GraceFall: Visualizer for Diverse Wear-Out Reliability Degradation Data Spanning Multiple Time Scales

Matthew Tang and Ian Hill

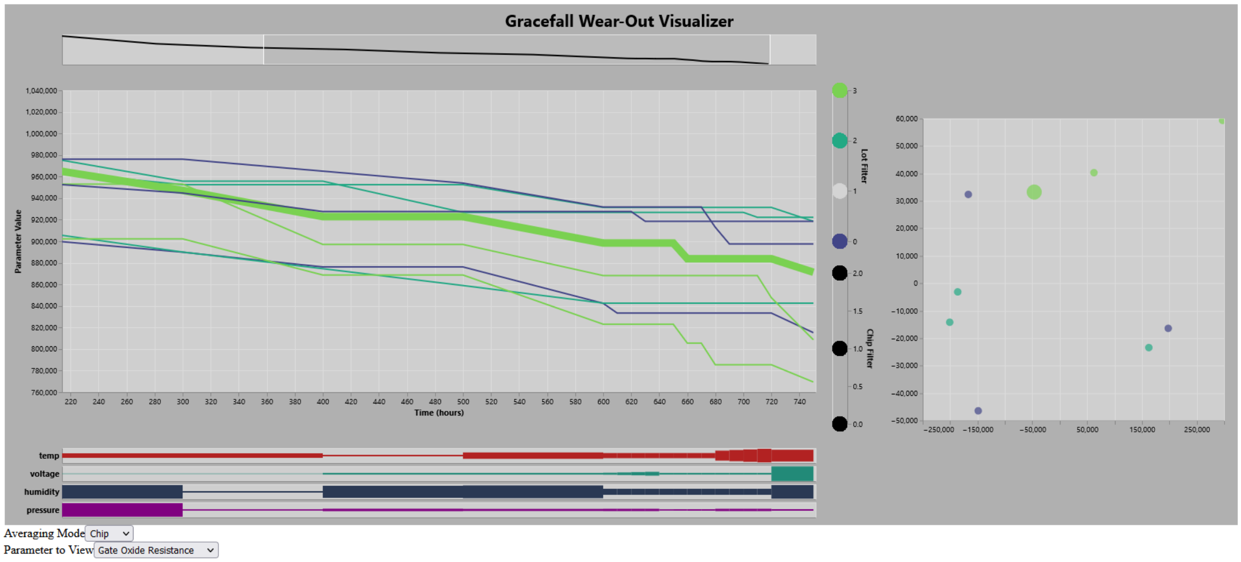

Fig. 1. Interface of the *GraceFall* visualization tool running on a data set showing degradation of gate oxide resistance for integrated transistors. Linked highlighting between a data series and the corresponding PCA mark is shown. All code to run the tool and the dataset seen here can be found on the project repository page at `https://github.com/ianrmhill/gracefall` [6].

Index Terms—Multiple time scale, many series summary, integrated circuit reliability.

1 Introduction

Wear-out reliability is a niche but critical aspect of modern electronics design. Testing and planning for degradation and failures as computer chips age is incredibly challenging; nearly all product development cycles are much shorter than the required product lifespans and so accelerated stress testing followed by extrapolation/prediction is leveraged in lieu of direct testing. These testing practices are the core focus of one author's (Ian's) doctoral thesis topic, and several shortcomings of existing reliability engineering practices were identified in a survey paper published in IEEE Transactions on Device and Materials Reliability [5]. A simulator has already been developed that enables stochastic temporal simulation of wear-out processes and Bayesian inference on wear-out models to address two of these shortcomings.

Whether wear-out test data is sourced from this simulator or physical experiments, engineers must interpret and process the conducted measurements to ensure the test was carried out successfully, that wear-out reliability requirements are met, and that unexpected results are investigated and attributed to stress conditions or other test parameters appropriately. This project introduces a visualization tool that can aid reliability engineers in carrying out the listed activities in an effort to overcome the conceptual challenges of modern reliability testing. These difficulties include non-constant stress conditions, probabilistic physical models, and increasingly diverse reliability requirements that engineers are not likely to yet be accustomed to.

Our preliminary investigation found that in all anticipated use cases engineers will need to compare large quantities of scalar data series arising from conducted test measurements under multiple stress configurations through time. The introduced solution, *GraceFall*, handles diverse line graphs with dynamic time scales to aid in these exploratory analysis tasks by providing a diverse range of statistical analysis techniques ranging from dimensionality reduction to view filtering to data aggregation. Additionally, *GraceFall* integrates with the aforementioned simulator to allow for rapid iteration on test design, simulation, and analysis tasks.

2 Related Work

There are two classes of related works to discuss in order to place this project in context, those within the integrated circuit wear-out reliability space, and then those within the field of info visualization.

2.1 Integrated Circuit Reliability Visualizations

Integrated circuit reliability is not well known for effective visualization tools, as the field is reasonably niche and driven by individual companies within the semiconductor industries. Reliability reporting is mostly opaque to end users of integrated circuit products, thus visual representations of reliability and reliability tests are often neglected. Typically, visualizations are focused purely on summary statistics for reporting and obscure all the data that produces these resulting statistics. Common visual representations of reliability test data include Weibull plots of cumulative failures against time 2, or box plots showing degradation distributions for different devices 3.

