ORNL/TM-2016/393

Falcon: A Temporal Visual Analytics System Applied to the Analysis of 3D Printer Log Data

Chad A. Steed
Ryan Dehoff
William Halsey
Sean Yoder
Vincent Paquit
Sarah Powers

August 2016

Approved for public release. Distribution is unlimited.

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2016/393

Computational Sciences and Engineering Division
Materials Science and Technology Division
Electrical and Electronics Systems Research Division
Computer Science and Mathematics Division

Falcon: A Temporal Visual Analytics System Applied to the Analysis of 3D Printer Log Data

Chad A. Steed
Ryan Dehoff
William Halsey
Sean Yoder
Vincent Paquit
Sarah Powers

Date Published: August 2016

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by

UT-Battelle, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ORNL	Oak Ridge National Laboratory
MDF	Manufacturing Demonstration Facility
CAD	Computer Aided Design
IR	infrared
EDEN	Exploratory Data analysis ENvironment
3D	three-dimensional
CSV	comma separated value

ACKNOWLEDGMENTS

This research is sponsored by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Advanced Manufacturing Office, under contract DE-AC05-00OR22725 with UT-Battelle, LLC and earlier by the Oak Ridge National Laboratory LDRD project No. 7409.

ABSTRACT

Flexible visible exploration of long, high-resolution time series from multiple sensor streams is a challenge in several domains. In the field of additive manufacturing, this capability is of paramount importance to realizing the full potential of large-scale 3D printers. In this report, we propose a visual analytics approach that helps researchers acquire a deep understanding of patterns in log and imagery data collected by 3D printers. We introduce Falcon, a new visual analytics system that allows users to interactively explore large, time-oriented data sets from multiple linked perspectives. Falcon provides overviews, detailed views, and unique segmented time series visualizations with multiple levels of detail. These capabilities are applicable to the analysis of any quantitative time series. To illustrate the effectiveness of Falcon at providing thorough and rapid knowledge discovery, we present a practical case study involving experts in additive manufacturing and data from a large-scale 3D printer.

1. INTRODUCTION

The ability to discover patterns in time series data is a fundamental requirement in many domains. With small to moderate size data sets involving a few variables and regular sampling intervals, basic graphs and statistical calculations are effective at revealing important features. However, these techniques are inadequate for analyzing time series streams from multiple sensors that are long (multiple days), large (millions of data points), multivariate, and irregularly sampled. Such is the scenario faced daily by researchers in the flourishing field of additive manufacturing, where large-scale 3D printers are used to synthesize complex objects for industrial purposes.

Recent advances in additive manufacturing have improved production value by removing traditional manufacturing constraints and providing unprecedented geometrical freedom to designers. The Oak Ridge National Laboratory (ORNL) Manufacturing Demonstration Facility (MDF) is at the forefront of this disruptive technology. Using multiple large-scale 3D printing systems, such as the Arcam Q10 system shown in Fig. 1, MDF researchers design and execute builds of complex objects and study both the printer log files and microstructure of the printed objects to make fundamental scientific contributions related to the efficacy of the 3D printing process. The results of these investigations have also enabled the construction of unique prototypes, namely aerospace components, advanced robotics, and automobiles.

The key to realizing the full potential of additive manufacturing lies in providing researchers with intuitive tools that support exploratory analysis of sensor data from 3D printing systems. However, the size and complexity of these data exceed the capabilities of most general purpose data analysis systems. To address this capability gap, we have developed the Falcon visual analytics system in close collaboration with additive manufacturing experts. Although we apply Falcon to additive manufacturing in the current work, it is applicable to any problem that involves the analysis of quantitative, time series data.

From a visual analytics perspective, Falcon combines several interactive data visualization techniques that extend the scalability of traditional time series analysis methods. The system design is inspired by the visual information seeking strategy [27] where zooming and filtering operations permit the descent from overviews to detailed data visualizations. Using both time-oriented and statistical views, Falcon coordinates user interactions in multiple visualizations to allow comparative analysis of long time series data that are sampled irregularly with subsecond precision. The system also utilizes the concept of information scent [39] derived from time series similarity algorithms to guide the user to the most

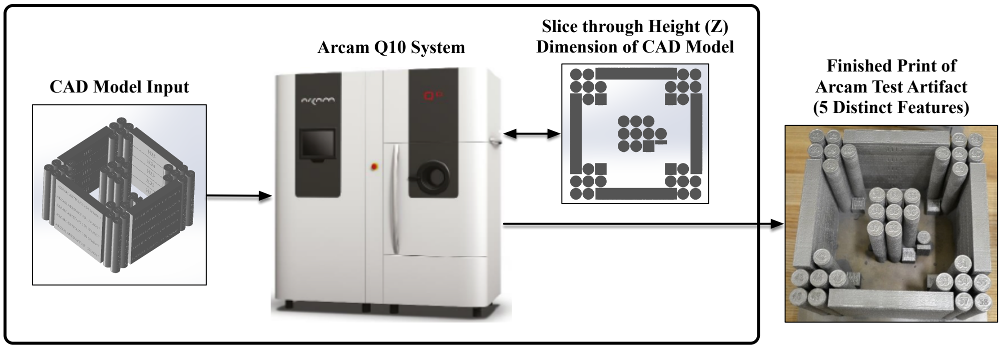

Fig. 1. The Arcam Q10 3D system is a large-scale 3D printer that uses electron beam melting to build metal objects. Each layer is melted using the geometry defined by slices taken through a CAD model progressing from the bottom to the top of the build.

interesting relationships. The coalescence of these capabilities into a visual analytics system helps researchers develop a deeper understanding of 3D printing systems while reducing knowledge discovery timelines—the central promise of visual analytics.

1.1 Contributions

As a visual analytics design study [26], this report describes a system that addresses a specific domain problem and has evolved based on an iterative design process with collaborators from that domain. The main contributions of this report include the following:

- We review the Falcon system design rationale. Falcon goes beyond previous systems to support scalable exploratory analysis of long and complex time series data streams involving multiple variables.
- We present a unique combination of interactive visualization techniques and an automatic recommender that identifies similar time series patterns. We also describe a new visualization technique, called the waterfall visualization, that combines overview and detail using miniaturized graphics.
- We describe the Falcon system in a real case study involving 3D printer data and utilizing workflows developed by domain experts. To the best of our knowledge, this report documents the first application of visual analytics to the field of additive manufacturing.
- Based on the case study and evaluation of Falcon's usage by experts, we discuss ideas to improve Falcon. In addition, we reflect on our experience as an interdisciplinary design team.

1.2 Outline

After a survey of related work in Sec. 2., we describe the motivating use case in Sec. 3.. The design rationale of Falcon is presented in Sec. 4., followed by a description of the user interface in Sec. 5.. In Sec. 6., we present a real case study in which 3D printer log data is analyzed using Falcon. The findings from this case study have led to new knowledge about the 3D printing process. In Sec. 7., we reflect on important design revisions and our experience as an interdisciplinary team. Finally, in Sec. 8. we summarize our results.

2. Related Work

Due to the copious nature of time-oriented data, a large body of work exists on time-based visualization techniques as evidenced by recent reviews of the field [1, 2]. Some systems are designed with a very specific use case in mind in order to maximize the discovery of time-based insight. For example, many social media visual analytics systems (e.g. Leadline [10], Matisse [28], Visual Backchannel [9]) incorporate some form of time-based visualization as the central view with other linked views serving to supplement the temporal patterns. Social media visual analytics systems are a specific form of text visualization systems (e.g., ThemeRiver [14], EventRiver [21], and TextFlow [8]), which also tend to include a prominent time-based visualization component. But time-based analysis permeates nearly all domains, namely, climate [19, 29], cyber security [12], and parallel computing performance monitoring [13].

In addition to domain based systems research, a substantial segment of time-based visualizations are devoted to generic techniques that can be applied to a range of data sets. Most of these techniques represent time with line plots or bar charts with either variations on the graphical representation or enhancements that allow interactive visual queries. Examples of line plot variations that are designed to increase the number of comparable time series include small multiples [32] and sparklines [31], horizon graphs [15], and braided graphs [17].

To provide multi-scale views, van Wijk [38] used a calendar-based visualization to analyze time series data aggregated on a daily, weekly, or monthly basis via a similarity clustering method. Spiral layouts of the time axis are used in the SpiralGraph [36] and SpiralView [5] techniques. These spiral layouts perform poorly with long time series, but excel at showing recurring patterns. VizTree [20] presents a very unique representation of time series data that uses a sequence of symbols and a suffix tree. Although VizTree may be helpful for very large data sets, the view can be difficult to decode, especially for fledgling users.

Buno et al. [7] introduce a visualization technique in the TimeSearcher 2 system that helps users see statistical summaries of time series data. The method renders the minimum and maximum range for each time record and a central line to show the mean value. In Falcon, we use an extended version of this statistical representation to visualize an overview of time series data. The TimeSearcher 2 shows multiple time series simultaneously and provides the ability to search for similar patterns. Falcon also offers simultaneous views, pattern matching, and it supports the visualization of irregular sampling intervals.

Some time series visualizations use lens-based techniques to magnify time ranges of interest by distorting the time axis. Kincaid et al. [18] introduced SignalLens which magnifies an area of interest and compacts the areas on either side to maintain context. Walker et al. [34] introduced the RiverLens technique, which combines the SignalLens [18] and the River Plot [6]. Brushing time ranges expands the time series plot and shows an overlay on the River plot. A River Plot is shown on either side to provide context. Introduced by Zhao et al. [40], ChronoLens is an interactive visual analytics system that includes a lens-based technique to support elaborate analysis tasks. Traditional time series plots are augmented with lens distortions based on user selections. In addition, derived data, such as derivatives and moving averages, can be shown on the graphs and zooming, resizing, and movement operations are applied to the lens to alleviate occlusion.

