selected time window. In addition, they allow revealing periodic patterns, such as circles or recurring peaks (see Section 5.4). However, these patterns can be distorted in cases in which the optional orthogonal stretching is applied. Depending on the chosen configuration, circular patterns can look elliptical, for example. In *ThermalPlot*, we make the user aware of the distortion by adding grid lines to the background of the plot.

(a) *ThermalPlot* with DoI mapped to x-axis and ΔDoI to y-axis.

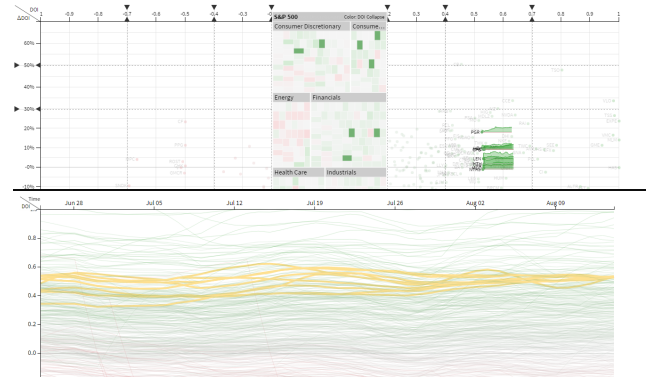(b) *Line chart* with time mapped to x-axis and DoI mapped to y-axis.

Fig. 14: *ThermalPlot provides a better overview for a large number of items, in this case companies from the S&P 500 index. Assertions about the trend of a certain item in the line chart is not possible. Only negative outliers (red), caused by stock splits, can be clearly identified and found in ThermalPlot at DoI = −0.5 (bottom).*
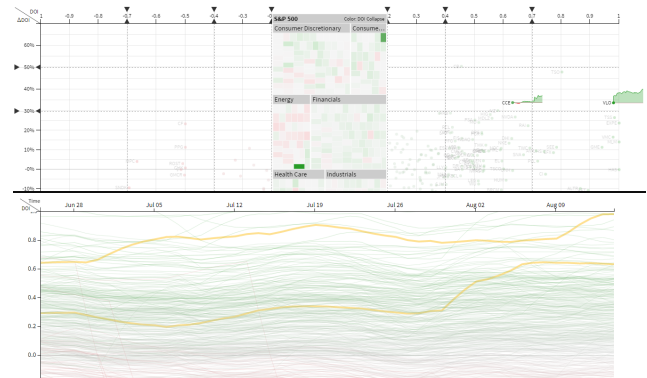
Granularity

In the visual analysis of time-dependent data, the level of granularity plays an important role [2]. In our stock market scenario, for example, we work with daily *closing prices*. However, the granularity could also be increased to one value per hour, minute, or second. For automated high-frequency trading programs, the level of granularity must be even higher. Granularity is a relevant factor to *ThermalPlot* because it determines the visual patterns and trends a user will see in the *ThermalPlot*. For instance, the granularity level has a large impact on the path of the trajectories. While a low sampling rate is sufficient for seeing macro patterns (e.g., visible as large loops in the trajectories), a higher frequency is required for micro patterns. Further, the smoothing algorithm and its parametrization have a large impact on the results. The appropriate level of granularity and the smoothing approach again depend on the specific analysis task.

Animation

By default, the selected time window in *ThermalPlot* is fixed to the user selection. However, the streaming capabilities of the implementation (see Section 5.5) also enable real-time data updates. In this case, the time window is shifted automatically to include the latest time point. Newly incoming streaming data triggers the re-computation of the user-selected DoI function (see Section 4.1),

(a) Selecting a **vertical slice** in *ThermalPlot* corresponds to a horizontal region selection in a line chart.

(b) Selecting a **horizontal slice** in *ThermalPlot* corresponds to an angular brush in a line chart.

(c) Selecting an **arbitrary region** in *ThermalPlot* corresponds to a combination of a region select and an angular brush in a line chart.

Fig. 15: *Different selection shapes in ThermalPlot and the corresponding results in a line chart.*

and cause the items' positions to be updated accordingly. However, if the time window is very small, covering only a few time steps, items can change their positions rapidly—making it hard to follow the position changes. Therefore, by choosing a reasonably large time window, it's the user's responsibility to control the change behavior.

Technical Considerations

Besides the visual scalability of the technique, we should also discuss briefly technical constraints comprising the performance in data loading, streaming, and caching. Naturally, the larger the user-selected time window, the more data needs to be transferred from

the server to the client. When selecting a time window $[t_s, t_e]$ with $\Delta t = t_e - t_s$, the actual required data time window is $[t_s - \Delta t - k, t_e]$. The additional history is needed for computing the ΔDoI value for the start point of the selected time window t_s (see Section 4.1 for details on the ΔDoI computation).

We indicate the data loading progress by gradually filling up the time window representation from left to right. In addition, the opacity of the *ThermalPlot* is substantially decreased and the thermal space is overlaid with an animated progress icon. Even when data loading is in progress, the user can follow the narrative resulting from the movements of the items within the selected time window in the *ThermalPlot*.