We believe a key missing element towards effective understanding of wear-out processes in existing reliability visualizations is temporal comparison. We hypothesize that there are three main reasons why

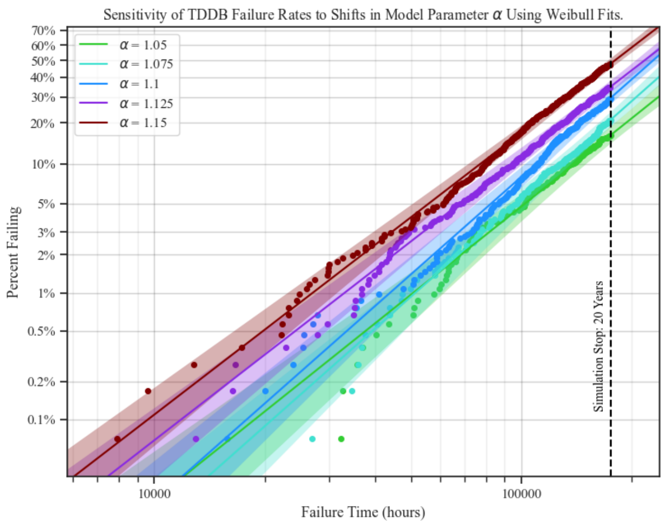

Fig. 2. Sample Weibull plot from Ian's recent conference submission.

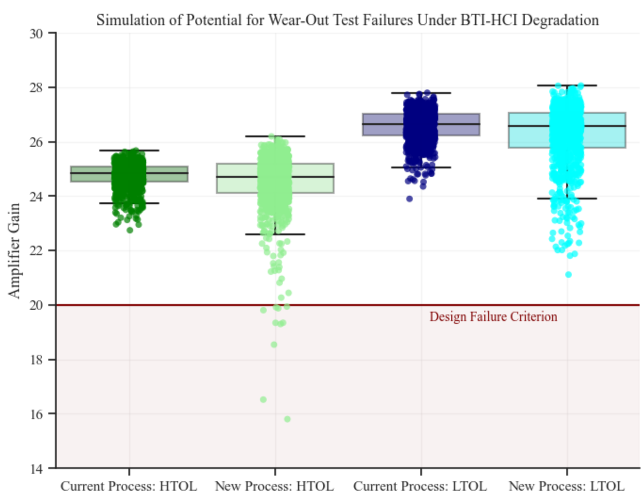

Fig. 3. Sample end-of-test degradation boxplots from Ian's recent conference submission.

temporal visualizations of wear-out are so rarely seen: (1) collecting measurements during stress is costly and thus rarely is sufficient data available for meaningful trends to be evident within the data; (2) product reliability requirements are typically based on failure rates at some time instant, thus results are often only displayed for that point; and (3) visualizing the evolution of derived statistical quantities through time is challenging from a design perspective as many channels are needed and large sample sizes are prone to visual clutter. The first two reasons for rarity are not applicable in the case of the developed simulator, which imposes no measurement costs and is intended for developing models and tests, not for evaluating product reliability at arbitrary end-of-supported-life instants.

Some additional existing wear-out visualizations published by one of the most recognizable researchers in the field are shown in Figure 4, highlighting the prevalence of visual clutter, overuse of contextual labelling to frame data, and minimal consideration of how degradation and/or failures evolve over time [13]. Interested readers can find even more sample visualizations in the works of Park et al. and Chuang Yang [14, 19]. For Ian's research, visual representations are needed to eventually help engineers explore test design spaces and compare different probabilistic models. The existing visualizations such as those shown are not effective for design space exploration tasks within this domain as they do not provide little to no information on the stress resulting in wear-out or the models used to produce or fit degradation and/or failure data.

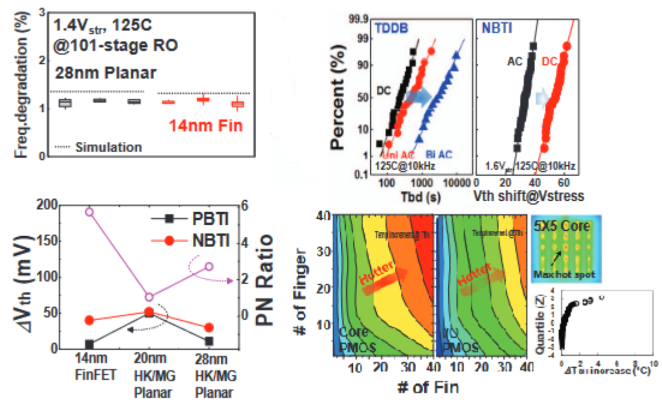

Fig. 4. Examples of accepted visualizations within the IC reliability research community from a well-regarded paper [13]

2.2 Relevant Information Visualization Design Studies and Techniques

Moving on to how *GraceFall* fits within data visualization research, the tool is primarily focused on the application of visualization techniques as opposed to the development of novel ones. With this in mind, the project is suitably described as a visualization design study and can be positioned relative to existing similar design projects of larger scope, namely KD-Box [21] and Vismon [3]. KD-Box is a method that addresses line cluttering of plotting many time-series by finding a representative line using density. It is useful as a comparative study because it addresses similar problems of visualizing time-series data from sensors where many independent data series need to be summarized or reduced to extract useful information about the collective. Vismon is a interactive visualization tool allowing fishery scientist to explore hypothesis about their models through sensitivity and constraint-based analysis. While Vismon does not consider time series plots, its treatment of model comparison across multiple views and incorporation of information uncertainty is comparable to the design components encountered further into the proposed project.

Additionally, specific design challenges within our project have been explored by previous works. Although our project allocates significant time for more detailed literature review, we have already looked at some initial solutions for overcoming anticipated design challenges:

1. Multiple time scale plots introduce challenges when visualization users are interested in both long-term summaries of temporal data and in the more detailed structure of the same data when considered over a short interval. Multiple techniques have been introduced to in addressing this problem. Many tackle this problem by showing a coarse to fine time scale approach. Time calendar [18] representation provides a month scale summary is provided by coloring the days on the calendar with a day-scale view on the side. Time-Series-Path [2] provide a squeezed colorized bar summarizing the the property of the time series with color while the detailed color representation is directly plotted for viewing. "Chronolensing" [20] allows interactivity enabling users to choose desired their time scale view.