Hao et al. [11] introduce an interest-based visualization using a layout of time series plots that reveals important hierarchical relationships. Similarly, Stack Zoom is a multi-focus zooming techniques described by Javed et al. [16] that maintains context and temporal distance during zoom operations. User selections produce a hierarchy of zoomed line graphics that are represented in a nested tree layout, which also serves as a graphical history.

Instead of lens-based techniques, Falcon uses multiple views and multi-scale zooming due to problems that may occur with distorting the time axis. Plaisant et al. [23] introduce the concept of overview and detail displays, which provide simultaneous views of a focus visualization and an overview of the entire data series. The overview provides context for the focus visualization and users can brush areas of interest for detailed investigation. Falcon also provides zoom-and-filter interactions in the time series plots and fine-grained control over the time scale and level of detail used in the visualization. As the zoom level is increased, the width of the time series graph is expanded while the display viewpoint remains constant. Scroll bars allow the user to freely navigate the expanded time series.

3. Motivating Use Case

The design of Falcon began as a collaborative endeavor between visual analytics, data science, and additive manufacturing researchers at the ORNL MDF. The initial objective of this project was to evaluate the potential of visual analytics tools at revealing new insight about the 3D printing process using both log data and near infrared (IR) imagery from large-scale 3D printers at the ORNL MDF. More specifically, we focused on the Arcam Q10 system that is shown in Fig. 1.

As depicted in Fig. 1, the printing process (also known as a build) begins when a Computer-Aided Design (CAD) model is uploaded to the system. Beginning at the bottom, the system prints each layer by extracting slices through the height (or z) dimension of the model. For each layer, the system prepares a layer of metal powder 50 microns (0.05 mm) thick and completes a sequence of 5 process or heating steps. During the melt stage, an electron beam is used to melt in the desired shape of the layer as defined by the model. This process enables the production of parts with very complex geometries that do not require additional tooling or fixtures and it avoids the production of significant amounts of waste material.

Initially, we analyzed the data using the multivariate data analysis capabilities of EDEN [30], a parallel coordinates based visual analytics system. Additive manufacturing researchers uncovered several correlations between variables by comparing statistical aggregations at different scales from a large repository of experimental 3D builds. Although these initial investigations were successful at demonstrating the potential of visual analytics tools at improving the depth of analysis, we observed some limitations. First, the parallel coordinates plot in EDEN made it difficult to focus on temporal patterns in detail. This issue was exacerbated by the long duration of the 3D prints, which often require multiple days to complete, and the fine-grain sampling of many of the logged sensor readings. Such long time series are difficult to analyze in a parallel coordinate plot due to condensed representation of the time variable. Furthermore, the scientists sought a visual representation of the data that permitted multiple temporal viewing scales.

Another issue was related to the system's reporting of individual sensor and process control module readings at irregular intervals, usually only when a change in value occurs. Consequently, many time instants in the file contain a single entry, which makes it difficult to construct a complete polyline of the detailed data values in the standard parallel coordinates plot. To create complete polylines, we aggregated values for both entire builds and specific layers of individual builds. Despite the limitations, the initial study demonstrated the promise of a visual analytics approach and resulted in a follow-on project to design and implement a more effective system.

4. Task Abstraction and Design Rationale

The Falcon system is designed to support the task of visually exploring streams of long, multivariate time series to find interesting patterns in both the temporal sequence and statistical distribution of data values. In particular, we target the analysis of log data generated by large-scale 3D printers to gain a deeper understanding regarding the quality of 3D-printed objects. With this task in mind, our interdisciplinary design team, which includes additive manufacturing experts who regularly use Falcon to analyze their data, have identified the primary design requirements that are described in the remainder of this section.

Time-based Representations: Time is the primary dimension for analyzing log data, especially with 3D printing systems where known event sequences are studied (e.g., initialization, build layers, material preparation). Thus, additive manufacturing researchers usually begin with time-based data visualizations. These data visualizations map the time instant of each data value to some visual property of the display (e.g., a point along a horizontal line). To support time-based pattern recognition, Falcon includes multiple visual encoding techniques that emphasize the relative positions and changes for temporal sequences of data values.

Statistical Representations: Descriptive statistics and their visual representations are vital to detecting broad trends from large data sets, and additive manufacturing researchers commonly use them to study the 3D printer logs. They primarily rely on visualizations of value distributions and descriptive statistics. In addition to broad trends, visual statistical representations help scientists identify normal variation ranges and detect outliers. For large data sets, the level-of-detail in statistical summaries can be adjusted to allow researchers to progress from overviews to more detailed views. Falcon includes visual representations of these key summary statistics in the form of histograms and modified box plot visualizations.

Linked Representations: Researchers often view and interact with different visualizations of the same data independently, which can delay the formulation of insight across multiple views due to human memory limitations from one glance to the next [24]. Linking techniques are used to reduce this issue by interactively relating information between different views [35, 22]. Brushing and linking is the most common view coordination strategy [4]. Here interactive selections in one view are propagated to other views, where corresponding items are highlighted automatically. The linkage helps users form a more complete mental model of the data by combining input from different representations that emphasize particular features. Falcon harnesses the power of linked visualizations by coordinating interactions between both different scales (overview and details) and different representations (time-based and statistical).

Show Multiple Scale Views: Large-scale 3D printers require several hours and sometimes multiple days to finish a build. In analyzing the data, researchers are interested in patterns that are visible at different time scales. Macroscale features, such as the relative duration required to print each layer, are visible in overviews of the entire data set. On the other hand, microscale features, such as system circuit resets, only appear at the highest levels of granularity. Therefore, the Falcon design includes overviews and detailed

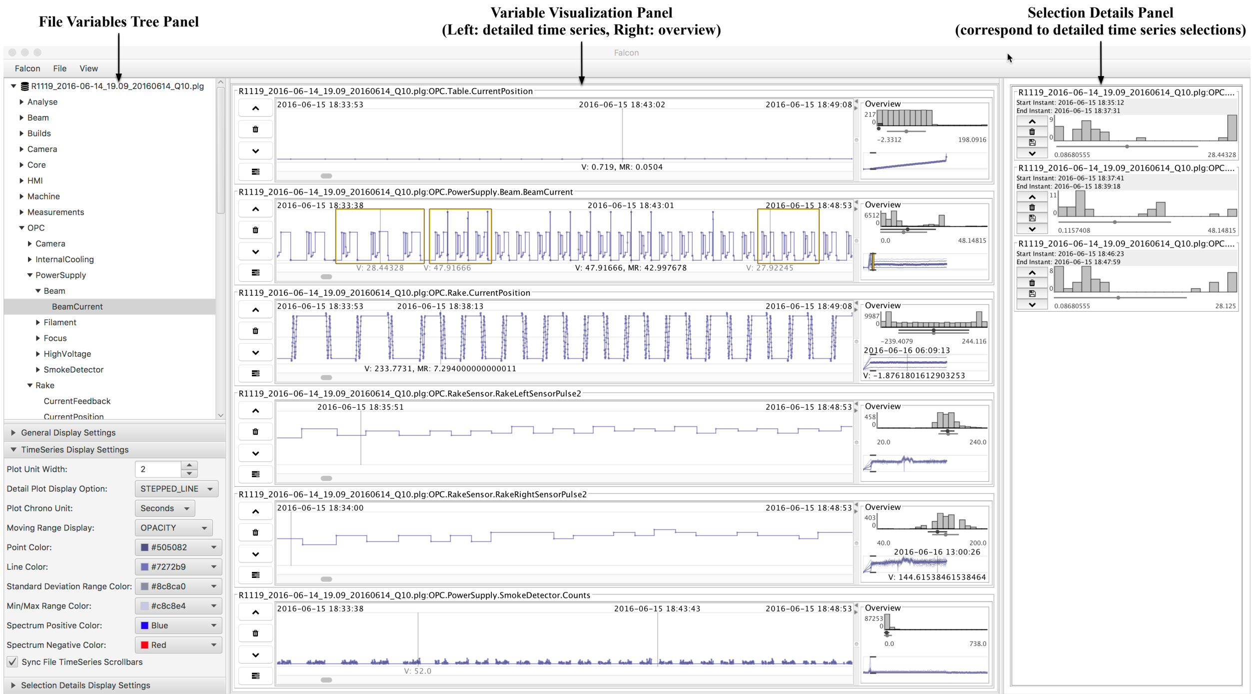

Fig. 2. Falcon is built around a set of highly interactive data visualizations that allow users to explore temporal and statistical patterns in long and complex time series data. The center variable visualization panel features both overview and detail visualizations. When time range selections are set (see the three rectangles with yellow borders in the *BeamCurrent* variable pane), a statistical summary pane is added to the selection details panel at right to summarize the selected data values.

views with linked interactions. In addition to showing broad trends, the overviews help the user maintain the context for selections in other detailed views. In the detailed views, researchers can interactively adjust the level-of-detail using the hierarchical nature of time units (e.g., hours, minutes, seconds). This capability provides free form access to the most appropriate time scale for particular features.

Support Human Interaction: Due to the emerging nature of the field, the questions asked of 3D printing data are often too exploratory for a completely scripted analytic workflow. Researchers require tools that allow them to interactively ask questions of their data, while not confining them to the original ideas that prompted the data collection. Indeed, historical reflections upon some of the most significant scientific discoveries show that profound findings are often unexpected (e.g., Pasteur's immunization principles, Columbus' discovery of America) [3]. Furthermore, the exploration process is iterative as new discoveries lead to new questions and the analytical discourse continues in a cycle until the researcher is satisfied. Falcon is designed to support human-centered interactions that give the researcher control over the direction of the analysis process. Interactions are coordinated across multiple views of the data to allow the scientist to highlight certain patterns from different perspectives. Furthermore, these interactions feed automated analytical algorithms to highlight similar time ranges inferred from user actions.

5. Introducing Falcon

The Falcon visual analytics system combines interactive data visualization, automated statistical analytics, and data processing routines. Together, these components form a human-centered interface that supports the formulation and confirmation of hypotheses in complex time series. Due to the platform independence, mature graphics capabilities, and scalable performance, Falcon is implemented entirely in the Java programming language. In the remainder of this section, we provide a technical description of the main features that comprise Falcon.

