Paper:
Walker, J., Borgo, R. & Jones, M. (2016). TimeNotes: A Study on Effective Chart Visualization and Interaction Techniques for Time-Series Data. *IEEE Transactions on Visualization and Computer Graphics, 22*(1), sss-eee.

http://dx.doi.org/10.1109/TVCG.2015.2467751

TimeNotes: A Study on Effective Chart Visualization and Interaction Techniques for Time-Series Data

James Walker, Rita Borgo and Mark W. Jones

Abstract— Collecting sensor data results in large temporal data sets which need to be visualized, analyzed, and presented. One-dimensional time-series charts are used, but these present problems when screen resolution is small in comparison to the data. This can result in severe over-plotting, giving rise for the requirement to provide effective rendering and methods to allow interaction with the detailed data. Common solutions can be categorized as multi-scale representations, frequency based, and lens based interaction techniques.
In this paper, we comparatively evaluate existing methods, such as Stack Zoom [15] and ChronoLenses [38], giving a graphical overview of each and classifying their ability to explore and interact with data. We propose new visualizations and other extensions to the existing approaches. We undertake and report an empirical study and a field study using these techniques.

Index Terms—Time-series Exploration, Focus+Context, Lens, Interaction Techniques.

1 Introduction and Motivation

The last decade has seen an explosion of interest in time-series data. Understanding temporal patterns is key to gaining knowledge and insight. However, collectively, our ability to store data now far exceeds the rate at which we are able to understand it [19].

One challenge is that screen resolution is small in comparison to data storage capacity. When more data items are rendered than the available pixels an over-plotting problem occurs where more than one data item is assigned to each pixel, which leads to a loss of information. Multi-scale representations, frequency, and lens based interaction techniques have been introduced to enhance the exploration of large time-series data.

John Stasko said at EuroVis 2014 [32], “Use data mining when you know the question and visualization when you do not”. Analysis often involves identifying segments of the time-series where phenomena occur and comparing between time segments for interesting patterns which can be used to form, prove or refute a hypothesis. After analysis the findings are communicated to a wider audience. Navigating and communicating through a large data space is an important task which is not fully supported by existing techniques (demonstrated in our task based evaluation).

In this paper, we evaluate current visualizations and extensions to these existing approaches. Based on our evaluation we propose TimeNotes, a visualization technique utilizing built-for-purpose interaction techniques that supports analysis, interaction and presentation. We evaluate the effectiveness of TimeNotes through an empirical study and a field study which highlights the application of our approach applied to time-series data.

Our work consists of the following contributions:

1. TimeNotes, a more than effective approach for chart visualization and interaction with time-series data.
2. A graphical survey, task based evaluation and classification of the related approaches.
3. A user study comparing TimeNotes to the state-of-the-art stack zoom method.
4. Feedback from a deployment of the software with biologists.

- *James Walker, Swansea University, E-mail: james@framework4.co.uk*
- *Rita Borgo, Swansea University, E-mail: r.borgo@swansea.ac.uk*
- *Mark W. Jones, Swansea University, E-mail: m.w.jones@swansea.ac.uk*

- *All authors contributed equally to this work. This work was part funded by Leverhulme grant RPG-2013-190.*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx xxx 2015; date of current version xx xxx 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

The rest of this paper is organized as follows. In section 2, we present the related work. In section 3, we introduce tasks and requirements when operating on time-series data. In section 4, we introduce TimeNotes. In section 5, we outline the empirical study and results obtained. In section 6, we detail the field study and findings, and in section 7, we conclude our findings.

2 Literature

In this section we present the current approaches in the literature for exploring large time-series data. Many effective methods have been introduced to address time-series mining tasks [36, 6, 23, 13, 12], however, in this paper we focus on effective exploration.

2.1 Exploration of Time-Series Data

The line graph is ubiquitous [17, 34, 25]. While the standard time-series graph is effective when dealing with a small data space, it is more challenging to perform common tasks on large data (Figure 1). Interaction techniques have been introduced to enhance the time-series graph for large data. In the current literature, these can be categorized as aggregation of time, lensing techniques and layout distortions.

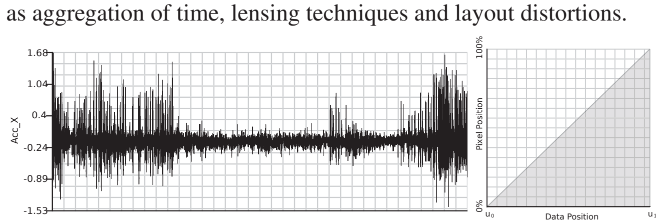

Fig. 1: Left: 1D Line Chart with 34,746 data elements. Right: Linear mapping from temporal domain to x-axis.

We now provide a graphical overview of these methods. Isenberg et al. [14] present a generic framework to provide design guidelines for charts containing dual scales to enhance data perception. A generalized transformation function is introduced which describes any chart showing two domain scales, which involves mapping data space to a display space. We utilize their terminology to describe the current literature and apply their work to describe any time-series chart featuring multiple scales and layouts. We have implemented all of the methods in our software in a consistent way and generate automatically the associated time domain to x-axis transformation functions as corresponding graphs. These are exported as SVG and incorporated into this document as an illustration of the method and of our classification.

We use for our data a 15 minute (approximately) sub-section of remote animal monitoring data obtained from a deployment of a behavioral data collection tag on a Condor consisting of 34,746 data items collected at 40Hz. Each of the visualization methods utilize 400 pixels width, which leads to an over-plotting ratio of 86 data items per pixel in the traditional time-series graph. This is problematic when considering a single Cormorant wing beat occurs at a frequency of approximately 5Hz [27]. The problem worsens when presented with larger data (e.g., the full deployment of 857,407 items).

2.1.1 Data aggregation

Frequency based approaches aggregate data points into segments of time. Rendering large data in a line graph implicitly aggregates time together (via over-plotting) but in a non-meaningful way representing a fraction of the underlying data per pixel. Effective aggregation depicts statistical features of the items in each segment of time through a meaningful visual mapping.

Pixel plot Pixel based displays represent a time-series as an arrangement of pixels encoded with different hues which encode the underlying data. Kincaid et al. [22] apply a pixel based display to multiple time-series graphs. Each time-series is split into uniform segments of time, such that each pixel in the visualization is assigned a time segment. Figure 2 shows the maximum value in each time segment mapped to a yellow to blue color hue. Each bin is one pixel wide, resulting in 400 bins, with each segment containing approximately 86 data items.

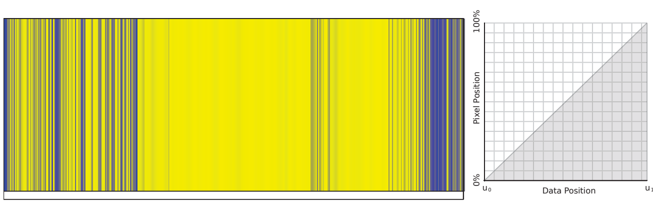

Fig. 2: Pixel Plot [22] and linear temporal to position map.

Two-dimensional pixel plot Hao et al. [10] extend the one-dimensional pixel display to two-dimensional space utilizing the *x* and *y* axes. Each segment of time is represented by a color cell chronologically ordered navigating from bottom to top, and left to right. By utilizing two-dimensions, the visualization efficiently occupies space, which increases the number of time segments which can be displayed resulting in less loss of information when summarizing. Figure 3 shows the maximum value in each time segment, each bin occupies 1 pixel width, and 3 pixels height. The resulting visualization contains 34,746 bins, which leads to 86 data items per column and 1 item per 3 pixels for this data size. In addition to the horizontal data mapping there would be a saw tooth like vertical mapping (not shown).

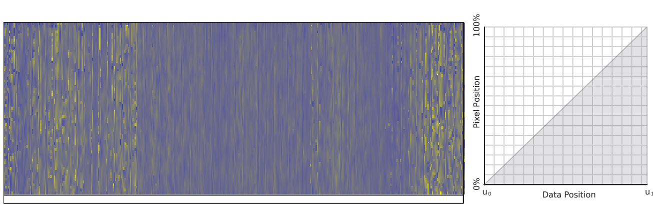

Fig. 3: 2D Pixel Plot [10] and linear temporal to position map.