2. There are much work on plotting many time series. We will focus on only ones that are relevant here. A general summary can be found in a survey paper by Ali et al. [1]. Many works attempt to directly plot the time series and reduce the cluttering afterwards. This class of methods include the use of density [12], grouping [10], and ordered plotting [17] to make the plots easier to view. However, by doing so, the visualization loses the fine

resolution of individual time series and, hence, difficult to detect anomalies from them. Other works that maps each time series to points 2D space [9] [16] can see the difference clearly. However, the dimension loses all semantic meaning.

3. Users of the visualization are likely to want to compare different tests or data sets in relation to one another, introducing a significant challenge to effectively lay out multiple test plots within a single view frame and navigate them. [3] can inform some initial design decisions for this project aspect.

4. Visualizing mathematical models is an interesting challenge as they don't fit as cleanly within standard visualization conceptual frameworks. The data object being shown is the output of an expression that is variable, making the data dynamic when dependent on other visualized data. A 2008 paper on plotting models based on parameter selections is considered as a useful starting point for this component of the project [11].

3 Data and Task Abstraction

A core problem reliability engineers encounter when analyzing a set of test results is a poor understanding of how observed component degradation depends on multiple induced stressors, especially as the models used to explain physical degradation are incomplete and rely on numerous physical assumptions. To understand how products degrade and fail as a result of stress or to evaluate how well a model captures complex physical phenomenon, engineers need to view the effects of wear-out processes as a function of time. Without existing visualization tools, engineers frequently need to compare multiple test results in the form of pure numerical tables, a notoriously difficult data form to analyze from. We introduce GraceFall to aid engineers in exploratory analysis of temporal information available in wear-out test/measured data, both simulated and real.

3.1 Data Set

Prior to dissecting the overarching visualization objective, it is necessary to first present the specific data available. There are three major data sets relevant to our visualization which we describe at an abstracted level.

1. Tabular data listing values associated with several attributes. This is long-form data with multi-attribute identification needed to determine the value "source", along with an independent time attribute. Specifically, each quantitative value is a measured quantity associated with some quantitative time instant and categorical sample source. A sample source is uniquely identified by the combination of four attributes: type, circuit number, chip number, and lot number. Each of these attributes are individually non-unique, and sample sources sharing the same chip number and/or lot number field values will have potentially correlated values. If the tabular data is separated by unique sample source, each separable set of values will represent a single data series of measured values at different instants in time. The quantity of sample sources, samples per source, and value ranges are fully unbounded, however will be restricted within the scope of the GraceFall design requirements.

2. Tabular data listing values associated with a unique time interval. Each row entry has an attribute for the quantitative value of each applied stressor along with three quantitative attributes for start time, end time, and duration of the unique time interval. A final categorical attribute indicates the name of the stress interval but is out of the scope of the GraceFall tool. The quantity of stressor attributes, stressed time intervals, and value ranges are once again unbounded but restricted within the scope of GraceFall.

3. Hypothesized mathematical models that can be fit to the measurement data. These are mathematical expressions attempting to explain the test results as fitted functions of the stress data. To keep the project in a manageable scope, this project will restrict the data to deterministic models as opposed to the eventual desire to additionally support probabilistic ones. As this data type does not fit into the framework introduced in the CPSC 547 course, the visualized data for a fitted model will be simply represented as sets of quantitative points.

3.2 Task Specification

To better analyze the visualization design problem of GraceFall we use a generalized framing, and thus the expected user tasks mentioned in the introduction were abstracted to a more general data perspective, with the resulting tasks summarized in Table 1.

Due to time constraints, only tasks **T1** and **T2** were addressed in the initial version of *GraceFall* completed in the scope of this project. Tasks **T3** and **T4** remain as valuable items to address and will be tackled for the next release version of *GraceFall*.

4 Solution Design

Based on the abstracted user tasks, the developed requirements for GraceFall are shown in Table 2. Note that the requirements **R3** and **R4** associated with tasks **T3** and **T4** were not able to be addressed within the scope of the initial version of the tool and so are left to future work.

4.1 Design Technique Decision Process

To contextualize the following discussion, an early mock-up using selected design features in shown in Figure 5. First, to address **R1.c** in a way that mitigates the clutter risk discussed in challenge **C1**, two view panes will be used for a given data set: a group line plot and a PCA plot.

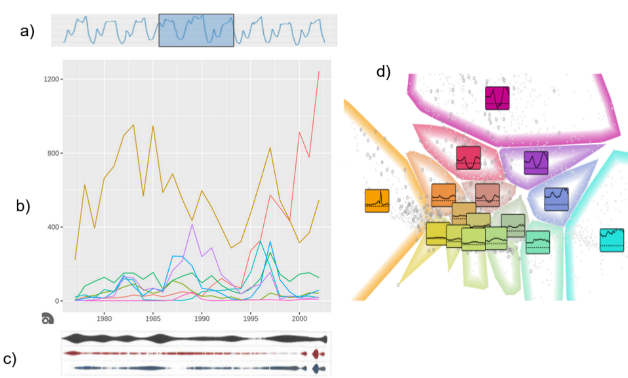

Fig. 5. Mockup of proposed visualization. All details can be found in section 4. **a)** The time summary plot for satisfying **R1.b**. **b)** The Group Line Plot. **c)** The three panes satisfying **R2**. **d)** the PCA Plot

Group Line Plot (GLP) The primary choice of visualizing the time-series will be standard direct line plots as required in **R1**. Other visualization techniques that conduct major data transform on the time-series data, such as those utilizing PCA will not work, because it is of critical importance for the user to see the detailed evolution of slope and aggregate behaviour as time progresses. To allow users to easily identify trends, k-means clustering will be used to meet **R1.e** and **R1.a**, and will partially overcome **C1**, cluster-colouring time series with similar properties as in [4, 7].