8 CONCLUSION AND FUTURE WORK

We have presented *ThermalPlot*, a scalable visualization technique for exploring multi-attribute time-series data. We use the position—the strongest visual variable—to encode item importance according to the DoI value in the horizontal direction and according to the change in DoI value (ΔDoI) in the vertical direction. This mapping allows users to see effectively the development of attributes over time at a glance. We introduce several scalability concepts and support views, including a timeline, a DoI editor, and a detail view. We have introduced the *ThermalPlot* technique using two data sets with different scale and complexity. We evaluated the technique by means of a stock market use case and expert feedback.

The *ThermalPlot* technique allows for monitoring multi-attribute time-series data by combining short-term and long-term value developments into a salient visual representation. Animation and trajectories can help the user to analyze how an item developed over time. On top of that, dedicated support for pattern search could be a fertile area for future work. Apart from patterns generated by the path of an item's position, value patterns over time such as 'down-up-down' are not specifically supported yet but are useful in a number of application domains. Consequently, better support for pattern search and exploration is another next step for future work towards a comprehensive tool for analysis and exploration of multi-attribute time-series data.

In addition to exploring historical and live data, we plan to integrate forecasting algorithms that can indicate possible future positions of items in the plot. This could be intuitively encoded by expanding the trajectories we currently use for showing the items' development over time. However, ways of encoding the introduced uncertainty must be taken into account.

In the presented exploration environment, items are part of a larger collection without relationships between them. In real-world scenarios, however, items often influence each other and therefore cannot be treated independently. In the financial market, for instance, a crash of a single company can have a negative impact on a large group of stakeholders, such as suppliers, customers, and shareholders. Other examples are the monitoring of IT networks, where a problem in the infrastructure can propagate to other parts of the network [32] and the exploration of biological pathways, where cellular effects can influence reactions further downstream [24]. As part of future work, we plan to investigate ways to conceptually integrate item relationships into the *ThermalPlot* environment. The challenges are manifold, ranging from the fact that the relationships can also change over time, to complex propagation effects. These are promising research directions which can help to understand complex dynamic, heterogeneous networks.

REFERENCES

[1] J. Abello, S. Hadlak, H. Schumann, and H. Schulz. A Modular Degree-of-Interest Specification for the Visual Analysis of Large Dynamic Networks. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 20(3):337–350, 2013.

[2] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Springer, 2011.

[3] M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets. In *Proceedings of the IEEE Conference on Visualization (Vis '96)*, 1996.

[4] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. ESRI Press, 2010. First published in French in 1967.

[5] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2301–2309, 2011.

[6] L. Byron and M. Wattenberg. Stacked Graphs - Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1245 –1252, 2008.

[7] A. Carvalho, A. S. A. De, C. Ribeiro, and E. Costa. A Temporal Focus + Context Visualization Model for Handling Valid-Time Spatial Information. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 7(3-4):265–274, 2008.

[8] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[9] D. Fisher. Animation for Visualization: Opportunities and Drawbacks. In *Bautiful Visualization*, volume Chapter 19, pages 329–352. 2010.

[10] J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, and P. Isenberg. Evaluation of alternative glyph designs for time series data in a small multiple setting. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*, pages 3237–3246. ACM, 2013.

[11] G. W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '86)*, pages 16–23. ACM, 1986.

[12] G. W. Furnas and B. B. Bederson. Space-scale diagrams: understanding multiscale interfaces. In *Proceedings on the Conference on Human Factors in Computing Systems (CHI '95)*, pages 234–241. ACM Press / Addison-Wesley Publishing Co., 1995.

[13] E. J. Gardner. Exponential smoothing: The state of the art–Part II. *International Journal of Forecasting*, 22(4):637–666, 2006.

[14] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2277–2286, 2013.

[15] H. Hauser, F. Ledermann, and H. Doleisch. Angular Brushing of Extended Parallel Coordinates. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '02)*, pages 127–130. IEEE, 2002.

[16] J. Heer, N. Kong, and M. Agrawala. Sizing the Horizon: The Effects of Chart Size and Layering on the Graphical Perception of Time Series Visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, pages 1303–1312. ACM, 2009.

[17] J. Heer and G. G. Robertson. Animated Transitions in Statistical Data Graphics. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1240–1247, 2007.

[18] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '10)*, 16(6):927–934, 2010.

[19] D. A. Keim, M. Ankerst, and H.-P. Kriegel. Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data. In *Proceedings of the IEEE Conference on Visualization (Vis '95)*, pages 279–286. IEEE, 1995.

[20] B. Kondo and C. Collins. DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '14)*, 20(12):2003–2012, 2014.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2015.2513389, IEEE Transactions on Visualization and Computer Graphics

PRE-PRINT VERSION   14

[21] S. Kriglstein, M. Pohl, and M. Smuc. Pep Up Your Time Machine: Recommendations for the Design of Information Visualizations of Time-Dependent Data. In W. Huang, editor, *Handbook of Human Centric Visualization*, pages 203–225. 2014.