5.1 Data Processing and Organization

Before analysis can begin, the log data must be converted from unstructured text into an organized form. Although it is possible to dynamically ingest streams of data, the logs are typically read from static files in a post-processing mode. The 3D printer logs contain readings from thousands of different sensors and diagnostic modules. The readings are captured and stored during the print, which may take several hours and sometimes multiple days to complete. The data values are stored with time stamps at sub-second accuracy and irregular time sampling intervals. When the system begins a new build, an initial reading for each variable is recorded in the log. Thereafter, new variable values are only recorded as changes occur. Consequently, except for the first time stamp, most time stamps recorded in the file will have only one variable value. Furthermore, some variables contain tens of thousands of values, others contain hundreds, and some may have less than 10. In addition to the log data, the printer captures a single near IR image at the completion of each build layer.

As Falcon is designed to support any time-based data, we include an option to ingest data in the comma separated value (CSV) format. This capability allows exploration of data from other systems without requiring modifications to Falcon. Alternatively, custom ingest modules can be developed to improve efficiency.

The time stamp for each value is used to index the data in a tree-based data structure and allow efficient access. Except for metadata information, data for a particular variable are read into the system memory only as the user adds the variable to the visualization. This practice reduces the memory footprint as most users confine their analysis to a particular set of variables. Also, by scanning the log file initially, the caching mechanism in the operating system will ensure fast access on future read operations. In addition to the time-based indexing, descriptive statistics (e.g., mean, standard deviation, median, and inter-quantile range) are calculated and stored for each variable. The system also computes histogram bin information in a frequency-based data structure.

Both the frequency-based and time-based data structures support multi-scale temporal binning at different levels of detail. For example, the system can create a hierarchy of time-based bins at hour, minute, or second intervals. The multi-scale summaries allow users and algorithms to descend to more granular time measurements for greater fidelity as needed. In addition to the statistical summaries, the raw variable records are accessible.

As shown in Fig. 2, Falcon lists the available file variables in a tree panel on the left side of the window. With 3D printer logs, variables names are organized hierarchically by separating the categories with a period character. For example, the variable selected in the tree view shown in Fig. 2, is stored as

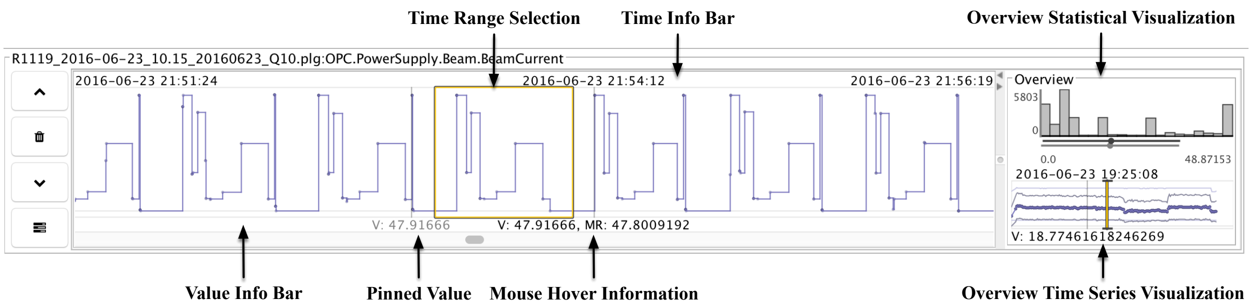

Fig. 3. This figure shows a single variable visualization pane for the *BeamCurrent* variable. At right, two overview visualizations provide statistical (top) and temporal (bottom) context for the main detail time series visualization at left. The detail time series visualization includes an interactive time range selection capability and details-on-demand features through mouse gestures.

`Builds.State.CurrentBuild.CurrentZLevel` in the log file. Falcon parses the variable names in this form to build the levels of the tree view. When analyzing hundreds of variables, the hierarchical organization helps users search more efficiently. When variable names do not use this hierarchical organization, the tree will only have one level, below the file node. For comparative purposes, Falcon also supports loading multiple files in the same session.

5.2 Variable Visualization Panel

The focal point of the Falcon user interface is the variable visualization panel (see Fig. 2), which provides multiple views for a specific variable. When a variable is selected, a new horizontally-oriented visualization pane is added to this panel. All variable panes currently under investigation are stacked vertically in the scrollable panel. The order of the panes and their removal from the panel can be controlled using the buttons on the left side. As shown in Fig. 3, each pane consists of a detail time series visualization (left) and two overview visualizations (right). These interactive views are linked so that brushings and other manipulations are shared.

5.2.1 Variable Overview Visualizations

Fig. 3 shows the overview region of the variable pane, which consists of two visualizations: a statistical visualization (top) and a time series visualization (bottom). The statistical visualization displays the frequency distribution and summary statistics of the entire variable distribution. We use a standard histogram plot, which counts the number of values that fall within equally sized bins over the distribution space. From the settings panel, the user can change the number of bins, plot height, minimum and maximum values of the x-axis scale, and maximum count value of the y-axis scale.

As shown in Fig. 4, the bin counts are visually encoded as the height of the bars in the histogram plot. The result is an overview of the distribution for the variable of interest. To access detailed information, the user can hover over a bin (see Fig. 4) to see a tooltip with the numerical values and the bin's upper and lower limits. As the visible time range of the detail time series visualization changes, the histogram plot receives

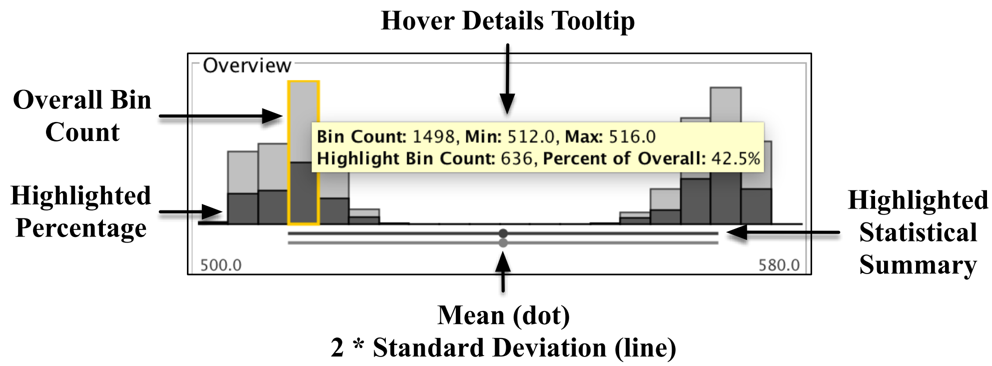

Fig. 4. The statistical view in Falcon encodes the frequency distribution of values using a histogram. The darker portions encode the percentage of values shown in the visible portion of the detail time series visualization. Below the histogram, the mean (dot) and two times the standard deviation range (line width) are represented for both the entire distribution (bottom) and the visible data values (top).

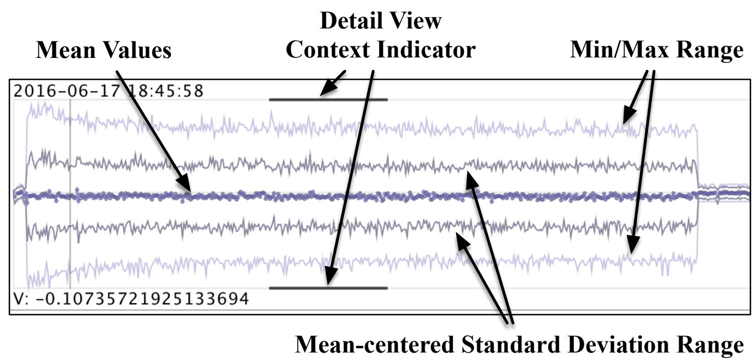

Fig. 5. The overview time series visualization creates an abridged visualization of the full time series. The visualization shows the mean values, and upper and lower standard deviation band, and the minimum and maximum value range. Two black horizontal bars show the relative position of the visible range in the detail time series visualization.

the updated set of visible values and shows darker bars on the histogram to indicate the percentage of the overall bin count.

In addition to the frequency distribution, summary statistics are calculated for both the entire distribution and the visible set of values. As shown in Fig. 4, these summary statistics are visually represented in a statistical visualization that is inspired by the box plot [33]. Here the dot represents the typical value (mean or median) and the width of the line represents the dispersion range (two times the mean-centered standard deviation range or the interquartile range). The statistics for the set of visible values in the detail time series visualization are also shown using a dark gray color (see Fig. 4).

The overview time series visualization is rendered below the overview statistical visualization (see Fig. 3). In this view, the entire time series is condensed to fit the width of the variable pane overview region. To avoid overcrowding, a statistical aggregation of the full time series is shown. The system computes the maximum and minimum value range, dispersion range, and typical values for equally-size time intervals (or bins) that cover the whole time series. Using these summary values, the system creates an abridged time series visualization (see Fig. 5). By default, the mean values are shown as points that are connected with line segments. The dispersion range is shown by doubling the standard deviation, and centering it about the mean value. The upper and lower bounds for both the dispersion and minimum/maximum value range are rendered as polylines. Alternatively, the visualization can be modified to render the bin medians and interquartile ranges. The overview time series visualization provides the context of the entire time series in a compact form. Although some details of the raw values are hidden, the range representations visually encode the essence of the variability to compensate for the loss of information.

5.2.2 Detail Time Series Visualization

In addition to the overviews, each variable pane includes a scrollable, detail time series visualization (see Fig. 3). Two user-defined settings control the layout of the the time scale along the *x*-axis: the chronological time unit (e.g., seconds, minutes, hours) and the pixel width of each time unit. Based on these settings, the system maps the time stamp for each data item to the time axis and renders the data points for the entire time series. If the width of the time scale is greater than the width of the panel, scroll bars appear to allow the user to virtually navigate forward and backward in time. To increase performance with long time series data, the system only renders the visible range of the time series, also known as the clip range.