River plot Buno et al. [7] introduce a river plot (Figure 4) to depict a number of statistical properties of time-series predictions. A bounded blue area connected through time represents the minimum and maximum bounds for each time segment. A black central line depicts the mean value for each segment of time.

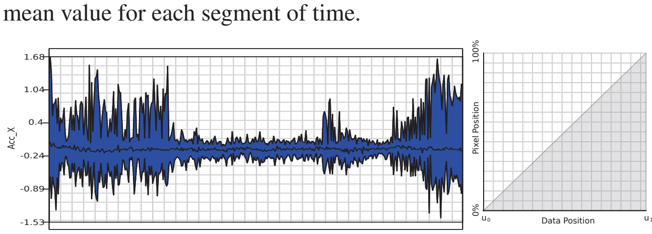

Fig. 4: River Plot [7] and linear temporal to position map.

2.1.2 Lens based

Lensing techniques provide on demand alternative visual representations of the data underlying user-defined regions. Typically, this entails time-axis distortion to enhance segments of interest, while maintaining context with the remaining series.

SignalLens Kincaid et al. [21] present SignalLens for the visual analysis of electronic time series. An in-place magnification is added to the time-series plot which distorts the time-axis to magnify areas of interest. The data either side of the magnified area is compacted to maintain context, while allowing the inspection of low-level details of interest. A number of lens functions are introduced for comparison: linear, cubic, quadratic, hyperbolic, spherical and Guassian. Figure 5 illustrates SignalLens applied with a linear magnification function.

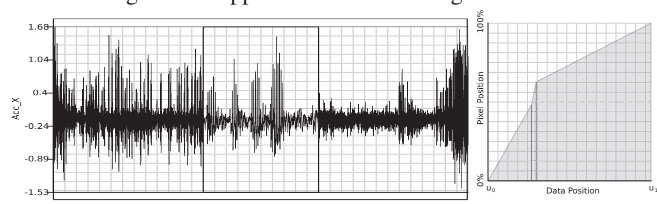

Fig. 5: SignalLens [21] and temporal to position map. The temporal to position map represents this region as a higher linear gradient.

Smooth SignalLens High magnification generally requires a smooth drop-off to avoid occlusion in the context region. Kincaid et al. [21] introduce a smooth visual transition option between the focus and context regions (Figure 6). This entails three zoom levels, the focus area which is of a fixed magnification in the center, a lower magnification drop off area either side, and finally the context displayed in the remaining space.

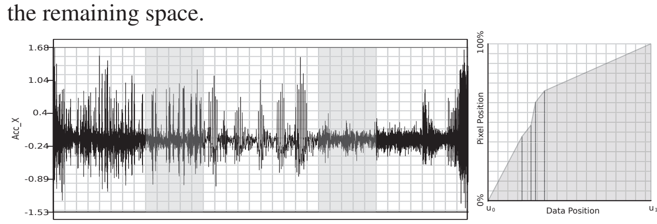

Fig. 6: Smoothed SignalLens [21] and temporal to position map. The temporal to position map represents this region as a higher linear gradient.

RiverLens We include a hybrid of the SignalLens and river plot displays to augment the river plot with details-on-demand. The user is presented with the river plot which provides an overview of the series. Brushing temporal regions expands a time-series plot, with an overlay of the river. The river plot is shown either side to provide context (Figure 7).

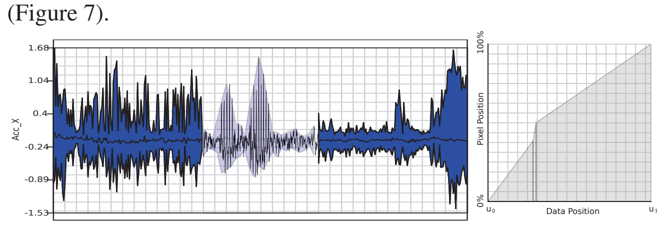

Fig. 7: RiverLens and temporal to position map. The zoomed in region features a lower-density of data items per pixel. The temporal to position map represents this region as a higher linear gradient.

ChronoLens Zhao et al. [38] present Chronolenses, an interactive, visual analysis lensing technique to support more elaborate data analysis tasks, without the need to derive new time series visualisations (Figure 8). Lenses are over-laid on top of traditional time-series graphs by user-selection and can display derived data, such as derivatives, and moving averages. Zoom, resize, and movement operations applied to lenses are used to overcome occlusion by magnification of the time-series by on the fly transformations of data points.

2.1.3 Layout based

Layout based techniques modify the spatial arrangement of the time-series to provide a linear mapping of time while transforming time-

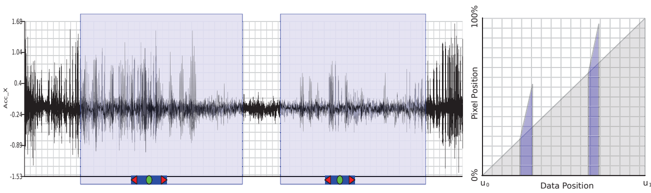

Fig. 8: ChronoLenses [38] and temporal to position map. Each lens is represented as a disjoint region overlaid on the map for the data range occupied.

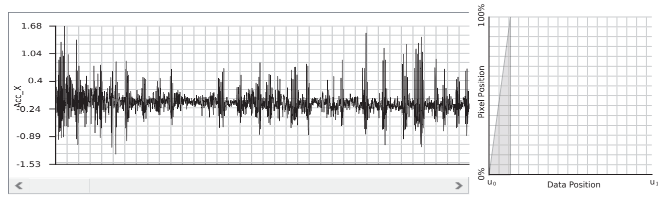

Fig. 11: Zoom plot and temporal to position map. The zoomed in view-port is displayed as a linear function on the map of the visible data range.

series graphs which enhance the display. Hao et al. [11] provide an early example of an interest-based visualization using an importance driven layout scheme applied to sets of time-series to perceive importance and hierarchical relationships.

Stack zoom Javed et al. [15] present stack zoom, a multi-focus zooming technique (Figure 9). Multi-focus zooming maintains context and temporal distance whilst zooming. User selection creates a hierarchy of zoomed line graphs, represented in a nested tree layout. Graphs are stacked on top of each other with the whole data set shown in a line graph at the root node. Each higher level zoom is represented as a new child node, stacked below the parent node. A layout manager maintains that the nodes on each level are temporally ordered. Coloring and arrows are used as visual cues to illustrate the positioning of child nodes in relation to their parent. The tree structure serves as a graphical history for communication purposes.

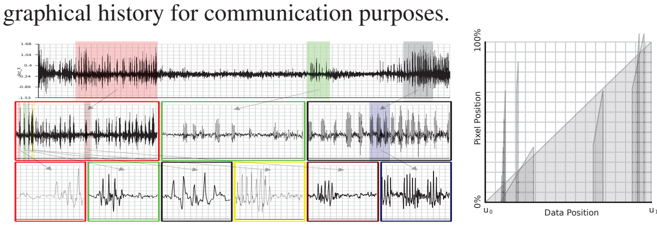

Fig. 9: Stack Zoom [15] and temporal to position map. Each node in the hierarchy is represented as a disjoint region overlaid on the map for the data range occupied.

Overview and detail Plaisant et al. [28] introduce the concept of overview and detail displays which provide simultaneous views of a focus region, along with an overview of the entire data series which provides context to the focus. A context line plot (Figure 10 top) visualizes the whole data set, from this, users can brush areas of interest to inspect in further detail. Selected data subsets are visualized in a separate focus line plot display (Figure 10 bottom), showing the data in more detail while still maintaining context with the whole data set.

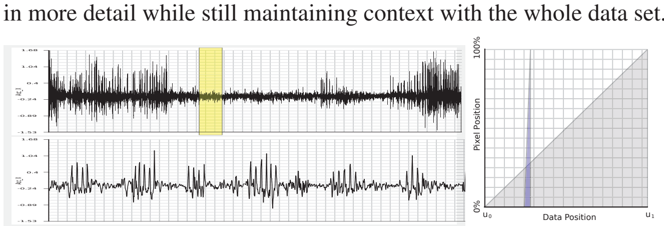

Fig. 10: Overview and detail plot [28] and temporal to position map. The overview is represented as a linear mapping of the entire data range. The detail is represented as a disjoint region overlaid on the map for the data range selected.