PCA Plot Clustering by itself is insufficient as direct line plot reduces the much needed resolution of the time series to find points of interest. To address this, PCA reduction will be used to map the time series down to a 2D point. Those points will be assigned colours as in the k-means approach. These points can better identify outliers as demonstrated by [16]. Further, when users pan over a region with the same color, a small panel will appear above the region showing the mean time series derived from K-means to give the user a sense of the underlying data. Additionally, within this view, the user can select and

Table 1. Abstracted task definitions for GraceFall visualization tool.

Task ID	Priority	Task Definition
T1	Mandatory	Examine the statistical distribution evolution of multiple data series as time progresses
T1.a	Important	Find identifiable groups that show similar behaviour within the larger set
T1.b	Mandatory	Roughly track sample variance and likelihood region evolution, identify asymmetries
T1.c	Important	Catch outlier samples and propose probable causes for the outliers using background/meta data
T2	Mandatory	Search for probable values in background/meta data that result in large changes to data series trends in time
T3	Important	Evaluate the success/performance of proposed models in explaining the observed data series
T4	Nice to Have	Compare multiple different sets of background/meta data and their corresponding sets of data series in terms of their influence on the outcomes of the previous three tasks

Table 2. GraceFall design requirements.

ID	Priority	Requirement Definition
R1	Mandatory	Tool must display multiple data series against time
R1.a	Important	Up to five sets of data series can be visualized on the same axes
R1.b	Mandatory	Can change focus between multiple time scales of interest on demand
R1.c	Mandatory	Up to 1000 data series must be individually viewable simultaneously
R1.d	Nice to Have	Aggregate properties of sets of data series can be overlaid or substituted for raw data on demand
R1.e	Nice to Have	Provide automated classification of data series into groupings by similarity
R2	Mandatory	Background/meta data is displayed alongside data series to maximize spatial "closeness" / minimize cognitive effort required to jump between viewing data series and meta data values
R2.a	Mandatory	Up to three meta data fields must be displayable simultaneously
R3	Important	Proposal model outputs, dependent on background/meta data values, are displayable on the data series axes on demand
R3.a	Important	Must be immediately visually distinguishable from displayed data series or statistical properties of the data series
R4	Nice to Have	Up to four data series axes views can be displayed simultaneously, arranged to maximize spatial ”closeness” / minimize cognitive effort required to jump between viewing different axes
R4.a	Nice to Have	Significant differences in background/meta data values between views can be highlighted on demand

filter only the time series they are of interest to show on the group line plot, allowing them to explore the data they are only interested in.

To address **R1.b**, a way of navigating multiple time scale is required. Although multiple techniques in the literature allow for this functionality, lensing similar to that in [20] was selected for one key reason. The other approaches mentioned in the Related Work section are discrete, limiting the scales at which the user can explore the data. For wear-out test data, the expected time scales of interest are not predefined and will have to be fine-tuned depending on the test conducted and user task. Allowing for continuously-defined time scales to be selected is thus valuable to prioritize. To ease lensing navigation, a squeeze overview of the full time interval will be provided. The user can select an area that they wish to explore. The selected sequence will appear in the Group Line Plot for exploration.

For requirement **R2**, the user needs to see stress data values with ease to identify causes of trends in the primary data series. To satisfy **R2**, a three long panes similar to [8] with small balls corresponding to the magnitude of stress parameters will be shown on the bottom of the Group Line Plot. This allows the user to compare a maximum of three stress parameters at a time, satisfying **R2.a**. Compared to alternative channels such as color, the eye can see the magnitude difference between size of balls with improved accuracy and precision, especially as the vertical positions are aligned, effectively allowing for length comparisons. This approach was compared to directly showing bar charts when hovering over the points similar to the box plots in [2], however that approach introduced occlusion and expected comparison difficulties when displaying the values of three stressors simultaneously.

4.2 Implementation

For the implementation of the design solution, it was desirable to have sample data sets for verification and validation. The data sets used in this project are generated through the existing wear-out reliability simulator and attempt to encapsulate a variety of different data sets with interesting features (e.g., subgroupings within the data, unique stress tests, different and/or multiple types of degraded values). In total, three data sets were constructed: **(1)** a ramp voltage test producing degradations in amplifier gain for highlighting simulator performance on large quantities of data series, **(2)** a riverbed erosion data set for highlight stochastic variation identification across data series groupings, and **(3)** a resistance degradation test with three separate resistor types to capture a maximally diverse range of degradation types and behaviours. Examples of the completed *GraceFall* tool running on datasets 2 and 3 are shown in Figures 6 and 7 respectively.

For the software implementation of *GraceFall* the project was constrained to a Python-based toolset for ease of compatibility with the existing wear-out reliability simulator and based on our prior familiarity with the language. Five interactive visualization libraries/frameworks were considered: Plotly, Bokeh, Mpld3, Gleam, and Vega-Altair. After feature comparison among other criteria, Vega-Altair was selected for implementing *GraceFall*. Although Plotly offered slightly more customized design and additional features, it was determined to be much more challenging to use, and Vega-Altair was a close second in terms of feature-rich visualization design. An additional benefit of using Vega-Altair is that it follows the Vega-Lite specification which follows an extremely similar design space specification to that presented in the CPSC 547 course, providing a valuable opportunity to work with a tool that helps reinforce the learned concepts from classes.