As shown in Fig. 3, the detail visualization offers several interactive mechanisms for probing the data. Above the time series visualization, a time information bar shows the start and end time instants for the visible range, as well as the time instant that corresponds to the mouse hover position. Below the visualization, a value information bar shows both the value and moving range when the mouse hovers over a data item. The user can click on the value information bar to pin a value marker, which causes it to persist in the display for comparative purposes (see Fig. 3). Clicking on a pinned value maker again removes it. The user may also drag a time range in the detail time series visualization. The selected time range is indicated by a rectangle with a yellow halo. It is also linked to the selection details pane, which is described in Sec. 5.3. The range can be quickly translated and removed using mouse-based gestures.

The detail time series visualization offers four different modes for displaying the data items (see Fig. 6). The point mode simply represents each data item as a circle. The circle interior is not filled to alleviate overplotting issues with dense collections of points. In the line mode, the data items are shown as points and connected with straight line segments based on the temporal sequence. With the stepped line plot, the value is assumed to remain constant until the time instant of the next value, which results in a horizontal line segment between data points. When the next value is reached, a vertical line segment is drawn to connect the new value to the horizontal line segment of the previous value. This rendering option produces a stepped line configuration, which is a more accurate representation for most sensor readings in the 3D printer logs.

The final representation option for the detail time series visualization is the spectrum mode (see Fig. 6). This mode is similar to a bar chart, except the bar is centered on the zero line, which passes through the middle of the *y*-axis. To indicate the sign, the bars are shaded differently for positive and negative values using user-defined colors. Inspired by audio spectrum plots, the spectrum mode emphasizes the magnitude of the values and tends to display changes between successive values in a more visually salient manner.

In addition to the data values, the detail time series visualization can optionally represent the moving ranges for the data values. Inspired by the process behavior chart introduced by Wheeler [37], the moving

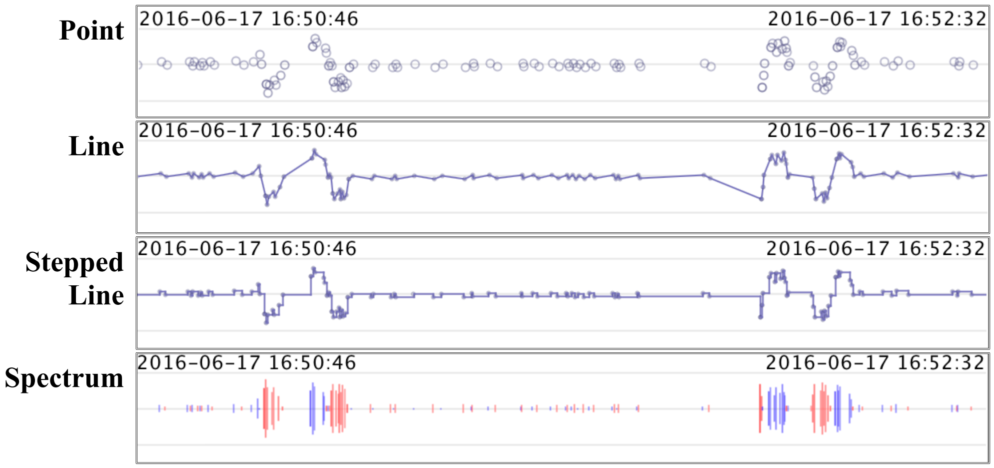

Fig. 6. The detail time series visualization can be configured to represent the data in one of four modes: point, line, stepped line, or spectrum. In addition to showing the data values, each mode can show the moving range values.

ranges are the differences between successive values and are used to measure routine variation. The user can choose to replace the data value with the moving ranges in the visualization or the moving ranges can be encoded in the opacity of the glyph (point, line, bar) representing the data value. When mapped to opacity, the smallest and largest range changes are mapped to the most and least transparent shadings, respectively. This approach emphasizes the variation in values while preserving the individual values. The visualization can also be configured to not show the moving ranges.

As shown in Fig. 3, contextual information is preserved in the overview visualization by linking the scroll actions in the detail visualization to highlight the visible range with two black bars that surround the visualization. Coupled with the overview visualizations, the detailed time series visualization allows fine-grained, multi-scale data exploration while maintaining an awareness of temporal context in the whole data set.

5.3 Selection Details Panel

Located to the right of the variable visualization panel (see Fig. 2), the selection details panel shows statistical information for selections made in the detail visualization panel. When a time range selection is created, a selection details pane is added. The panes are stacked vertically with the most recent selection appearing at the bottom. Each selection pane includes a statistical visualization identical to the one used in the overview region of the variable visualization panel (see Sec. 5.2.1). Reconfiguration of the pane layout and exporting of data associated with a selection can be accomplished via the button panel at right. The user can also adjust the x and y axis limits to normalize comparisons of different selections.

In Fig. 2, three selections are set within the *BeamCurrent* variable visualization pane. From left to right, the selections correspond to the panes shown in the selection details panel from top to bottom. As the time range selections are translated in the detail time series visualization, the corresponding selection pane visualizations are updated automatically. The selections provide the user with a lens for detailed comparisons.

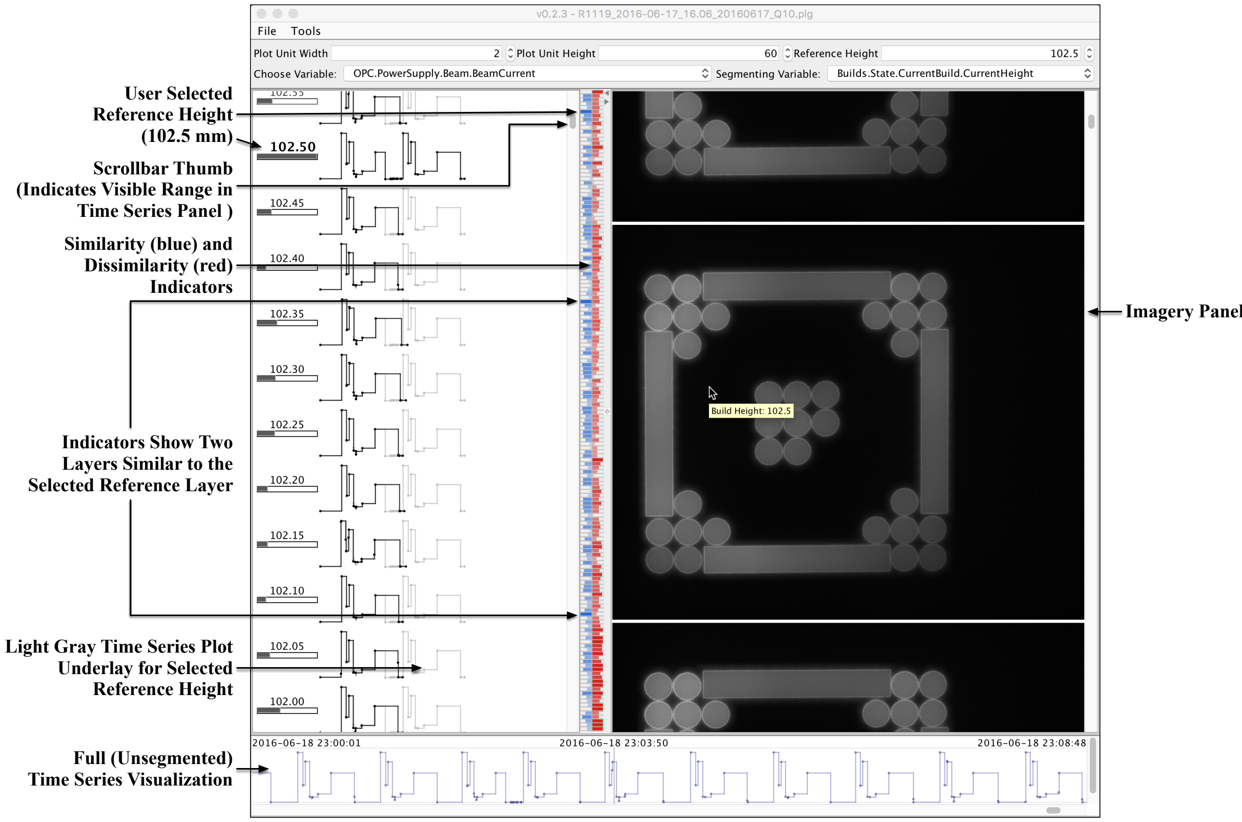

Fig. 7. The segmented time series view partitions the full time series for a variable of interest into multiple time series plots using a user-defined segmenting variable. Here the segmenting variable is the layer build height. The view consists of three main visualization panes: the segmented time series pane (left), the similarity/dissimilarity pane (middle), and the image view pane (right).

5.4 Segmented Time Series Visualization

The segmented time series visualization (see Fig. 7) is a separate window that is accessible from the main Falcon user interface. It allows the user to partition the full time series for a single variable into multiple time series plots for detailed comparative analysis. In addition to the main time series variable, the user specifies a segmenting variable to control when the time series breaks occur. For 3D printer logs, we often segment using the build height variable, which changes as the system begins printing a new layer. Although generalizable to other scenarios, most of the segmented visualization features are designed specifically for the 3D printing scenario. Therefore, the following description will focus on build height as the segmenting variable. As shown in Fig. 7, the segmented view consists of three main visualization panes: the segmented time series pane (left), the similarity/dissimilarity indicator pane (middle), and the image view pane (right).