Zoom plot A zoom plot display embeds a time-series graph within a zoom-able widget which the user has control over the zoom level to define the level of detail they require (Figure 11). As the zoom is increased, the width of the time-series graph is expanded. The display view-port remains constant. Scroll bars allow the user to scroll smoothly through the expanded time-series.

While we have presented a graphical overview of the current state of the art techniques for time series visualization. Javed et al. [16, 15] provide a comparison of stack zooming against standard techniques for navigation of temporal data. They performed both design based and controlled user studies to assess advantages and disadvantages of multi-focus techniques versus overview+detail techniques. Findings include applicability of stack zooming approach to several scenarios and increase in performances when compared to current standard techniques. In the following section we propose a classification of the above literature not only in terms of visual encoding but in function of tasks plus domain.

3 Tasks and Design

In this section we present the domain level tasks and design of our visualization techniques.

3.1 Data

Biologists studying animals in their natural environment are increasingly using autonomous logging devices which record parameters such as acceleration, magnetic field intensity, pressure, light intensity and temperature [29, 4, 9]. These devices acquire large quantities of high quality time-series data from free-living animals which can be used to derive, and quantify, the behavior of the animals. It is widely acknowledged that this approach can enhance the understanding of ecological and behavioral processes.

3.2 Domain Characterization

Movement ecologists manually explore time-series graphs of several attributes to gain an understanding of the mapping from signal to behavior [30]. Often this is a combination of applying domain knowledge, inspecting environmental attributes, and deriving additional attributes (e.g., posture and energy use). After a preliminary investigation, the data is analyzed after labeling the behaviors of interest which have occurred throughout the duration of the deployment. This is a cognitively demanding and time-consuming process when considering data recorded at a high-frequency over long periods of time. The results of the analysis are presented to the wider research community. This typically involves extracting the signals of interest and annotating them, before inclusion in publications or presentations.

3.3 Exploration, Analysis and Presentation

Effective visualization assists the user in accomplishing the tasks they wish to undertake with their data. Ward et al. [37] identify *exploration*, *analysis* and *presentation* as three abstract tasks the user seeks to accomplish with visualization. Many of the techniques in Section 2 feature scalability issues. Approaches are in some way constrained to the display space which impacts their suitability for exploring large data. Chronolenses is suited for analysis. Stack zoom, which utilizes a hierarchical layout, is suited to presentation (communicating hypothesis and evidence). TimeNotes is built with all three tasks in mind.

We give further breakdown of these three abstract tasks.

Exploration and Analysis

Shneiderman [31] presents a task by data type taxonomy which lists several low-level tasks required to perform analysis and exploration in large data collections:

T1. *Overview* - Gain an overview of the entire collection of data.
T2. *Zoom* - Zoom in on data items of interest.
T3. *Filter* - Filter out uninteresting items.
T4. *Details-on-demand* - Select an item or group and get details when needed.
T5. *Relate* - View relationships among items.
T6. *History* - Keep a history of actions to support undo, replay such that the user can retrace their steps to show how they obtained their findings.

T7. *Extract* - Allow extraction of sub-collections and of the query parameters, so that once users have obtained what they desired, the set of parameters that facilitated their findings can be sent to others which illustrates the steps they undertook.

When specifically dealing with time-series data, one can also consider the task models by Andrienko and Andrienko [2] and McEachren [24] as presented and described in Miksch et al. [1].

The *elementary* tasks (Andrienko and Andrienko [2]) consist of data element look-up, comparison and relation seeking. *Synoptic* tasks involve patterns and trends in the data and relationships within data or to external data. For the list of tasks presented by McEachern, we test aspects of *identification* (in our tasks B and C – see section 5), and *localization* (in our tasks D and E – see section 5). In particular, our task E is high level in that it tests *when* behavioral patterns occur, and also *ordering* and *relationships* of alternate behavioral contexts. Specific examples of localization tasks (similar to synoptic tasks) are: Temporal Pattern: How often does a (behavioral) pattern occur? (Task E); Sequence: What order do (behavioral) patterns occur? (Task E); We also study Rate of Change (Task C). A user study by Borgo et al. [5] has studied the performance of Elementary tasks with respect to reading bar charts.

It can be seen that the time-series data tasks complement the Shneiderman task taxonomy. The former concentrate on the ability to analyze, where as the latter suggests the provision of the mechanisms and environment for that analysis to take place. In TimeNotes we provide functionality for all of Shneiderman's tasks, address elementary and synoptic tasks (of Andrienko and Andrienko [2, 1]) and identification and localization tasks (of McEachren [24, 1]) with regard to behavioral patterns. We also provide functionality for presentation intent.

Presentation

Aigner et al. [1] introduce three requirements for incorporating detected event instances into a visual representation which communicates to the user relevant information, namely communication, emphasis and conveyance.

None of the existing methods were built with presentation intents. In stack zoom, a disjoint nested tree view is automatically built during exploration which contains the hierarchical layout and acts as a management interface for constructing a presentation view. Each node is labeled with the stack coordinates, from which the user selects nodes and sub-trees to display. However, by using a separate view the tree loses the context with the underlying data and stack zoom layout to be fully regarded as a presentational tool.

4 TimeNotes

Hierarchical zooming [15] provides an efficient method of navigating through time-series by allowing the user to divide the information space and build a view of only the relevant data at the required granularity which also acts as an implicit graphical history of user actions.

In this section we present TimeNotes (available at `http://framework4.co.uk/`), inspired by the stack zoom approach. It contains additional features over and above stack zoom (such as, a flexible node-link layout, overlays, bookmarks, smooth curves to increase usability for reading hierarchies, fused interaction for presentation, and an integrated workspace with import/export of visualization state). Chronolenses does not include any hierarchical features which are the main focus of this work, but does include overlays and excellent analysis tools, although we offer similar functions and improved overlays. All of our new features facilitate exploration, analysis, and presentation of time-series data using hierarchical zooming. We demonstrate the features of TimeNotes that satisfy Shneiderman's task taxonomy and we test these features in the task based user study demonstrating their increased effectiveness.

On initialization of TimeNotes an overview of the whole data set is drawn on a time-series graph at the root node (T1) (these T numbers refer to Shneiderman's task taxonomy in Section 3.3). Applying rubber band selection across the series creates a new zoom level (referred to as a child node) of the selected data range in further detail (T2). Each node can be repeatedly drilled down such that a hierarchy is generated of the relevant data at the desired level of detail. A flexible node link layout is adopted to represent the generated hierarchy which allows the user to move nodes to a suitable location and size across the two-dimensional viewing plane. Nodes and whole sub-trees can be filtered (T3) by collapsing them to bookmarks. These can be later reopened for further inspection or to communicate the related data. Details on demand can be accessed for each node by viewing statistical summaries of the data contained in each node (e.g., min, max, skew) (T4). Snapping nodes together combines them into an overlaid time-series graph which allows relationships (i.e. frequency and amplitude) between temporal regions to be perceived (T5). The hierarchy created serves as a history of user actions [15] (T6) from which the user can construct a visualization of the relevant data sub-sections for exploring and analyzing the data. Data can be exported as raw (sensor) and/or derived (posture, energy use) for report inclusion (SVG export) or in interactive presentation (via our integrated work-bench) (T7). Data can be exported as CSV files. We built TimesNotes with presentation in mind.

Figure 13 illustrates TimeNotes on the Condor data set (see Section 2.1). The user has selected several repetitive flapping patterns across the data-series. The data is too dense to identify behaviors at the root level, but it is possible to gain an indication as to the presence of a behavior by inspecting the high energy portions of the signal. Zooming shows a detailed view of these allowing the user to differentiate between the signals. Interesting activity is bookmarked (on the center right of the data-series) which is minimized for the user to further explore later. We allow any node to depict any data channel. This image mainly depicts accelerometer (x-axis) with one switched to magnetometer to see thermalling behavior (from compass heading).