With development data and the visualization framework available, we organized the structure of our program into three main components. These include **(1)** creating Altair selectors that supports interactivity across views, **(2)** generating the main views, such as the time series and the PCA plot, and **3)** combining them all together. Selectors include single and multiple selection. The selectors are combined with Altair conditioning to filter the data to be visualized chosen by the user. The combination of these two enables diverse interactivity and responses across all views from a single interactive action performed by the user. Altair, unfortunately, does not offer manipulation in the colour luminance channel. As consequence, it limited our ability to implement

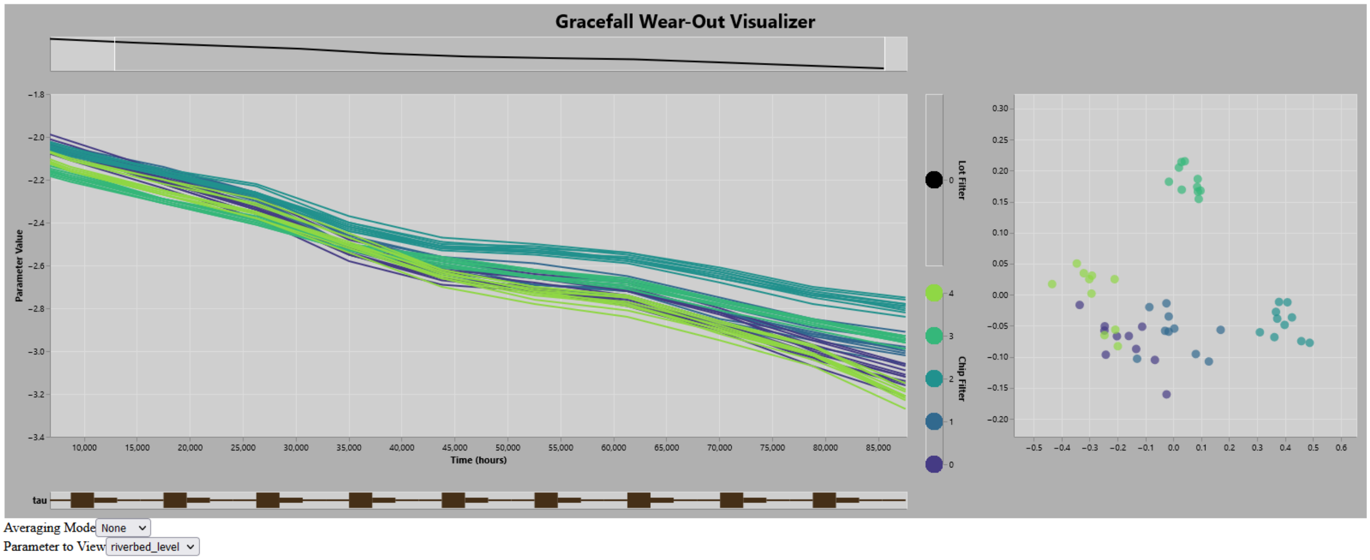

Fig. 6. Completed visualization tool displaying riverbed erosion data, with each "chip" group representing a single river and "lot" representing a set of rivers in a similar geographic environment.

a encoding that distinguishes the different semiconductor chips. Further, feature gaps forced us to give up on creating a nice area mark since they are unsupported in Altair for K-means. This is discussed further in section 5.1.

5 Results

This section describes the completed visualization tool as well as a possible use case walkthrough.

5.1 Resulting Visualization

Our resulting visualization closely follow the designs described in section 4. It consists of two main plots, a time-series plot (TSP) and a PCA plot, to visualize the time series data. To support, panes below the TSP show the stress applied at each time interval for comparison within the TSP. A time summary plot above the TSP enables navigation in multiple time scale. We added two filters on the side for users to select time series of interests to reduce the cluttering. Different from what is described in section 4, K-means is not used to color the points in the PCA view. Also, the showing of the k-means time series when panning over a clustered region is not implemented due to limitations in Altair and will be discussed in Section 6.5. Instead, we colored them with respect to hierarchical groupings. To be specific, the hierarchical groupings are Lot and Chip where Lot refers to the batch of computer chips produced at the same time, and Chip refers to the location of the placement of the computer chip. Further, we included average operations over the time series at the global, Lot and Chip level. The operations are selectable in the drop down menu at the bottom of left of our visualization tool. These operations reduce clutter and allow users to have a quick summary of the time series. The change of K-means colouring scheme and implementation of the filters and average operations will be discussed in section 6.1.

5.2 Qualitative Validation

Our visualization tool can aid in finding outlier time series and the identifying the cause of the time series' abnormal behavior. We tested it ourselves. The PCA plot easily enable us to find a time series with abnormal change. Subsequently, from looking at the stress data panes, we identify the stress factor that it was causing the abnormal behavior.

5.3 Use Case: Investigating Cause of Degradation

An engineer is interested in investigating the experiment results of the stress applied to batches of computer chips and placed at different locations referring to different Lot and Chip respectively. After loading the data set into *GraceFall*, all the time series are on the TSP. Unfortunately, the large data set results in the view beingover cluttered. To get a broad sense of time series behaviour, the engineer looks at the PCA plot finding the time series are generally clustered together on a Lot level. Further investigation found that there is one batch of chips that are away from the other batches, a cluster of outliers. The engineer select the batch on the PCA plot and apply a Lot level average operation to get a sense of the behavior of the time series in that batch. The engineer cycles through the batches using the Lot filter to see what the other cluster of time series look like. The engineer found that the outlier cluster degrades severely between time 300 and 400. The engineer looks at the stress panes at the bottom of the TSP and found that the temperature is increasing rapidly at that period of time. The engineer revisit the hierarchical group filters and finds that those experiments belong to Chip 2. Subsequently, the engineer form a hypothesis that the placement of those chips cause the chips to be more susceptible to heat. This finding, then, guides the engineer to investigate in the real hardware, saving him much time in exploring the experiment data when *GraceFall* is unavailable.

5.4 Developer Use

Another good indicator for the success of the developed visualization tool is its use by the developer Ian in pursuit of generating interesting data sets for his research. Using *GraceFall*, it was easy to identify behaviours resulting from simulation parameters and to iterate on the tests and device models to produce intended behaviours that could very quickly be analyzed in the tool. This rapid iterative process would not have been otherwise possible.

6 Discussion

This section will describe and motivate our design changes. Subsequently, we will discuss the strengths and limitations of *GraceFall* in its current state.