The time series visualization panel displays the segmented time series plots by stacking them vertically from the top to the bottom of the build. The numerical labels shown to the left of the time series indicate the segment build heights. To select a reference segment, the user clicks on its label. Then, the time series for the reference segment is drawn beneath the other segments' time series plots using a light gray color to support comparative visual analysis. The reference segment is also compared numerically to the other

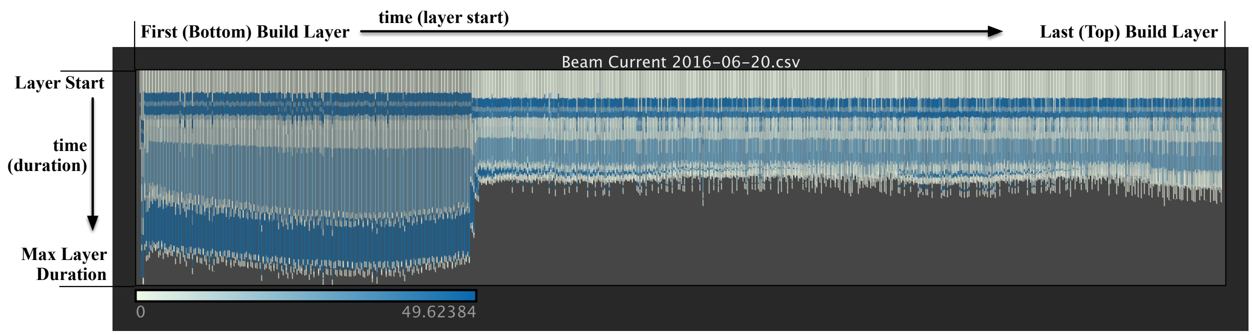

Fig. 8. The waterfall visualization shows a segmented overview of an entire time series using detailed micrographics. Each vertical line represents a build layer from a 3D print. The layer start time determines the *x* location of the line. The line length shows the layer time duration. The value is encoded in the color of the line segments using the color scale shown below. In this figure, it is clear that that the layers in the first third of the build are significantly different.

segments using an implementation of the dynamic time warping algorithm [25]. This algorithm finds the optimal alignment between two times series and yields a distance metric for estimating similarity. The normalized similarity metric is shown graphically as the bar below each segment label. Here the more filled the bar, the more similar the segment is to the reference segment. For example, in Fig. 13, the segment at height 20.00 mm is most similar to the reference height of 102.5 mm.

In the similarity/dissimilarity indicator panel, the dynamic time warping distance values are binned and used to visually represent similarity (blue) and dissimilarity (red). Corresponding to the layout of the time series segments, the indicators are arranged vertically to represent the interesting regions of the entire time series. As the number of build height segments typically exceeds the amount of available space, the segments are group together using a binning algorithm. For each bin, we calculate the median distance for all segments from the reference segment. The smallest distance values are represented by the blue bars and the largest by the red bars. Currently, both the color saturation and bar length encode the difference from the bin's median distance. The distance indicator pane provides an overview that highlights similarity/dissimilarity for all segments, even those whose time series plots are not visible in the left pane.

The rightmost imagery panel shows the near IR images for each build layer in a scrollable view similar to the time series visualization panel. The user can change the image scale by adjusting the panel bounds. To maintain a consistent view, scrolling in all of the panels can be synchronized. The image view provides a different perspective for visually examining build quality and identifying defects such as porosity, swelling, and temperature variation in a layer.

5.5 Waterfall Visualization

The waterfall visualization (see Fig. 8) provides an overview of a particular variable while also showing the detailed value changes for segments of the full time series. This technique combines the benefits of the overview, detail, and comparative analysis views. For the 3D printer scenario, we use it to gain a unique overview of the build process.

Using the same segmentation process described in Sec. 5.4, the waterfall visualization is constructed by segmenting the full time series into smaller time series. For the 3D printer logs, the segmenting variable is the build height variable. The segmented time series are visually represented as a vertical line. Successive layers are plotted with the layer start instants aligned along the top of the visualization. As shown in Fig. 8, the start time of the layer is used to determine the *x* position of the vertical line in a descending layout from left to right. The vertical line is shaded using the values of the variable. Like the stepped time series line drawing mode described in Sec. 5.2.2, a step encoding is used where the variable value remains constant until a new value is encountered. The layer segments are shaded with a semi-transparent color, which reveals portions of the build with shorter layer build times and more layers.

The waterfall visualization provides an important component to understanding segments of time series data. It provides both an overview of the entire build and details about individual stages within each build layer using an approach reminiscent of the micro/macro readings described by Tufte [32]. Therefore, the broad trends and details are displayed together and scientists gain a inferential perspective.

6. Case Study: 3D Printer Log Analysis

In this section, we describe a practical case study to demonstrate the effectiveness of Falcon at revealing patterns at multiple scales in complex time series data. Using the Arcam Q10 system, we recently printed a particular test artifact that acts as a control for the system. The geometrical configuration of this test, which is shown in Fig. 1, is used to ensure that the system is functioning properly by printing a structure that is created entirely from the build platform without support structures. This configuration includes four distinct geometrical layouts as well as five specific features. The features include 4 - 104 x 104 x 15 mm blocks, 58 - 105 x 15 mm cylinders, 5 - 15 mm cubes, 1 - 15 x 30 x 5 mm block, and 1 - 30 x 15 mm cylinder. The data generated from a build of this test configuration is the subject of the analysis described in the remainder of this section. We describe the analysis of this build as a story told from the perspective of the additive manufacturing researcher.

6.1 How The Researchers Use Falcon

First, we describe the typical strategy that additive manufacturing researchers use to analyze their data with Falcon. This general workflow has evolved based on several months of experience using Falcon to analyze a variety of builds. The workflow closely follows the stages of Shneiderman's visual information seeking mantra: "Overview first, zoom and filter, then details-on-demand" [27]. Although the design of Falcon may include some indirect guidance to proceed from an overview to details during analysis, the researchers formed this workflow based on their own independent explorations. The fact that they naturally gravitated to this strategy corroborates the notion that it is a favorable strategy for designing new data visualization systems.

The researcher begins by visualizing an overview of the entire build using a limited set of variables. By adjusting the display parameters, they create a view that shows the full time series with aggregated statistical summary information. This view is studied, using the zoom-and-filter capabilities, to find interesting trends and patterns based on the researcher's background knowledge and intuition. During this stage, the researcher notes the time and build layer of interesting patterns, particularly those that may indicate abnormal conditions. Then, they conduct more detailed investigations of the data by creating fine-grained views, comparing the current data to other log data files, and viewing the near IR images for

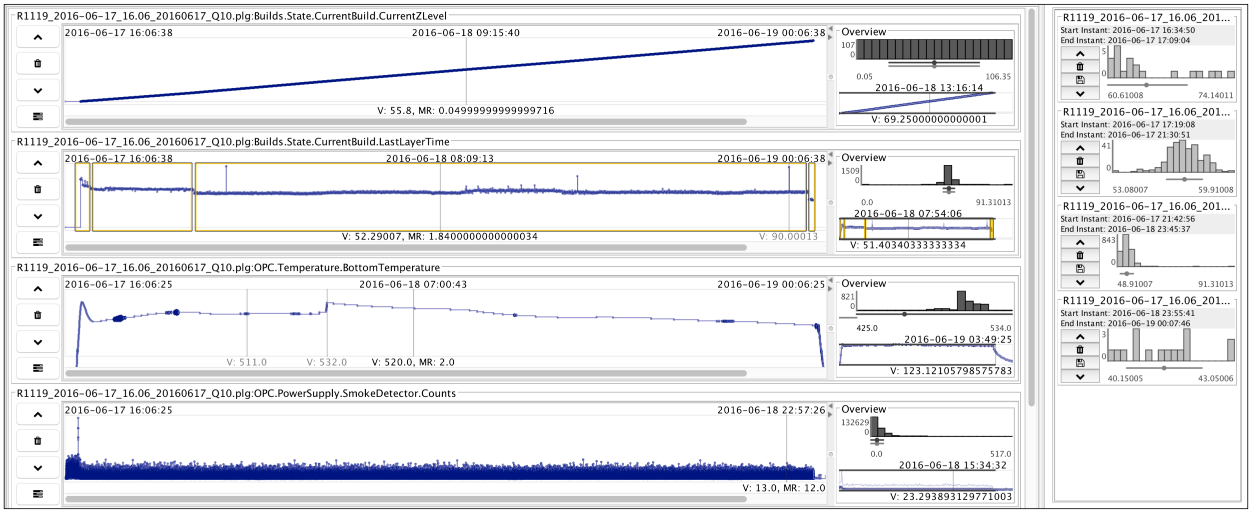

Fig. 9. We begin analysis of the 3D printer log by visually inspecting overview visualizations for a set of key variables. Here we show four of key variables, but a typical session would involve dozens. The three outliers in the *LastLayerTime* visualization and a spike in the *BottomTemperature* visualization require further investigation as they may indicate problems in the build.

each build layer. They iterate using this general process, moving back and forth between different levels-of-detail, until the results are satisfactory.

6.2 Overview First

We begin the analysis session by loading the printer data (a log file and near IR images) into Falcon and creating an overview of the entire build for the set of key variables. Based on our experience acquired through the analysis of other builds and our background knowledge of additive manufacturing, we have determined that these variables are prime indicators of the overall quality of the build. Although this case study focuses on a few variables from this limited set, it is important to note that many other variables are studied during a normal analysis session.

As shown in Fig. 9, we construct the visual overview by adjusting the parameters to show the entire time series in the detail time series visualization for each variable. In this case, the overview is created by setting the *Plot Chrono Unit* to hours and the *Plot Unit Width* to 30 pixels, which yields a 2-minute resolution. The figure shows four from the full set of key variables, but a typical analysis session may include dozens of variables from multiple files.

In Fig. 9, the top variable visualization panel shows the *CurrentZLevel* variable, which indicates the distance/height in millimeters from the bottom layer. We recognize the steady increase in the values of this variable over the duration of the build as an artifact of the standard manner by which the system prints from the bottom to the top of the model. The mouse hover query on the *CurrentZLevel* detail time series visualization shows that the height is 55.8 mm at 9:15:40 and the moving range is approximately 0.05 mm, which is the expected layer thickness. From this coarse view, the build height plot is consistent with a normal build.