4.1 Layout

Many methods exist for representing hierarchical data structures effectively [3], we refer the reader to chapter 9 of Munzner [26] for an overview. The most common representational form for a tree is the node-link diagram [37]. This explicitly illustrates the relationship between parent-child nodes at the expense of the display space occupied by the visualization. Space filling techniques (e.g., tree-maps and stack zoom) attempt to optimally utilize the space they occupy, however they fall short at representing the hierarchical structure of the tree [35]. A comparison of these is shown in Figure 12. Perceiving the connection between nodes is vitally important to identify the context in which a signal occurs, which is often the case when undertaking higher-order tasks. The intuitive nature of the visualization need also be considered for presentation purposes where the learning aspect needs to be minimal.

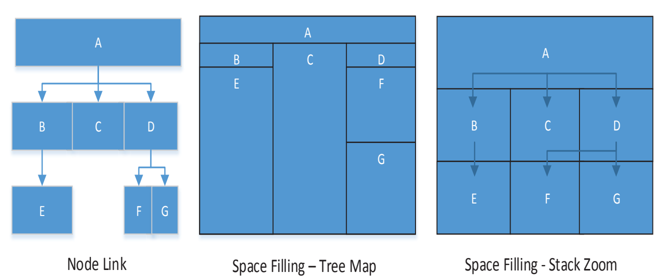

Fig. 12: Layout techniques for visualizing hierarchical data.

TimeNotes utilizes a space filling node-link diagram to represent the hierarchical zoom structure. Each child node defaults to being placed directly below its parent and within its horizontal and temporal extents (although the user may later move and resize). The allocation of display space for each child is proportional to the amount of data represented within that layer. If a new child is added or removed the space occupied by each child is recomputed so that the display space is used optimally.

Space allocation for each child node is computed using the following formulas to calculate the width $w_i = (s \times S_i)$, and 2D coordinates $\left(x_i = \sum_{j=0}^{i-1} w_j,\, y_i = d \times (h + ls)\right)$ of a child node i on the viewing

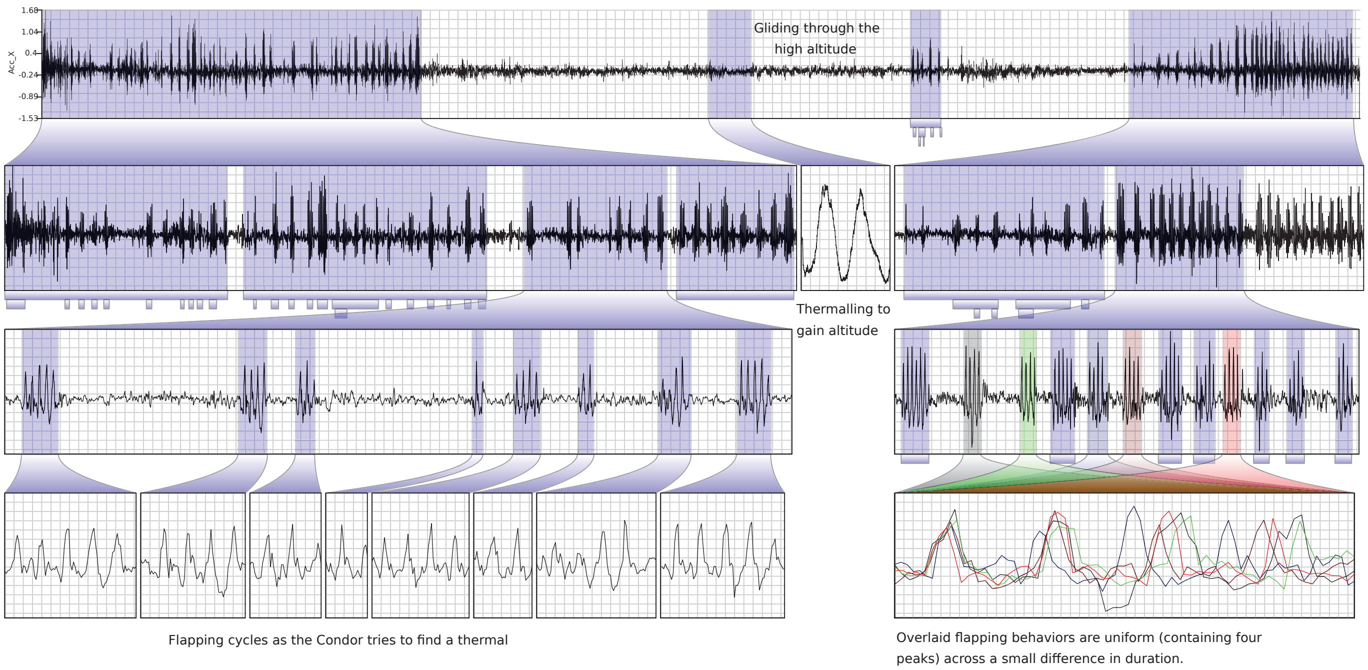

Fig. 13: TimeNotes displaying several flapping patterns across the data series. Firstly using the standard zoom approach (left tree) and using the stretched overlay view containing five segments overlaid (right tree). A thermalling activity is shown (center tree) using the magnetometer data.

plane, where $s = \left(w/\sum_{i=0}^{N} S_i\right)$ is the horizontal space per data item, w is the width of the parent node, N is the number of children, S is the set containing the number of data items each child represents, d is the current depth in the hierarchy, h is the height of each line plot, and ls is the spacing between each layer. This can be extended for nodes of varying height. A layout manager maintains the temporal ordering of the nodes from left to right for optimal chart readability. Ordering nodes also ensures that connections between layers do not overlap that would otherwise occlude the display and make the hierarchy difficult to trace through. A special case occurs where the temporal bounds of children overlap [15], for this, we only readjust the position when the entire data bound is not overlapping to minimize node movement.

The connection between each parent-child node is represented as a filled spline to provide a smooth transition through the hierarchy. Opacity is used so if connections overlap it is still possible to trace through the hierarchical structure. The user can change the color of connections and the distance between the parent and child nodes using the associated properties panel.

4.2 Node interaction and rendering

Each node in the hierarchy is represented as a 1D line-chart. The root node represents the whole data series, while all other nodes in the hierarchy represent a subset (x_1, x_2) of the data, where x_1 and x_2 represent the start and end indices of the data subset where $x_1 < x_2$ holds true. A number of interaction techniques are applied to each node to assist in completing the users intents.

Nodes may be resized in height and width by dragging markers on a node bounding rectangle, allowing the user to accentuate a node of interest. Free movement of nodes on the 2D plane allow the user to reposition a node at any location on the display. The auto-filling layout is disabled for a parent node once one of its direct children is moved, this avoids repositioning nodes the user has purposefully moved and preserves their mental map of the information space. The user can override this by unlocking the child node at any time.

The data bounds each child represents is overlaid in a transparent blue on the parent node. Grabbing the associated region with the cursor and dragging across the parent node pans through the data set. The child bounds $((x_{c1}, x_{c2}))$ must be constrained to the parent bounds $((x_{p1}, x_{p2}))$, such that, $x_{c1} \geq x_{p1}$ and $x_{c2} \leq x_{p2}$. This prevents the user panning beyond the extents of the data subset represented via the parent. Panning updates the range of data visualized in the child node and all subsequently related children further down the hierarchy by adding the offset moved to each of the child ranges visualized.

Hierarchical layouts are limited to the number of elements that can be visualized. To free up canvas space, nodes and whole sub-trees can be minimized. In doing so they are represented as a rectangular section below the parent node, which we call a bookmark. Hovering over a bookmark shows a graphical preview of the underlying data. Double clicking reopens the sub-tree, using the layout algorithm. Nodes can be deleted from the hierarchy using the delete key or right click menu. If the node is an intermediate node (i.e. it contains children) the user is asked if they would like to move the sub-trees of the deleted node to those of the parent node (see additional material and video). Similarly the user can disconnect subordinate nodes at any time and reconnect to any superior node in the hierarchy.

4.3 Overlay

Often during analysis, signal characteristics (e.g., frequency and amplitude) need to be compared. This can be difficult when they are positioned far away in the hierarchy or even side-by-side when there is a just noticeable difference. In TimeNotes, snapping nodes together (by dragging and dropping them on top of each other) overlays the nodes together into the same 1D line plot (Figure 13 right tree) to allow the direct comparison of signals. By default the plots are stretched such that they occupy the same display width in the visualization. This is useful when the phenomena varies in speed or time (see dynamic time warping [20]). When this is not the case, we allow the user to align the signals, left, central, or right to maintain temporal duration in relation to each other. Phase can be adjusted by moving the parent panning slider. Connections are mapped to the overlay plot, with a unique color applied to each which is also applied to each associated line in the overlay so the user can correlate between where each signal originates from and by association its temporal position in the data set. By using the right click menu the plots can be snapped away from each other and restored to their original location on the display.