6.1 Design Choice Changes

6.1.1 PCA Point Coloring

In our original designs, the points on the PCA plot are to be colored by a class assigned by K-means and overlayed with an area mark encircling the class. During implementation, it was found that it was important for the user to identify when and where the chip is produced and placed. Hence, we use the color channel of the PCA points to represent the Lot and Chip hierarchical levels. However, after such a change, the color of each point conflict with the color of our in-development area

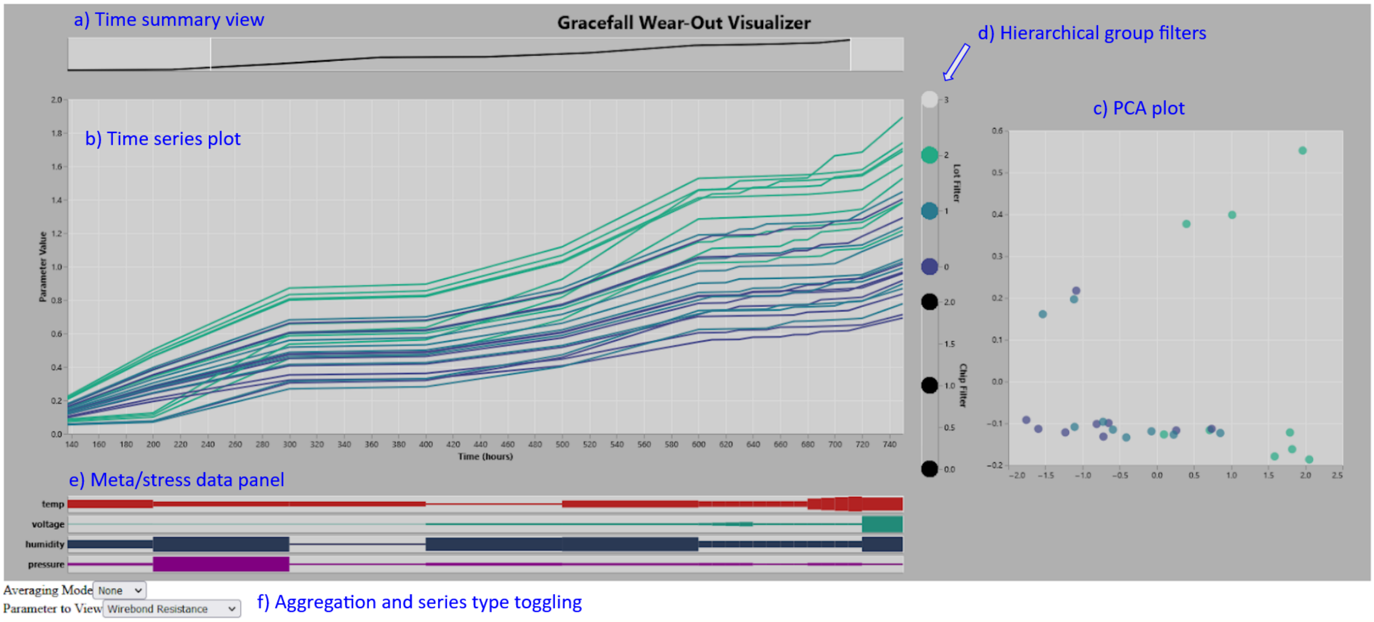

Fig. 7. Completed visualization tool displaying a data set of increasing wirebond resistance as corrosion affects the composing metal alloy. Annotations showing the different elements of the tool interface are shown.

mark labeled by K-means. The conflict overloads the color channel leading to ambiguity as to what attribute it represents (saturation and luminance channels were not available). To deal with this issue, the decision is made not to use K-means coloring. As a consequence of the color encoding change, the Group Line Plot will no longer be colored by K-means classes and will be colored by the hierarchical levels, and the Group Line Plot will just be a normal time-series plot (TSP).

6.1.2 Hierarchical Group Filters

With the K-means coloring strategy removed, the cluttering issue becomes unaddressed. We exploit the fact that users are interested in investigating one subset of time series at a time and introduce hierarchical group filters. Hierarchical Group Filters provide the option for the user to select the Lot and Chip they are interested in investigating by hiding the unselected time series. The hiding of unselected time series reduce clutter and enables the user to focus on the specific data series of interest and analyze them.

6.1.3 Aggregation panel

Even after hierarchical group filtering is applied, we found the number of time series on the TSP still present a problem of information overflow. Concretely, it is difficult to parse useful information when looking at a large number of time series. To address this issue, we introduce averaging operations that allos users to get a quick sense of the data present through reduced visual clutter via aggregation. The operation can be performed on a global scale where by the average operation is performed across all time series by averaging their values at each time stamp to obtain a new average time series. Similarly, such operation can be performed on a Chip and Lot level. The averaged time series will capture both the values and gradient of the selected subset. As a result, they provide a summary information of the selected time series.

6.2 Strengths

Our tool presents several advantages in visualizing a large number of time series through the use of PCA plot, hierarchical group filters, and averaging operations. Additionally our tool enable the ease of observing stress factors affecting the times series through data panes at the bottom of the TSP. The use of a PCA plot allows users to see the differences between time series easier. It is difficult to identify differences between time series directly observing the time series in the TSP. The PCA takes into account of the properties of each time series, namely their value and gradients, and project them onto a 2d plane. This projection will serve as summery of the entire time series. Comparing time series in a 2d representation is much easier as one does not need to memorize the shape of each time series to conduct a comparison. Since PCA takes into many properties of a time series, time series that are similar are close together while those that are different are far away. Hence, it is also much easier to identify outliers in the PCA plot. In our application, it is easier to identify time series that behave differently from others under different stress factors.