[22] T. Lammarsch, W. Aigner, A. Bertone, J. Gartner, E. Mayr, S. Miksch, and M. Smuc. Hierarchical Temporal Patterns and Interactive Aggregated Views for Pixel-Based Visualizations. In *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, pages 44–50. IEEE, 2009.

[23] S. T. Lei and K. Zhang. A visual analytics system for financial time-series data. In *Proceedings of the Symposium on Visual Information Communication (VINCI '10)*, page 20. ACM, 2010.

[24] A. Lex, C. Partl, D. Kalkofen, M. Streit, S. Gratzl, A. M. Wasserman, D. Schmalstieg, and H. Pfister. Entourage: Visualizing Relationships between Biological Pathways using Contextual Subsets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2536–2545, 2013.

[25] J. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.

[26] Martin Wattenberg. Visualizing the Stock Market. *Extended Abstracts on Human Factors in Computing Systems (CHI '99)*, pages 188–189, 1999.

[27] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: Interactive Visual Exploration of System Management Time-Series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, pages 1483–1492. ACM, 2008.

[28] A. Rind, W. Aigner, S. Miksch, S. Wiltner, M. Pohl, F. Drexler, B. Neubauer, and N. Suchy. Visually Exploring Multivariate Trends in Patient Cohorts Using Animated Scatter Plots. In M. Robertson, editor, *Ergonomics and Health Aspects of Work with Computers*, volume 6779 of *Lecture Notes in Computer Science*, pages 139–148. Springer, 2011.

[29] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1325–1332, 2008.

[30] M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the rubber sheet: A metaphor for viewing large layouts on small screens. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '93)*, pages 81–91. ACM, 1993.

[31] T. Schreck, T. Tekušová, J. Kohlhammer, and D. Fellner. Trajectory-based visual analysis of large financial time series data. *ACM SIGKDD Explorations Newsletter*, 9(2):30–37, 2007.

[32] H. Stitz, S. Gratzl, M. Krieger, and M. Streit. CloudGazer: A Divide-and-Conquer Approach for Monitoring and Optimizing Cloud-Based Networks. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis '15)*, pages 175–182. IEEE, 2015.

[33] T. Tekusova and J. Kohlhammer. Applying animation to the visual analysis of financial time-dependent data. In *IEEE Transactions on Visualization and Computer Graphics (InfoVis '07)*, pages 101–108. IEEE, 2007.

[34] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. A Survey on Interactive Lenses in Visualization. In *Proceedings of the Eurographics Conference on Visualization (EuroVis '14) – State of The Art Reports*, pages 43–62. Eurographics, 2014.

[35] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2nd edition, 1983.

[36] F. van Ham and A. Perer. "Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):953–960, 2009.

[37] A. Vande Moere. Time-Varying Data Visualization Using Information Flocking Boids. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '04)*, pages 97–104. IEEE, 2004.

[38] T. von Landesberger, S. Bremm, P. Rezaei, and T. Schreck. Visual analytics of time dependent 2d point clouds. In *Proceedings of the Computer Graphics International Conference (CGI '09)*, pages 97–101. ACM, 2009.

[39] M. O. Ward. Multivariate data glyphs: Principles and practice. In *Handbook of Data Visualization*, pages 179–198. Springer, 2008.

[40] P. R. Winters. Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science*, 6(3):324–342, 1960.

[41] H. Ziegler, M. Jenny, T. Gruse, and D. Keim. Visual market sector analysis for financial time series data. pages 83–90. IEEE, 2010.

Holger Stitz is currently working as PhD student at Johannes Kepler University Linz Austria in the visualization group of Marc Streit at the Institute of Computer Graphics. His main research interest are multi-attribute time-series data in the context of large networks. For more information see http://holgerstitz.de.

Samuel Gratzl is a PhD student at Johannes Kepler University Linz, Austria. He is associated with the Institute of Computer Graphics working at Marc Streit's visualization group. His main research field is guided visual exploration of heterogeneous biomedical data. For more information see http://sgratzl.github.io.

Wolfgang Aigner is scientific director of the Institute of Creative\Media/Technologies at St. Poelten University of Applied Sciences, Austria. His main research interests include visual analytics and information visualization, particularly in the context of time-oriented data. Wolfgang holds a PhD from Vienna University of Technology, Austria (2006) where he also received his habilitation in computer science in 2013. For more information see http://mc.fhstp.ac.at/people/wolfgang-aigner.

Marc Streit is assistant professor at the Institute of Computer Graphics at Johannes Kepler University Linz, Austria. He finished his PhD at Graz University of Technology in early 2011. In 2014 he received a Fulbright scholarship for research and lecturing at the Harvard University. His areas of interest include visualization, visual analytics, and biological data visualization. For more details see http://marc-streit.com.