4.4 Annotation

While interacting with the data, users may have comments or insights for themselves or others. Our annotation function allows text to be placed anywhere on the display space. Annotations can further be attached to nodes, such that they move with the node and are hidden when a node is minimized. Text can be resized and colored according to the users preference.

The data workspace may be saved at any time and reopened at a later date. This allows TimeNotes to be shared between individuals to communicate findings. This also allows the use of TimeNotes to create interactive presentations as an alternative to using current presentation software. This adds an extra dimension of engagement for viewers with the presentation. More importantly, it directly allows access to the original data which means the full context of the data can be shown during a presentation.

5 User Study

A user study was carried out to assess the effectiveness of our new visual design with respect to existing similar solutions. A comparison with ChronoLenses [38] was considered for testing low-level tasks focusing on pure data analysis, our main focus however was to test the power of using a hierarchical layout with explicit node-link relationships. In this context a comparison against Chronolenses would have been unfair as much of the hierarchical information is implicit in the visualization. TimeNotes design was significantly inspired by the approach proposed by Javed et al. [15] (which will be referred to as StackZoom for the remainder of the document), moreover Javed et al. [15] successfully compared their approach to existing state of the art techniques; we therefore decided to start from their findings, reproduce the stack zooming software according to its description in [15], and use it as our worthy antagonist.

To design our study we consulted with researchers in Biological Sciences to identify suitable tasks that would address questions of potential interest during the analytical process of charted information. A set of four major actions was identified: data traversal and labeling, trend detection and comparison. Further scrutiny allowed to group actions into two main categories: (hierarchy) navigation and comparison. Each action was then broken into its core components, each component refined and translated into a tasks generating a total of five main tasks: Leaf Counting (Task A), Amplitude Comparison (Task B), Frequency Comparison (Task C), Label Analysis (Task D), and Zoom/Pan and Labeling (Task E). To ensure consistency the same notation is used for the remainder of the document.

5.1 Tasks and Stimula Design

Hierarchy Navigation (Leaf Counting) - Task A. The objective of this task is to measure how well the user is able to traverse the hierarchy represented by the stimulus. The stimulus is a simulated hierarchical data interrogation where the user has made multiple selections drilling down to some detailed behavior in several leaf nodes. The top level depicts all of the data with up to three segments preselected and expanded in the second level of the hierarchy. Further selections are made resulting in internal nodes or leaf nodes. The task is to count the leaf nodes for each segment, entering the answer in the corresponding text box. The time is measured from the presentation of the stimulus until the user clicks on the submit button. There are six different hierarchies presented using the two visualization styles. Each question is presented twice. These twenty four stimuli are presented in random order with a constraint that the same stimulus must be at least three questions apart. Accuracy is measured as pass / fail on whether the user counted the correct number of leaves for that segment.

Comparison (Amplitude) - Task B. Biologists will compare the amplitude of behaviors across the data set. For example, the strength of a wing beat. This task is designed to measure the effectiveness of our new overlay function. The stimulus is a simulated hierarchical data interrogation where two leaf nodes are brought into close proximity to compare the amplitude of the signal. We compare bookmark charts with and without the overlay function, and stack zoom charts. In this task there are three visualization types, six hierarchies, and each question is presented twice. These thirty six stimuli have the same constraint on random order as above. Timing is from stimulus presentation until clicking on the submit button. The user must select whether the left or right signal has the highest amplitude. Accuracy is measured as pass / fail.

Comparison (Frequency) - Task C. Feature frequency within the behavior are compared across the signal (e.g., the speed of a wing beat). This task is designed to measure the effectiveness of our new overlay function. The stimuli follow the same pattern as task B except the user now determines which leaf node has the highest frequency.

Hierarchy Navigation (Zoom/Pan and Labeling) - Task D. We simulate the behavioral labeling process that biologists undertake with this task. To simplify the task we highlight the behaviors in the signal with a gray block. Users are required to indicate whether each one is behavior A or B (by right clicking in the block and selecting the appropriate label). The block turns to the color representing that behavior. Within this task we increase participant degree of freedom as they are allowed to interact with the hierarchy via panning, selection and labeling. The introduction of pan and zooming feature can lead to loss of context [8] we therefore decided to test the effectiveness of this feature within our system. We present twelve stimuli where half enable the pan function.

With the pan function enabled, users are able to grab the segment at a higher level of the tree and move it left and right, scrolling the signal at the lower level of the tree. It provides a mechanism to traverse the time-line allowing inspection of the data at a zoom level the user feels comfortable with. We present the whole time-series data, and allow the user to interact with the time-series in any way they choose, apart from the constraint that only half of the stimuli allow the pan function.

We provide a counter of the number of segments left to label. We ask users to target getting this to zero, but not to spend minutes looking for the last remaining one or two segments. We time from the presentation of the stimuli until the user clicks next. Accuracy is measured according to how many behaviors are correctly labeled.

Hierarchy Navigation (Label Analysis) - Task E. The objective of this task is to measure a more complex use of the hierarchy. Firstly, we test how well the user is able to locate target behaviors in the hierarchy. This involves scanning the hierarchy for a specific pattern. Secondly, we test how well the user is able to relate the found pattern to the overall data time-line which is a critical function for understanding time-series data. Thirdly, we test how well a user is able to perceive temporal ordering of the remaining patterns.

Our chosen task to fulfill these conditions is to present a hierarchy to the user where the data has been segmented and labeled using two contrasting behaviors (A and B). The user must locate the first occurrence of a segment labeled as A. This tests scanning and relation to the time-line. They must also count, and label, how many occurrences of segments labeled B precede it. This tests the temporal ordering implied through the hierarchy. It also requires hierarchy navigation.

We enforce the last condition by setting line transparency to 10% in the second highest level of the hierarchy. At the top level, participants are unable to discern whether labeled segments are A or B because the signal is too dense. At the second level, this may be possible, and therefore the segments could be counted without referring to the hierarchy. By setting lines to 10% transparency, participants cannot use this short-cut and are forced to refer to the presented hierarchy. In real situations data would be dense, and therefore the hierarchy would be used, or it would not matter if a short-cut is found and taken. It is for the purposes of the user study where we want to test the effectiveness of the hierarchies that we must employ this.

5.2 Study Hypothesis

In the comparison between TimeNotes and StackZoom we formulated the following hypothesis:

- H1 Task A - TimeNotes will perform faster than StackZoom. We think that being a standard counting operation, with no time limit, both visual designs will perform equally in terms of accuracy; however the increase in clarity with respect to hierarchy linking structure, will help participants to find targets more quickly with TimeNotes.
- H2 Task B and C - TimeNotes with Overlay will perform faster and more accurate than both TimeNotes without Overlay and StackZoom. Since overlay exploits basic Gestalt principles such as

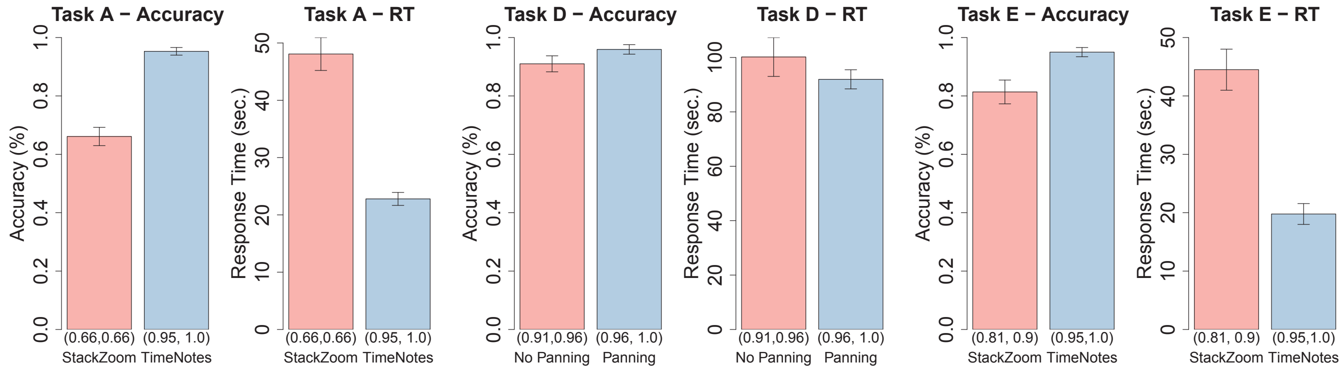

Fig. 14: Analysis of performance results for Tasks A, D and E, (mean, median) values are indicated below each bar. Error bars show 95% confidence intervals.

grouping effect, the automatic alignment and overlapping of features should ease the mental task of computing differences in amplitude, which is reduced to measuring distance between inner and outer boundaries, and distances between frequency peaks. Without overlay the user needs to mentally perform both tasks of alignment and translation into a unique system of reference. We therefore think that the overlay feature will allow participants to be both faster and more accurate than similar visual designs without such feature. No difference is expected between TimeNotes without Overlay and StackZoom.