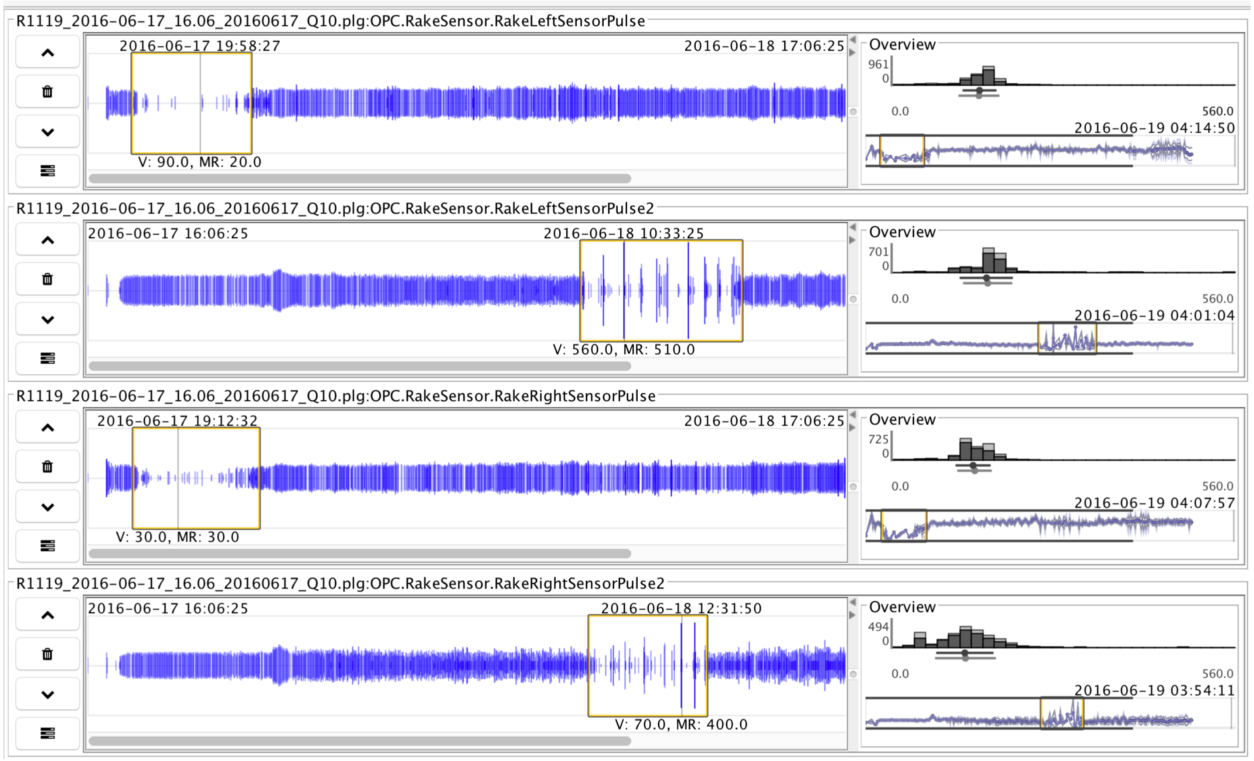

Fig. 10. We use the spectrum time series plot mode to analyze four rake sensors over the entire build. In a normal build, these plots should show a minimal variation without large gaps. Here the gaps and spikes that appear may indicate problems with the power bed consistency or rake blade. The spectrum plot mode excels at reveal such conditions.

Our eye is drawn next to the third variable panel, which shows the *BottomTemperature* variable. This variable indicates the temperature at the base plate, which should steadily decrease after the build initialization phase as the print layer height increases. In the detail time series visualization, we note a sharp increase in temperature near the middle of the build process (see the second pinned value of 532°C in Fig. 9). This abnormal pattern warrants further investigation as it may indicate a change in processing parameters due to the geometry of the object.

As shown in Fig. 10, we change to the spectrum time series plot mode to check the printer rake sensor readings. These sensors measure the consistency of the powder bed by raking powder over the sensor flaps and measuring the time that they are open. Since a consistent power bed is vital to an optimal melt process, the gaps and value spikes that appear in the highlighted ranges of this visualization are troubling. The spectrum visualization excels at revealing patterns of this nature. In addition to prompting us to inspect the consistency of the printed object, we are compelled to physically inspect the powder (e.g., distribution and chemical composition) and rake blade (e.g., bends, cracks, or holes) mechanisms within the printer.

Turning our attention back to Fig. 9, we study the second variable pane, which shows the *LastLayerTime* variable. As it captures the print time of the preceding layer, this variable is directly related to each layer's melt area as more time is required to melt a larger area, and vice versa. From this view, we identify and highlight four distinct stages of the overall print process for the test configuration using the time range

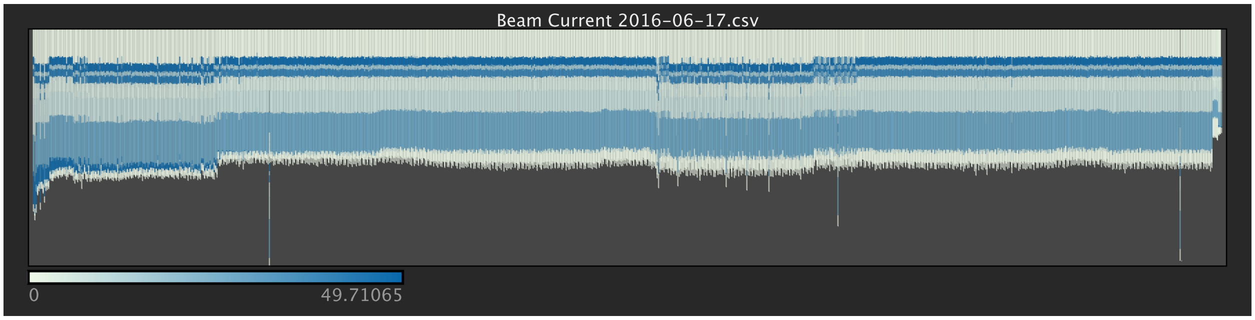

Fig. 11. We construct a waterfall visualization of the *BeamCurrent* variable for a more focused overview. In addition to the geometry ranges and outliers that we identified in the overview line plots, we see subtle variations in the layer process and build stages through the color encoding.

selection tool (see the yellow rectangular highlights in Fig. 9). These stages correspond to four regions of the melt area for the build starting with the largest per layer melt area at the bottom and progressing to the smallest at the top. In the bottom region (see the far left selection that corresponds to the top statistical selection view at right), the build layers require the most melt time per layer (between 60 and 74 seconds). In the next selected region (see the second selection from the left and second statistical selection view from the top at right), the build layers require less melt area and exhibit a more normally distributed range of melt times (between 53 and 60 seconds). The smaller dispersion of values and apparent lack of significant outliers in this region (see the statistical summary histogram) increases our confidence in the quality. Jumping ahead in the build time series, we note that the fourth region (see the far right selection and bottom statistical selection view at right) is also characterized by relatively small deviations in layer build times and requires the least amount of time to complete (approximately 13 minutes).

We now turn our attention to the third selection range in the *LastLayerTime* detail time series visualization (see the third selection from the left and the third statistical selection view from the top in Fig. 9). We observe three outliers occurring near the beginning, middle, and end of this range, the largest of these being approximately 91.3 seconds (see max value shown in the statistical view). These outliers skew the histogram plot, which reduces the perceivable structure of values within the range of normal layer times. As shown in Fig. 9, we set a pinned value marker on the last outlier showing the value is about 90 seconds. These aberrations suggest that something occurred in the preceding layer to cause a significantly longer time for its completion, which may indicate problems in the build. Therefore, we must investigate the corresponding layers in more detail.

6.3 A More Focused Overview

We create a more focused overview using a waterfall visualization of the *BeamCurrent* variable (see Fig. 11). As described in Sec. 5.5, the waterfall visualization is constructed by segmenting the variable into smaller time series, which correspond to the different build layers. Then, each segment time series is displayed as a vertical line where changes in value are encoded using a color scale and step encoding function. This view provides an overview of the *BeamCurrent* time series by showing the detailed information in miniature, while preserving the overall context.

Again we see the four distinct regions that characterize the test pattern. Moreover, we immediately

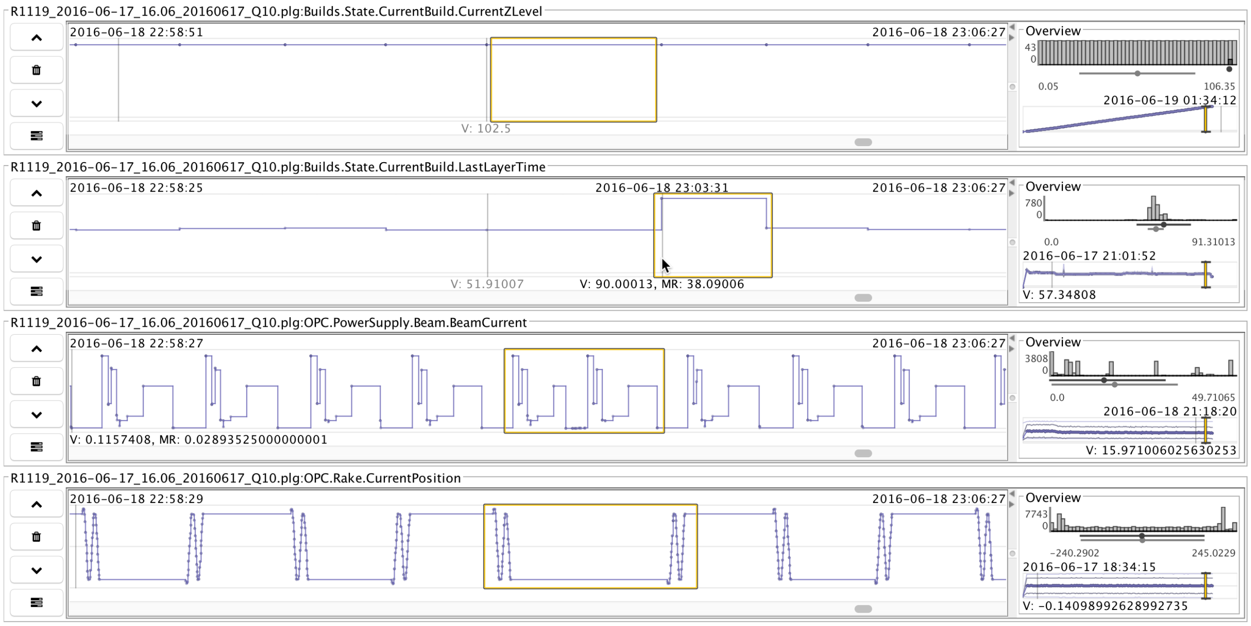

Fig. 12. We zoom into the detail time series visualization by changing the time scale settings to investigate one of the outliers detected in the overview visualization. We see a repeat pattern in the *BeamCurrent* detail visualization that is caused by a system arc trip. The arc trip may indicate problems in the microstructure of the printed objects.