Additionally, the tool addresses the time series cluttering issue through the use of of hierarchical group filtering and average operations. Specifically, hierarchical group filtering will remove time series that are not of interest causing cluttering to reduce. Though the number of time series have reduced, deriving insights can still be difficult if many time series remain. Our averaging operations can address this by providing a summary of the remaining time series which gives the user an idea of their overall performance as well as properties to look for when searching for outliers.

GraceFall can also assist in analyzing relationship between the time series and their underlying stress factors through the stress factor data panel. By representing the value of the each stress as a rectangle between each time period, users can easily discern the different period which the stress are applied as well as their magnitude. The direct placement under the TSP allows quick comparison between the stress factors and the time series, reducing the amount of attention required for analysis.

6.3 Limitations

Though much effort has been put into resolving the time series clutter issue, it is not fully resolved in our tool. This problem is apparent when comparing many different time series. Multiple time series still overlay on top of each other despite our efforts. The overlaying obstructs requires the user to first parse a clean version of their target time series, as consequence, inducing mental effort during time series comparison and in selecting the filtering or aggregation techniques to apply. Another challenge is that the current solution to supports the exploration of the data sets in various time scales will fail in extreme cases where a single data set incorporates time ranges in both large magnitudes and sub-second stress phases.

Regarding the meta/stress data, though having the stress data panel under the TSP eases comparison, the problem is not fully solved. For time series that are on the top of the TSP, it is difficult to identify where is the exact location at which the stress is being applied. This difficulty translates to difficulty in conducting comparison between time series.

Finally, the PCA plot introduces a new problem that is not present originally. The dimensions resulted from performing dimension reduction are not interpretable. The user will have no idea what the underlying time series look like simply from observing the point without referring back to the TSP. The only information the PCA plot communicates is that some time series are similar and different and that there are trends in the data set. It does not communicate any information about what those trends might be.

7 Future Work

Over the course of this project, we have made great strides in addressing tasks **T1**, **T2**. However, due to time limitation, tasks **T3** and **T4** were not addressed, focusing on the visualization of models and multiple tests. Additionally, some of the limitations of the developed solution described in the previous section can be improve with further efforts, namely the stress data comparison issue and the lack of interpretability in the PCA plot. After addressing some of the current limitations, model visualization will be prioritized over multiple test visualization based on our use case prioritization. The key enhancements scoped for the second version of the tool are detailed in the following subsections.

7.1 Stress Meta Data Vertical Time Markers

The ambiguity of when exactly a stress is applied can be mitigated by adding additional markers. A obvious marker to add is to add vertical lines at locations where the change of stress value happens. The vertical lines will clearly crossover with the time series. This solution is imperfect as it conflicts with the color channel of the time series and does not address that there are many stress factors. As such, more thoughts are required.

7.2 Interpretability of PCA

The lack of interpretability of the PCA plot can be alleviated in two ways. The first is to implement a pop up of the mean time series when a mouse hovers over a set of points in the PCA plot. The mean time series can be obtained by averaging the time series of the points within a circle with some radius. This will give a sense of what the time series the point on the PCA plot might represent. Second, a small summary window should be provided providing the top component of the data series contributing to the resulting axis during dimension reduction. The weights of the components can be easily obtainable in the weights of the PCA algorithm. These weights can inform the user what the PCA algorithm deemed most important when constructing the 2d points.

7.3 Model Visualization

Model visualization is an unexplored task we in this project due to limited time. This task is critical to be pursued further due to its immense application. Having a model visualization will allow users to quickly form a hypothesis and test the hypothesis against the recorded data. This allows for iterative hypothesis formulation and aid further experiment designs.

8 Lessons Learned

Over the course of this project we were particularly excited to apply some of the techniques learned through the Visual Analysis & Design text. One of the reasons we selected the Altair for Vega-Lite library was due to its close correspondence with the marks and channels design framework introduced in the course. This selection helped us gain a firmer grasp of the course terminology, along with an intuitive understanding of the type of data that each marks and channels can represent. It was also a joy to discover the many other methods in the visualization literature. The use of PCA plots for summary was a particularly enjoyable avenue to explore.

In the early stages of the implementation process, we were impressed with the convenience that the visual grammar Vega-Lite provided [15]. After some time, however, we realized that no tool can act as a panacea. Vega-Lite lacked many features required by our designed solution and resulted in design changes and limitations. Some examples include the lack of dedicated saturation and luminance channels, or the inability to have text or views change dynamically based on interactions with the visualization. Perhaps we could have selected a more versatile visualization library, however our lack of experience made this selection process challenging and we do not feel that our decision was poor within the context of our knowledge at the time.

Another lesson we learned, arguably the most important one, is the process of design. Early on we stumbled into the blunder of trying to a nail for the chosen hammer. After finding interesting visualization techniques there was incentive to implement them without a proper analysis of the tasks a user will perform, and the initial resulting design had a slight mismatch with our targeted tasks as a result. We believe that our final design, incorporating proper task and data set analysis, is a much better match for the anticipated use cases. It is immensely valuable to see the quality of the different products that resulted from two different design mindsets.

9 Conclusions

The developed visualization tool *GraceFall* successfully tackles a challenging visualization problem. Capable of cleanly displaying highly varied scalar data series and providing valuable tools to investigate both individual and group behaviours, the tool enables reliability engineers to analyze their simulated and physical test results for anomalies, complex physical influences, and predicted wear-out reliability. Although many features had to be relegated to future versions, the initial release implements all the core functionality required for a successful minimum viable product. Ongoing work to iterate on the design will result in further improvements, with *GraceFall* hopefully providing invaluable aid in tackling the complexities of modern integrated circuit reliability challenges.

10 Milestones

With the conclusion of the course project, a summary of the expected and actual number of hours invested in each project task are shown in Table 3. Overall timeline conformance was around anticipated, with minor slippage on a few tasks due to illness and other course responsibilities, however no major issues were encountered.