H3 Task D - Visual design with panning option will perform faster than same visual design without panning option. We think that panning (present in both StackZoom and TimeNotes) is a crucial feature when performing analysis of charted data. Panning allows to reduce the hierarchy growth, inevitable when only zooming option is available, and increase space usage.

H4 Task E - TimeNotes will perform faster than StackZoom. We think that being a standard searching operation, with no time limit, both visual designs will perform equally in terms of accuracy; however the increase in clarity with respect to hierarchy linking structure, will help participants to perform traversal and target search more quickly with TimeNotes.

5.3 Study Analysis

A pilot study was carried out involving eight participants including: co-authors, four postgraduate students and one member of our research staff. Together with the collection of preliminary results we aimed at testing length of tasks and study, to avoid confounding effects due to tiredness, randomization of stimula, to ensure that repetitions of the same stimula were not apparent within a task, robustness of the study interface. The five non-author participants were unaware of any of these factors. Pilot study results were positive and revealed trends in the data supporting our initial hypothesis, minor issues with the interface, especially with respect to Task D the only one involving direct interaction, were also noted; all issues were addressed for the main study.

The final study therefore consisted of five tasks, 128 stimula, two visual designs. Supplementary material contains the presentation used for participants training.

5.4 Experimental Setting

Participants. A total of 30 participants (2 females, 28 males) took part in this experiment in return for a £10 book voucher. Participants belonged to both the student and academic communities. Prerequisites to the experiment were basic knowledge of Calculus such as line charts, phase, frequency, amplitude and familiarity with concepts such as hierarchies and hierarchical organization of data, for this reason recruitment was restricted to the departments of Mathematics, Physics, Computer Science and Engineering, and in the case of students, year 2 and above. Ages ranged from 20 to 54 (Mean=25.34, SD=8.27). All participants had normal or corrected to normal vision and were not informed about the purpose of the study prior to the beginning of the session.

Apparatus. The visual stimuli and interface were created using custom software written in C++ with OpenGL and QT as the graphics library. Experiments were run using Intel 2.8GHz Quad-Core PCs, 4GB of RAM and Windows 7 Enterprise. The display was 19" LCD at 1440×900 resolution and 32bit sRGB color mode. Each monitor was adjusted to the same brightness and level of contrast. Participants interacted with the software using a standard mouse at a desk in a dimmed experimental room. The absence of windows in the room allowed us to maintain a constant and uniform lighting environment.

Procedure. The experiment began with a brief overview read by the experimenter using a predefined script. Detailed instructions were then given through a self-paced slide presentation. The presentation included a description of the study and also a briefing on how to interpret each visual design and, in the case of Task D, how to interact with both designs for labeling purpose. Participants also received a color copy of the presentation for reference during the study if desired. The experiment was divided into 5 main blocks with a chance of resting between each block.

All five tasks were completed in sequential order. Given the nature of the experiment each section assessed a different aspect of the analytical process performed by Biologist as described in section 5.1. Maintaining the same section order for each participant meant that each participant experienced similar experimental conditions. This increased robustness of the analysis of the collected data. Randomness was introduced at trial level. Within a task, trials were randomized to avoid learning effects. A training section preceded each task to familiarize the participant with both task and visual layout.

For Task A, D and E a total of 4 practice trials (two per visual layout) were completed, for Task B and C a total of 6 trials (two per visual layout, with 3 layout options presented in these tasks) were completed. Each training trial included a feedback to the participant regarding the correct answer. Participants were invited to take a short break at the end of each task, if needed. Participants were invited not to take breaks once a task had commenced.

The study was closely monitored, at least two experimenters were always present in the room and participants abode to the study requirements. At the end of each task a short multiple choice questionnaire was presented to collect qualitative information from the participant. At the end of the study each participant completed a short post-experiment debriefing interview and questionnaire to collect demographic and further qualitative information. The purpose of questionnaire and debriefing was to obtain comments and recommendations concerning both the experimental procedure, design and usability of both visualizations. Due to the qualitative nature of the feedback, results were used to support the discussion of quantitative results gathered from the testing phase. Both visualizations were at all times presented as valid options, especially during post-processing interview, to maintain unbiased judgment and preserve validity of the collected qualitative feedback.

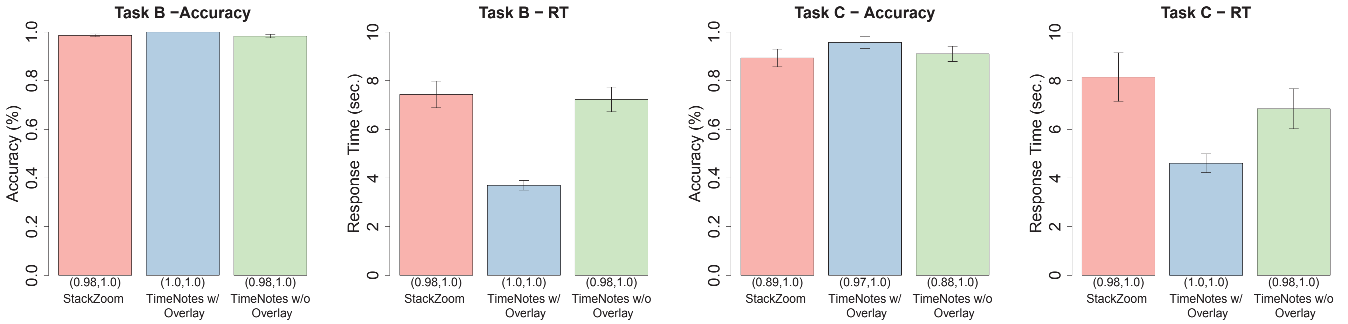

Fig. 15: Analysis of performance results for Tasks B and C, (mean, median) values are indicated below each bar. Error bars show 95% confidence intervals.

5.5 Analysis Results

In our analysis we mainly considered the effect of task vs. visual encoding. We focused on a comparison of the newly designed TimeNotes performances against the StackZoom approach as this was our primary research question. To perform our analysis we first tested data normality via a Shapiro-Wilk test, more appropriate for small sample sizes. For normally distributed data a repeated measure analysis of variance (ANOVA) was used to test for differences between groups, when data characterized a non-normal distribution, instead of a non-parametric distribution, the Friedman test was used. Both tests were performed assuming a standard significance level $\alpha = 0.05$ to determine statistical significance between conditions. For non normally distributed data post-hoc analysis was performed via separate Wilcoxon signed rank-tests on related groups for which significance was found. Significance threshold was adjusted using a Bonferroni correction, with corrected significance value of $\alpha = 0.016$ for Task C and D. No cases where found in which both time and error data produced significant results, therefore no correlation analysis, testing the presence of a trade-off effect (e.g., less time leading to more errors) was required.