recognize the three outlier layers, as the vertical lines are significantly longer than the others. In addition, we see more subtle variances that we overlooked in the initial overview. We detect slight variances in the initial preheat stages of the layers (see the slight shift in the length of the white segments at the top of each vertical line). We also see smaller variations in layer duration and the beam current values during the middle of the build. Specifically, seven build layers show some variation within the normal range. Nevertheless, during a typical analysis session these variations would warrant deeper investigations; but, for the sake of brevity, we do not discuss them in the current work.

6.4 Zoom and Filter

At this point, we have visually identified potential issues at specific times/heights in the build using overview visualizations. Next, we increase the detail of the visualizations to conduct more thorough analysis of the abnormal build layers. In the detail time series visualization, we increase the level-of-detail by setting the *Plot Chrono Unit* to seconds with a *Plot Unit Width* of two pixels. Fig. 12 shows the resulting view after scrolling to the time range associated with the third outlier, which is indicated in Fig. 9 by the rightmost pinned value marker of approximately 90 seconds.

In Fig. 12, the black context bars, which halo the overview time series plot for each variable, show the relative position of the visible time range in the detail time series visualization. In the *LastLayerTime* detail time series visualization, we highlight the outlier using a time range selection. We use the mouse hover query to confirm that the preceding layer print time value is approximately 90 seconds, which is approximately 38 seconds more than the previous reading of approximately 51.9 seconds (see the pinned marker value to the left of the mouse selection range in the detail time series visualization).

Since *LastLayerTime* values describe the previous layer, we glance back one layer on the other three variable plots shown in Fig. 12. In the detail time series visualization for the *CurrentZLevel* variable, we notice the longer time gap between *z*-level changes (see the highlighted range in the top variable plot). We pin a value marker on this plot to show that the outlier layer occurs at a height of approximately 102.5 mm from the bottom layer. Likewise, we see that the outlier layer is also visible in the wider gap between the *Rake.CurrentPosition* variable readings (see the highlighted range in the bottom variable plot). This variable captures the position of the rake (a mechanism that moves metal powder across the platform before the melting process) relative to the center of the build. The normal rake pattern is visible in the detail time series visualization—the rake traverses three times from one side to the other depositing and smoothing the powder over the print area as it moves.

The most telling discovery is found in the *BeamCurrent* detail time series visualization. In Fig. 12, we highlight the time range for the outlier layer using the time range selection. The normal *BeamCurrent* pattern can be observed in the other layers as it modulates between different predictable levels during each of the five stages the system goes through as it prints a layer. However, in the outlier layer, we see that the pattern is repeated, which narrows down the the cause of the outlier. We look at the last peak of the first five stage pattern where the process seems to stop and restart the print pattern. To see a more focused view, we access a more detailed set of visualizations.

6.5 More Detail, On Demand

To study the abnormal layers in more detail, we open the segmented visualization panel as shown in Fig. 13. We segment the full *BeamCurrent* variable time series into smaller time series visualizations using the *CurrentHeight* variable. The resulting view shows the individual time series visualizations for each build layer. We set the outlier layer at 102.5 mm as the reference segment by directly clicking on its time series. This action causes the panel to render the time series for the reference segment as an underlay, using a light gray color, on all the other time series plots. The reference plot highlights the repeated print pattern. At right of the time series panel scroll bar, the blue similarity indicators reveal the locations of the other two outlier build layers.

In Fig. 13, we scroll to the bottom of the build near 20 mm, where one of the other outliers appears. The indicator shown beneath the height label for each build layer visually encodes the quantitative measure of similarity between the two time series as well as the differences with the other nearby time series. The 20 mm layer indicator suggests that it is similar to the reference layer. In the time series plot, we see the same repeated pattern although the reference signal (at 102.5 mm) appears to be slightly shifted to the left. Despite the slight shift, this evidence suggests that the outliers at 102.5 mm and 20 mm are caused by the same condition.

Finally, the right image panel shows a scrollable view of the near IR images for each layer. This image panel is synchronized with the layer time series panel so that we see the image for the layer at 20 mm. This view aids in the discovery of microstructure problems such as porosity, hot spots, or swelling that may occur during the stages of each layer build. Together with microscope images (not shown), the image views help explain the issues that are seen in the other visualizations and ascertain the quality of the material. In this instance, we see no microstructure problems in the image at layer 20 mm.

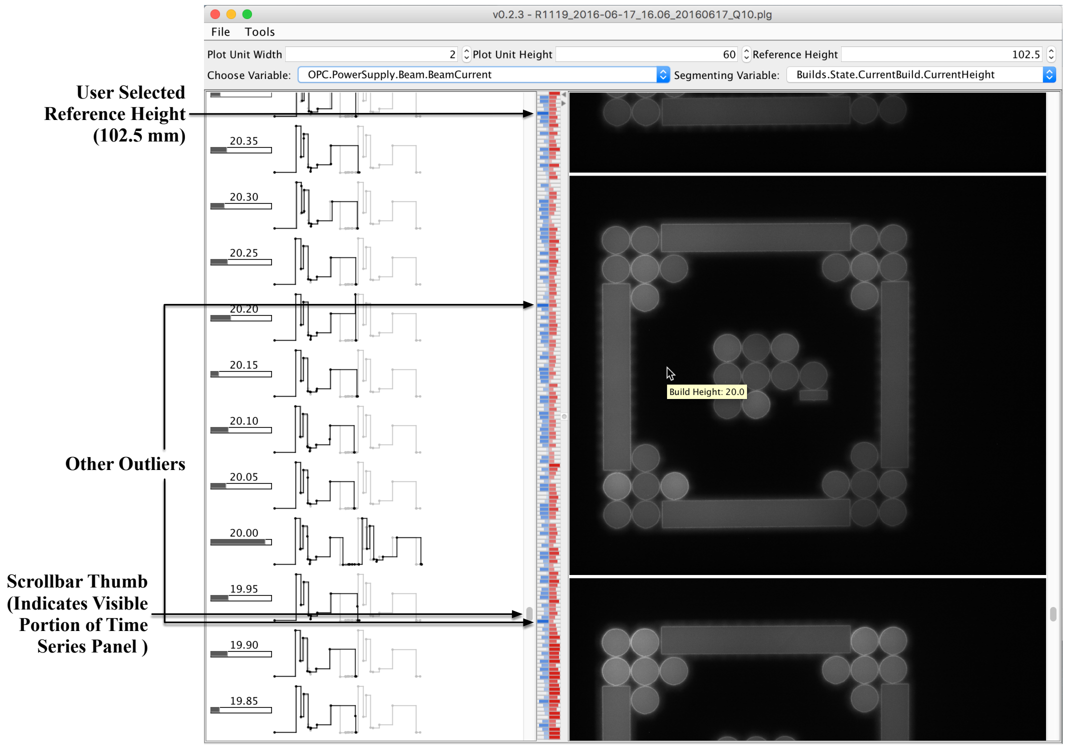

Fig. 13. The segmented view panel provides the highest level of detail in Falcon. Here changes in build height are used to segment the full time series into separate time series for each layer. Using dynamic time warping, we shown similarity/dissimilarity metrics for a user-defined reference layer. At right, we show the near IR imagery for each build layer in a scrollable panel.

6.6 Understanding the Cause of the Outliers

From experience, we know that such abnormal layer times are usually caused by either an arc trip or a smoke detection event. Both conditions cause us to question the build quality. When the beam is shut down during the melting stage (as is the case in this instance), the microstructure in that layer can be altered. Consistent microstructure of the material is essential for creating parts that are usable for production.

An arc trip occurs when the beam forces too much energy density through the filament. The system responds to the arc trip by immediately shutting off the beam and completely restarting the layer build process. In the time series visualization for the *BeamCurrent*, an arc trip appears as a repeated signal of the electron beam current as it progresses through the five layer stages.

The detection of radiation from ionized powder particles in the vicinity of the electron gun is characterized as a smoke event. This condition causes the powder to scatter forming a dust cloud in the build chamber, which can harm the beam filament. As with the arc trip condition, a smoke detection event causes the beam to shut off and the layer print is restarted, which also manifests in the variable visualization as a repeated signal.

By separately analyzing the smoke sensor readings that are also stored in the log file, we determined that the outliers were caused by an arc trip. The bottom variable visualization pane in Fig. 9 shows the

SmokeDetector.Counts variable. Except for a spike in smoke detector counts during the initialization phase, which is expected, the remainder of the detail time series visualization shows normal variations.

The ability to discover and thoroughly investigate such complex patterns helps us formulate algorithms to automatically detect these and other conditions that impact the quality of the print—a critical capability for certifying the quality of 3D printed objects for applications where failures are not tolerable, such as printing aircraft parts. Additional study of this data is also helping us fine tune the print parameters to avoid such conditions in future builds. Moreover, we have engaged in a supplemental investigation of this particular print to understand how changes in the melt pool affect the microstructure. These investigations involve image processing algorithms to detect porosity using the near IR imagery, viewing cross sections of the material through optical microscopes, and using a scanning electron microscope to view extremely fine resolution structure.