References

[1] M. Ali, A. Alqahtani, M. W. Jones, and X. Xie. Clustering and classification for time series data in visual analytics: A survey. *IEEE Access*, 7:181314–181338, 2019.

[2] J. Bernard, N. Wilhelm, M. Scherer, T. May, and T. Schreck. Timeseriespaths : Projection-based explorative analysis of multivariate time series data. In *Journal of WSCG*, pp. 97–106, 2012.

[3] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum*, 31(3pt3):1235–1244, 2012. doi: 10.1111/j.1467-8659.2012.03116.x

[4] T. Fujiwara, J. K. Li, M. Mubarak, C. Ross, C. D. Carothers, R. B. Ross, and K.-L. Ma. A visual analytics system for optimizing the performance of large-scale networks in supercomputing systems. *Visual Informatics*, 2(1):98–110, 2018.

[5] I. Hill, P. Chanawala, R. Singh, S. A. Sheikholeslam, and A. Ivanov. Cmos reliability from past to future: A survey of requirements, trends, and prediction methods. *IEEE Transactions on Device and Materials Reliability*, 22(1):1–18, 2022. doi: 10.1109/TDMR.2021.3131345

[6] I. Hill and M. Tang. Gracefall. GitHub, 2022. Available: https://github.com/ianrmhill/gracefall.

[7] S. P. Kesavan, T. Fujiwara, J. K. Li, C. Ross, M. Mubarak, C. D. Carothers, R. B. Ross, and K.-L. Ma. A visual analytics framework for reviewing streaming performance data. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 206–215. IEEE, 2020.

[8] M. Krstajic, E. Bertini, and D. Keim. Cloudlines: Compact display of event episodes in multiple time-series. *IEEE transactions on visualization and computer graphics*, 17(12):2432–2439, 2011.

[9] J. Li, K. Zhang, and Z.-P. Meng. Vismate: Interactive visual analysis of station-based observation data on climate changes. In *2014 IEEE*

Table 3. Time allocations by project sub-task.

Date	Tasks	Expected Hours	Matthew Hours Actual	Ian Hours Actual
W1 (Oct 26 - 30)	Similar tools literature review	8	4	4
	Proposal	8	5	5
W2 (Oct 31 - Nov 6)	Relevant techniques literature review	8	7	3
W3 (Nov 7 - Nov 13)	Time series view design	3	3	1
	Multiple time scales design	4	4	1
W4 (Nov 14 - 20)	Additional literature review from feedback	4	4	2
	Project update report	8	4	4
W5 (Nov 21 - 28)	PCA plot design	6	7	0
	Test data series development	3	0	4
	Stress data pane design	3	3	1
	Boilerplate code base implementation	6	0	6
W6 (Nov 28 - Dec 4)	Time series view implementation	10	0	8
	Time summary view implementation	4	0	4
	Stress summary panes	6	0	4
	Hierarchical filtering implementation	3	0	7
W7 (Dec 5 - Dec 11)	Multiple view interactions	6	6	4
	PCA plot implementation	10	16	3
	Aggregation functionality implementation	4	0	6
	Visual design implementation	3	0	4
W8 (Dec 12 - Dec 16)	Series highlighting and switching implementation	4	5	3
	Prepare presentation	3	6	3
	Final report update	15	14	5
		Totals	88	82

Conference on Visual Analytics Science and Technology (VAST), pp. 133–142. IEEE, 2014.

[10] L. Liu and R. Vuillemot. Groupset: A set-based technique to explore time-varying data. In *Computer graphics Forum (Proc. Eurovis), 2022*, 2022.

[11] K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual steering - rapid visual prototyping of a common rail injection system. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1699–1706, 2008. doi: 10.1109/TVCG.2008.145

[12] D. Moritz and D. Fisher. Visualizing a million time series with the density line chart. *arXiv preprint arXiv:1808.06019*, 2018.

[13] S. W. Pae, H. C. Sagong, C. Liu, M. J. Jin, Y. H. Kim, S. J. Choo, J. J. Kim, H. J. Kim, S. Y. Yoon, H. W. Nam, H. W. Shim, S. M. Park, J. K. Park, S. C. Shin, and J. W. Park. Considering physical mechanisms and geometry dependencies in 14nm finfet circuit aging and product validations. In *2015 IEEE International Electron Devices Meeting (IEDM)*, pp. 20.6.1–20.6.4, 2015. doi: 10.1109/IEDM.2015.7409744

[14] G. Park, H. Yu, M. Kim, and C. H. Kim. An All BTI (N-PBTI, N-NBTI, P-PBTI, P-NBTI) Odometer based on a Dual Power Rail Ring Oscillator Array. In *2021 IEEE International Reliability Physics Symposium (IRPS)*, pp. 1–5. IEEE, Monterey, CA, USA, Mar. 2021. doi: 10.1109/IRPS46558.2021.9405181

[15] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics*, 23(1):341–350, 2016.

[16] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer graphics forum*, vol. 33, pp. 401–410. Wiley Online Library, 2014.

[17] T. Trautner and S. Bruckner. Line weaver: Importance-driven order enhanced rendering of dense line charts. In *Computer Graphics Forum*, vol. 40-3, pp. 399–410. Wiley Online Library, 2021.

[18] J. J. Van Wijk and E. R. Van Selow. Cluster and calendar based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, pp. 4–9. IEEE, 1999.

[19] C. Yang and F. Feng. Multi-Step-Ahead Prediction for a CMOS Low Noise Amplifier Aging Due to NBTI and HCI Using Neural Networks. *J Electron Test*, 35(6):797–808, Dec. 2019. doi: 10.1007/s10836-019-05843-7

[20] 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.

[21] Y. Zhao, Y. Wang, J. Zhang, C.-W. Fu, M. Xu, and D. Moritz. KD-Box: Line-segment-based KD-tree for interactive exploration of large-scale time-series data. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):890–900, 2022. doi: 10.1109/TVCG.2021.3114865