Hierarchy Navigation (Leaf Counting) - Task A Performance in Task A, summarized in Figure 14 as a function of visual design, revealed a noticeable variation between conditions, the Shapiro-Wilk test revealed a normal distribution of performance for both TimeNotes ($p \leq 0.8$) and StackZoom ($p \leq 0.3$). The ANOVA's test showed a significant main effect in response time ($p \leq 0.02$). Accuracy data revealed a non normal distribution, the Friedman's test showed a significant main effect ($\chi^2 = 25.13$, $p \ll 0.02$). A closer analysis showed:

- Mean Accuracy: TimeNotes (mean = .95) significantly more accurate than StackZoom (mean = .66) ($p \ll 0.001$);
- Mean Response Time: TimeNotes (mean = 2.27) significantly faster than StackZoom (mean = 4.81) ($p \ll 0.001$);

Comparison (Amplitude) - Task B. Performance in Task B, summarized in Figure 15 as a function of visual design revealed a noticeable variation across conditions, Friedman's test showed a significant main effect in both accuracy ($\chi^2 = 5.43$, $p \leq 0.02$) and response time ($\chi^2 = 42.07$, $p \ll 0.001$). A closer analysis showed:

- Mean Accuracy: TimeNotes with Overlay (mean = 1.0) significantly more accurate than TimeNotes without Overlay (mean = .98) ($p \ll 0.016$) and StackZoom (mean = .98) ($p \ll 0.015$);
- Mean Response Time: TimeNotes with Overlay (mean = 3.69) significantly faster than TimeNotes without Overlay (mean = 7.23) ($p \leq 0.001$) and StackZoom (mean = 7.43) ($p \ll 0.001$);

No other significant differences were found.

Comparison (Frequency) - Task C. Performance in Task C, summarized in Figure 15 as a function of visual design revealed a noticeable variation across conditions, Friedman's test showed a significant main effect in both accuracy ($\chi^2 = 7.68$, $p \leq 0.02$) and response time ($\chi^2 = 32.07$, $p \ll 0.001$). A closer analysis showed:

- Mean Accuracy: TimeNotes with Overlay (mean = .97) significantly more accurate than TimeNotes without Overlay (mean = .88) ($p \leq 0.016$) and StackZoom (mean = .89) ($p \ll 0.012$);
- Mean Response Time: TimeNotes with Overlay (mean = 4.46) significantly faster than TimeNotes without Overlay (mean = 7.66) ($p \ll 0.001$) and StackZoom (mean = 8.15) ($p \ll 0.001$);

No other significant differences were found.

Zoom and Pan (Labeling) - Task D Performance in Task D, summarized in Figure 14 as a function of visual design with panning option and without, revealed a significant variations across conditions with respect to response time. Friedman's test showed a significant main effect ($\chi^2 = 3.2$, $p \leq 0.05$) with mean response time for visual design with panning option (mean = 91.9) significantly more accurate than without panning option (mean = 105.68) ($p \ll 0.04$). Further analysis was performed on accuracy with respect to correctly labeled versus wrongly labeled events. Friedman's test showed a main effect ($\chi^2 = 3.00$, $p = 0.059$). Post hoc analysis with Wilcoxon signed-rank tests revealed mean accuracy for visual design with panning option (mean = 0.96) significantly more accurate than without panning option (mean = 0.91) ($p \leq 0.046$).

Hierarchy Navigation (Label Analysis) - Task E Performance in Task E, summarized in Figure 14 as a function of visual design, revealed a noticeable variation between conditions, the Shapiro-Wilk test revealed a normal distribution of performance for both TimeNotes ($p \leq 0.13$) and StackZoom ($p \leq 0.24$). The ANOVA's test showed a significant main effect in response time ($p \ll 0.001$). Accuracy data revealed a non normal distribution, the Friedman's test showed a significant main effect ($\chi^2 = 4.0$, $p \leq 0.046$). A closer analysis showed:

- Mean Accuracy: TimeNotes (mean = .95) significantly more accurate than StackZoom (mean = .81) ($p \ll 0.001$);
- Mean Response Time: TimeNotes (mean = 19.78) significantly faster than StackZoom (mean = 44.5) ($p \ll 0.007$);

5.6 User Study Discussion

All hypotheses stated in section 5.2 were confirmed by our study. We also reached significant differences in accuracy across all tasks without any trade-off effect.

Task A's and E's unexpected increase in accuracy results further confirmed the effectiveness of hierarchical visual layout for data navigation and target identification, also noted by Javed et al. [15]. Differences in both accuracy and response time of TimeNotes versus StackZoom in Task A and E can be explained by the increase in the TimeNotes design of information grouping. Information grouping has a strong effect on both perception and attention [33], in TimeNotes this is achieved by strengthening the semantic relationship of the node-link structure. The use of colors in StackZoom might have also added an extra layer of visual complexity which would affect the process of information decoding, e.g. color interpretation for every hierarchy level. Post-experiment interviews also confirmed participants preferences towards TimeNotes visual encoding of the hierarchy (*"Counting leaves in Task A was hard with arrows as it made parents ambiguous."* (anon.), *"I tried using colors [instead of arrows] and a depth first* *search approach." (anon.) "While the stack plot is usable, the way it subdivides the data does not feel natural. You have to think for a second about where the links go. With the bookmark plot traversing the hierarchy feels automatic and natural." (anon)*).

Task B and C reached significance in both accuracy and response time, this leads to the conclusion that overlay is an important feature when considering comparison/estimation tasks. It is fair to note that, especially in Task B, accuracy results are close to optimal. It would be of interest to see how an increase in the sample size would affect the emerging trend. It is also worth noting that in both tasks we only tested pairs, it would be interesting to also test the effect on accuracy when increasing the number of comparisons to more than two signals.

In Task D we tested the effect that increasing the degree of freedom by introducing panning, zooming and labeling would have on user performances in the context of our visualization interface. Panning is a ubiquitous style of navigation in present-day user interfaces, however in the analysis of large sets of data increasing interactions can lead to loss of context [8]. In Jetter et al. [18] a comparison of panning vs. zooming with panning is performed. Results showed increasing in performance for the former but not for the latter. Task D confirmed the effectiveness of panning, integrated within a hierarchical layout, when dealing with visual search tasks. When sufficient level of detail is reached, while preserving context through the hierarchy linking structure, panning facilitates quick scanning of the zoomed in data (post-experiment feedback: *"Task D. Where panning was not available I had to add larger views and then repeatedly remove them after labeling" (anon.), "The zooming ability made it very easy to zoom in to a usable level and then simply slide across the plot, labeling the data as you come to it in an efficient manner. Without panning however was very much a guessing game to find the data, and with continuously adding and removing levels many times I miss clicked and removed a node when I meant to label a segment instead." (anon)*). Data collected also showed a steep difference between the number of events missed with panning enabled (53 miss) and no panning (110 miss).

Task D and Task E were the most complex tasks as they allowed participants to actively interact with the hierarchy through zoom (e.g. creation of lower hierarchy levels), pan and labeling. Both tasks feature a high accuracy rate with faster response when data segmentation is provided (Task E), e.g. participants need to restrict target search only to highlighted regions. Results in response time of Task D vs. Task E confirms the complexity of visual search tasks even more prominent when handling dense collections of data.

6 Field Study Discussion

We have been conducting a longitudinal study on interactive visualization software use with biologists over 5 years. We have provided tools for labeling behaviors using visualization, template matching, visual analytics and machine learning [4, 9, 36]. This work on hierarchical chart visualization initiated from the way in which they work with and present behavioral patterns.

For the field study, we provided training on the new TimeNotes feature in the already familiar software package. The biologists utilized the software (already installed on their machines) to inspect their own animal data. We observed them using the software and answered any questions which were raised. We held a session after everyone was well acquainted with using the software and held a focus-group like discussion where they discussed features. We summarize their feedback below. We meet with the biologists on a regular basis (every couple of weeks). They provide us with a constant source of feedback which we use to provide innovative solutions to their problems. TimeNotes was inspired by this continual cycle of changes.

Zooming and panning provides overview and efficiency. At an overview level they need to know whether the deployment was successful. Did the sensor collect the data? Did all data channels collect? Does it look right? Did it work for the duration of the collection period? Does the attachment on the animal shift at all? All these can be answered in a short time by inspecting the sensor trace. Either at a global level for the former questions, or by zooming and panning the data for the latter question.