7. Discussion

By bringing to bear the latest advances in interactive data visualization, we are helping additive manufacturing researchers develop a deeper understanding of their data, improve the quality of 3D-printed objects, and make fundamental scientific contributions to this expanding domain. In this section, we reflect on ideas to improve Falcon and our experience as a interdisciplinary design team.

7.1 Improving Falcon

Falcon continues to evolve based on practical use in scientific domains, such as additive manufacturing. Through our informal evaluations with expert users and their feedback, we have identified several modifications that may increase its usability and efficiency. First, as noted in the case study, the researchers usually begin with a known set of key variables. Currently, they must manually recreate this view, adding variables to the visualization and fine-tuning the display parameters. We are developing a new capability to allow the user to save the entire view configuration as a template that can be applied to other data sets. Although this capability is not novel from a visualization standpoint, it is tedious to implement and we believe it will significantly reduce the start up time, increase the production of new knowledge, and assist novice users.

Interactive similarity/dissimilarity calculations via dynamic time warping have demonstrated a promising capability that we are expanding into an automated process that guides the user to the most promising time ranges. We are also incorporating new image-based processing algorithms to automatically detect and mark porosity, swelling, and hot spots based on the near IR imagery. The integration of these and other machine learning algorithms will help reduce the search space, recommend hidden features in the data, and increase the overall efficiency of the analysis process.

We are integrating new visualization techniques into Falcon. Additional multivariate visualizations, such as parallel coordinates [30], are under development. We are studying different visualization strategies for representing Boolean and categorical information in time series visualizations. Soon, we plan to investigate new focus + context techniques for time series visualizations to prevent perceptual issues related to analyzing multiple separate views [24]. Another improvement lies in the incorporation of provenance and annotation capabilities in Falcon. Currently, the researchers keep notes separately and refer back to these notes during analysis sessions. Including the ability to capture notes within a session file and tracking the user's history of interaction with the data will further increase the efficiency and foster more collaborative analysis.

7.2 Reflections on the Interdisciplinary Team

The success of Falcon is largely attributed to the assembly of an interdisciplinary team of highly motivated individuals each bringing a specific area of expertise to the project. The additive manufacturing experts at the ORNL MDF constantly push the limits of manufacturing technology to build 3D printed vehicles, complex aircraft parts, and even sustainable houses. These researchers are the primary users of our tool and we meet with them weekly to discuss new developments, evaluate their use of the tools, and plan future milestones.

This collaboration works well because of several factors. First of all, we are located in close geographic proximity. Communication and coordinating working meetings with the team is easy. Next, for design meetings we use a specially equipped visualization laboratory with a large display wall and several other technologies that foster collaborative discussions. Personalized training also increases the researchers' proficiency with the visual analytics process. When new features are added to Falcon, we devote some time to helping them learn how to utilize them. Although this detracts from development time, it pays dividends through increased scientific output. Training time is also decreased by having the domain experts directly involved in the implementation of new features.

Finally, we have sacrificed some generality in Falcon to implement features that are specific to the additive manufacturing scenario, such as segmentation according to build heights and porosity detection. Although these decisions make Falcon less compatible with all possible scenarios, it has enabled discoveries that would be far more difficult, if not impossible, without them. That said, we have striven to modularize such domain specific capabilities to permit future utilization in fields that face similar challenges, such as cyber security and climate science.

8. Conclusion

In this report, we present Falcon, a new system that follows a visual analytics approach to improve knowledge discovery in long and complex time series data with practical applications to the field of additive manufacturing. Falcon leverages a human-centered design grounded in the visual information seeking strategy [27]. Falcon provides linked visualizations from both temporal and statistical orientations with automated analytics to highlight interesting features. In addition, Falcon offers intuitive mechanisms to access multiple levels-of-detail as necessary.

From our informal evaluations of the applied use of Falcon in additive manufacturing, we have learned that non-visualization experts can be vital members of interdisciplinary design teams as they help design new capabilities that respond to their actual needs, and they quickly employ new visual analytics techniques in creative ways to solve problems. The parallels between the analytical goals in additive manufacturing and other domains suggest that these capabilities are broadly applicable to many domains as they help users develop and refine a more complete mental model of complicated and large-scale time series data.

9. REFERENCES

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.

[2] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A review of temporal data visualizations based on space-time cube operations. In *Proceedings of EuroVis*, 2014.

[3] W. I. B. Beaveridge. *The Art of Scientific Investigation*. The Blackburn Press, Caldwell, NJ, 1957.

[4] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[5] E. Bertini, P. Hertzog, and D. Lalanne. SpiralView: Towards security policies assessment through visual correlation of network resources with evolution of alarms. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 139–146, 2007.

[6] P. Buono, C. Plaisant, A. Simeone, A. Aris, G. Shmueli, and W. Jank. Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting. In *Proceedings of the International Conference Information Visualization*, pages 191–196, 2007.

[7] Paolo Buono, Aleks Aris, Catherine Plaisant, Amir Khella, and Ben Shneiderman. Interactive pattern search in time series. In *Proceedings of SPIE Visualization and Data Analysis Conference*, volume 5669, pages 175–186, 2005.

[8] Weiwei Cui, Shixia Liu, Li Tan, Conglei Shi, Yangqiu Song, Zekai Gao, Huamin Qu, and Xin Tong. TextFlow: Towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, 2011.

[9] Marian Dörk, Daniel Gruen, Carey Williamson, and Sheelagh Carpendale. A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1129–1138, 2010.

[10] Wenwen Dou, Xiaoyu Wang, Drew Skau, William Ribarsky, and Michelle X. Zhou. LeadLine: Interactive visual analysis of text data through event identification and exploration. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 93–102, 2012.

[11] M. C. Hao, Umeshwar Dayal, D. A. Keim, and T. Schreck. Importance-driven visualization layouts for large time series data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 203–210, Oct 2005.

[12] Lane Harrison, Xianlin Hu, Xiaowei Ying, Aidong Lu, Weichao Wang, and Xintao Wu. Interactive detection of network anomalies vis coordinated multiple views. In *Proceedings of the IEEE Symposium on Visualization for Cyber Security*, pages 91–101, 2010.

[13] Blake Haugen, Stephen Richmond, Jakub Kurzak, Chad A. Steed, and Jack Dongarra. Visualizing execution traces with task dependencies. In *In Proceedings of the 2nd Workshop on Visual Performance Analysis*, pages 1–8. ACM, 2015.

[14] S. Havre, B. Hetzler, and L. Nowell. ThemeRiver: Visualizing theme changes over time. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 115–123, 2000.

[15] Jeffrey Heer, Nicholas Kong, and Maneesh 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*, pages 1303–1312, 2009.

[16] W. Javed and N. Elmqvist. Stack zooming for multi-focus interaction in time-series data visualization. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 33–40, March 2010.

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

[18] R. Kincaid. SignalLens: Focus+context applied to electronic time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):900–907, 2010.

[19] J. Li, K. Zhang, and Z. P. Meng. Vismate: Interactive visual analysis of station-based observation data on climate changes. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pages 133–142, 2014.

[20] Jessica Lin, Eamonn Keogh, Stefano Lonardi, Jeffrey P. Lankford, and Donna M. Nystrom. Visually mining and monitoring massive time series. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 460–469, 2004.

[21] Dongning Luo, Jing Yang, Milos Krstajic, William Ribarsky, and Daniel Keim. EventRiver: Visually exploring text collections with temporal references. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):93–105, 2012.

[22] Chris North. Multiple views and tight coupling in visualization: A language, taxonomy, and system. In *Proceedings CSREA CISST Workshop of Fundamental Issues in Visualization*, pages 626–632, 2001.

[23] C. Plaisant, D. Carr, and B. Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Software*, 12(2):21–32, 1995.

[24] Ronald A. Rensink. Change detection. *Annual Review of Psychology*, 53:245–577, 2002.

[25] Stan Salvador and Philip Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. In *Proceedings of the ACM KDD Workshop on Mining Temporal and Sequential Data*, pages 70–80, 2004.

[26] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[27] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[28] C. A. Steed, M. Drouhard, J. Beaver, J. Pyle, and P. L. Bogen. Matisse: A visual analytics system for exploring emotion trends in social media text streams. In *Proceedings of the IEEE International Conference on Big Data*, pages 807–814, 2015.

[29] C. A. Steed, K. J. Evans, J. F. Harney, B. C. Jewell, G. Shipman, B. E. Smith, P. E. Thornton, and D. N. Williams. Web-based visual analytics for extreme scale climate science. In *IEEE International Conference on Big Data*, pages 383–392, 2014.

[30] Chad A. Steed, Daniel M. Ricciuto, Galen Shipman, Brian Smith, Peter E. Thornton, Dali Wang, Xiaoying Shi, and Dean N. Williams. Big data visual analytics for exploratory earth system simulation analysis. *Computers & Geosciences*, 61:71–82, 2013.

[31] E. R. Tufte. *Beautiful Evidence*. Graphics Press, 2006.

[32] Edward R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, CT, 1990.

[33] John W. Tukey. *Exploratory Data Analysis*. Addison Wesley, Reading, MA, 1977.

[34] J. Walker, R. Borgo, and M. W. Jones. TimeNotes: A study on effective chart visualization and interaction techniques for time-series data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):549–558, 2016.

[35] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 110–119, 2000.

[36] M. Weber, M. Alexa, and W. Müller. Visualizing time-series on spirals. In *Proceedings of the IEEE Symposium on Information Visualization*, 2001.

[37] Donald J. Wheeler. *Making Sense of Data*. SPC Press, Knoxville, TN, 2003.

[38] J. J. Van Wijk and E. R. Van Selow. Cluster and calendar based visualization of time series data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 4–9, 1999.

[39] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.

[40] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011.