We demonstrated in the formal user study (task D) that labeling benefits from zooming and panning. Feedback from biologists concurs this. *"Its very cool that you can zoom in without losing context". "We can see from the signal that the collar has shifted at this point in time".*

Hierarchical layout aids side-by-side comparison and thinking of behavior at different scales. To support behavior discovery they use an overview of the data (for time of day and duration of behavior over larger scale) and zoom down to finer behaviors, for example a wing beat. These are commonly hierarchical. For a sea bird, the top contains many feeding sequences. Each sequence contains behavior to hyperventilate, dive, swim, prey capture, ascend and rest. Biologists label the overall sequence, then consider each dive sequence, label each component part, and compare across sequences. They view different sensors to see if the behavior has been captured more fully in a different axis to others, or for example environmental sensors to observe water pressure for dive depth. Our bookmark design provides a clearer link between levels of the hierarchy.

We demonstrated in the formal user study (tasks A and E) that the bookmark hierarchy provides richer context than the stack zoom hierarchy through clearer organization of the hierarchy. They can pan at any level in the hierarchy with lower levels either moving in synchronization (padlocked together) or float on their own. *"This is so good for picking out individual behaviors".*

Overlay aids comparison. By enabling the *snapping* together of two chart windows, two disparate parts of the signal can be brought into immediate focus for comparison of amplitude or phase. For example a tag is often deployed on a few animals in quick succession, and utilizing this function the same behavior in different animals can be compared for speed and vigor. We demonstrated in the formal user study (tasks B and C) that the overlay function is more accurate and faster than side-by-side comparison.

Factoring in presentation functionality aids communication. When communicating behavior, the typical work flow in previous software was to locate the behavior, export that segment as raw data, read it into Excel, create a line chart, export that to powerpoint and annotate the behavior. Likewise for inclusion in publications. This loses contextual information, and also typically required a new data (visual) scanning to locate the behavior. We built in the idea of presentation by allowing the user to reposition bookmarks. Levels can be minimized and maximized. Text can be associated with bookmarks to add annotation. The work area may be saved for future reference. Line charts can be exported as raw data or as SVG. All figures in this paper are generated using the software, with the benefit that the SVG is vectorized, aiding zooming within publications. This is not tested in the formal user study since it is a feature specific to this software and trivially faster and preferable to the above mentioned work flow. Feedback is that this will be useful for presentations, teaching and will speed up results inclusion in publications.

7 Conclusion

In this paper, we comparatively evaluate existing methods for exploring time-series data, giving a graphical overview of each and classifying their ability to explore and interact with data. Based on this, we introduce TimeNotes, a hierarchical navigation technique for visualizing and interacting with time-series data. We undertake and report an empirical study and a field study. We test both static and interactive features of our environment confirming validity of state of the art techniques and their integration into a novel approach. Our findings illustrate that TimeNotes provides a more effective working environment for the exploration, analysis, and presentation of time-series data. We also presented RiverLens (Figure 7) as a combination of River Plot [7] and SignalLens [21]. We did not comment or evaluate this any further in the paper because it arose as part of our discussion and evaluation with biologists, who wished to see this combination in a future version of the software. As future work we would like to incorporate it into TimeNotes and evaluate it. We seek to perform an additional evaluation of TimeNotes in respect to user interaction with our interface, and explore the use of interactive lenses in conjunction with a multi-focus interface.

REFERENCES

[1] W. Aigner, S. Miksch, H. Schuman, and C. Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction. Springer Verlag, 1st edition, 2011.

[2] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer, 2006.

[3] D. Archambault, T. Munzner, and D. Auber. Topolayout: Multilevel graph layout by topological features. *Visualization and Computer Graphics, IEEE Transactions on*, 13(2):305–317, March 2007.

[4] J. Blaas, C. P. Botha, E. Grundy, M. W. Jones, R. S. Laramee, and F. H. Post. Smooth graphs for visual exploration of higher-order state transitions. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):969–976, 2009.

[5] R. Borgo, J. Dearden, and M. Jones. Order of magnitude markers: An empirical study on large magnitude number detection. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2261–2270, Dec 2014.

[6] P. Buono, A. Aris, C. Plaisant, A. Khella, and B. Shneiderman. Interactive pattern search in time series. *Proceedings of SPIE*, 5669(1):175–186, 2005.

[7] P. Buono, C. Plaisant, A. Simeone, A. Aris, B. Shneiderman, G. Shmueli, and W. Jank. Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting. In *Information Visualization, 2007. IV '07. 11th International Conference*, pages 191–196, July 2007.

[8] M. Glueck, T. Grossman, and D. Wigdor. A model of navigation for very large data views. In *Proceedings of Graphics Interface 2013*, GI '13, pages 9–16, Toronto, Ont., Canada, Canada, 2013. Canadian Information Processing Society.

[9] E. Grundy, M. W. Jones, R. S. Laramee, R. P. Wilson, and E. F. Shepard. Visualization of sensor data from animal movement. *Eurographics/ IEEE-VGTC Symposium on Visualization (Eurovis) 2009, Computer Graphics Forum*, 28(2):815–822, June 2009.

[10] M. Hao, U. Dayal, D. Keim, and T. Schreck. Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. In K. Museth, T. Moeller, and A. Ynnerman, editors, *Eurographics/ IEEE-VGTC Symposium on Visualization*. The Eurographics Association, 2007.

[11] M. C. Hao, U. Dayal, D. A. Keim, and T. Schreck. Importance-driven visualization layouts for large time series data. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 203–210, 2005.

[12] M. C. Hao, M. Marwah, H. Janetzko, U. Dayal, D. A. Keim, D. Patnaik, N. Ramakrishnan, and R. K. Sharma. Visual exploration of frequent patterns in multivariate time series. *Information Visualization*, 11(1):71–83, 2012.

[13] C. Holz and S. Feiner. Relaxed selection techniques for querying time-series graphs. In *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 213–222, New York, NY, USA, 2009. ACM.

[14] P. Isenberg, A. Bezerianos, P. Dragicevic, and J.-D. Fekete. A study on dual-scale data charts. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2469–2478, 2011.

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

[16] W. Javed and N. Elmqvist. Stack zooming for multifocus interaction in skewed-aspect visual spaces. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1362–1374, Aug. 2013.

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

[18] H.-C. Jetter, S. Leifert, J. Gerken, S. Schubert, and H. Reiterer. Does (multi-)touch aid users' spatial memory and navigation in 'panning' and in 'zooming &; panning' UIs? In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 83–90, 2012.

[19] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. *Mastering the Information Age - Solving Problems with Visual Analytics*. Eurographics Association, 2010.

[20] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, Mar. 2005.

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

[22] R. Kincaid and H. Lam. Line graph explorer: scalable display of line graphs using focus+ context. In *Proceedings of the working conference on Advanced visual interfaces*, pages 404–411. ACM, 2006.

[23] J. Lin, E. Keogh, and S. Lonardi. Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization*, 4(2):61–82, July 2005.

[24] A. MacEachren. *How maps work: representation, visualization, and design*. Guilford Press, 1995.

[25] 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, 2008.

[26] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[27] C. Pennycuick. Speeds and wingbeat frequencies of migrating birds compared with calculated benchmarks. *Journal of Experimental Biology*, 204(19):3283–3294, 2001.

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

[29] Y. Ropert-Coudert and R. P. Wilson. Trends and perspectives in animal-attached remote sensing. *Frontiers in Ecology and the Environment*, 3(8):437–444, 2005.

[30] E. L. C. Shepard and L. G. Halsey. Identification of animal movement patterns using tri-axial accelerometry. *Endangered Species Research*, 10:47–60, 2008.

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

[32] J. T. Stasko. The value of visualization...and why interaction matters, capstone speech, eurovis 2014, 2014.

[33] A. Treisman. Perceptual grouping and attention in visual search for features and objects. *Journal of Experimental Psychology: Human Perception and Performance*, pages 194–214, 1982.

[34] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, second edition, 2001.

[35] J. J. Van Wijk and H. Van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 73–78. IEEE, 1999.

[36] J. Walker, M. W. Jones, R. Laramee, O. Bidder, H. Williams, R. Scott, E. Shepard, and R. Wilson. Timeclassifier: a visual analytic system for the classification of multi-dimensional time series data. *The Visual Computer*, 31(6-8):1067–1078, 2015.

[37] M. O. Ward, G. Grinstein, and D. Keim. *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010.

[38] 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, Oct. 2011.