ABSTRACT

Title of thesis: Shape Identification And Ranking
In Temporal Data Sets

Machon Gregory, Master of Science, 2009

Thesis directed by: Professor Ben Shneiderman
Department of Computer Science

Shapes are a concise way to describe temporal variable behaviors. Some commonly used shapes are spikes, sinks, rises, and drops. A spike describes a set of variable values that rapidly increase, then immediately rapidly decrease. The variable may be the value of a stock or a person's blood sugar levels. Shapes abstractly describe a variable's behavior. Details such as the height of a spike or its rate increase, are lost in the abstraction. These hidden details make it difficult to define shapes and compare one instance to another. For example, what attributes can be used to define a spike's behavior? And what attributes of a spike determine its "spikiness"? The ability to define and compare shapes is important because it allows shapes to be identified and ranked, according to an attribute of interest. A lot of work has been done in the area of shape identification through pattern matching and other data mining techniques, but ideas combining the identification and comparison of shapes have received less attention.

This dissertation fills the gap by presenting a set of shapes and their attributes, by which they can be identified, compared, and ranked. Neither the set of shapes,

nor their attributes presented in this dissertation are exhaustive, but it provides an example of how a shape's attributes can be used for identification and comparison. Spikes, sinks, rises, drops, lines, plateaus, valleys, and gaps are the shapes presented in this dissertation. Several attributes for each shape are identified and defined. These attributes will be the basis for constructing definitions that identify a particular behavior of a shape and allow it to be ranked.

The second contribution of this work is an information visualization tool, TimeSearcher: Shape Search Edition (SSE), which allows users to explore data sets using the identification and ranking ideas, presented in this dissertation. Case studies were performed to evaluate the benefit of shape identification and ranking in different data sets. Four case studies were performed with a single user, exploring network traffic data and X-ray diffraction data.

Shape Identificaiton And Ranking In Temporal Data Sets

by

Machon Gregory

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2009

Advisory Committee:
Professor Ben Shneiderman, Chair/Advisor
Associate Professor Ben Bederson
Professor Dana Nau
Dr Catherine Plaisant

Dedication

To Renee

Acknowledgments

First, I would like to thank my Lord and Savior Jesus Christ for providing me with all the things that I needed to complete this thesis. I would like to take a moment to thank the individuals that He placed in my life that helped me with this thesis.

Dr. Shneiderman has been a patient, enthusiastic, and motivating adviser. I can not put into words how much I appreciate all he has done. He has been an excellent guide as I traveled through graduate school with this thesis concluding the journey. It has been a pleasure to have you as an adviser, teacher, and mentor. I will never forget the things I learned under your tutelage.

This work began as a class project and without that base I would have been lost. I would like to extend my thanks to my group members Anthony Don, Elena Zheleva, and Sureyya Tarkan. TimeSearcher SSE was built upon Harry Hochheiser TimeSearcher tool and I thank him for providing me with a codebase that I could work with.

Finally, I would like to thank my family. Mom, Dad and Mike, thank you for all your encouragement throughout this process. And to Renee, thank you for all your support, you did everything I could not do during this entire process. You are the perfect example of what a wife should be. I don't know what I would have done without you.

Table of Contents

List of Figures

Chapter 1

Introduction

Shapes are a succinct way of describing the behavior of a temporal variable. For instance, a spike describes a sharp increase followed by a sharp decrease. A shape describes a behavior abstractly. Therefore, the rate at which a spike increases or the height of the peak, as well as other details about the variable's behavior is lost. The absence of these details makes it difficult to compare one shape to another. For example, given a spike, how can it be described or compared to another spike? A lot of work has been done identifying a particular shape in a specific data set, but little work has been done to examine individual shapes and generalize their use. This thesis focuses on the shapes that are created when a single variable is plotted over time. Some of the shapes that may be created are spikes, sinks, lines, rises, and drops.

Shapes such as spikes, drops and increasing lines are used by professionals in many different fields to describe the behavior of temporal variables. Doctors look for a spike in blood pressure as a sign of a panic attack or a much worse condition. Stock market analysts use shapes to describe changes in stock prices. For instance, a drop in prices may indicate a bad day for the market. On the other hand, stocks steadily increasing may indicate a time of prosperity. Published research results offer concrete evidence of the usefulness of shape identification. For example, spikes were used by Balog et al. to understand the mood of bloggers in relation to world events[3] and by Dettki and Erisson to analyze the seasonal migration patterns of moose[9]. These shapes are obvious in a visual representation to the informed observer, but they are often hard to describe precisely and compare to other shapes of the same type. When comparing drops in stock prices or spikes in electrocardiograms it is hard to compare the individual shapes. The ability to identify and rank shapes of interest in a visualization of temporal data sets can be helpful in analysis and knowledge discovery.

This thesis describes a set of commonly used shapes, listed below.

- Spike – a significant increase in value followed by a significant decrease in value in a set of sequential points
- Sink – a significant decrease in value followed by a significant increase in value in a set of sequential points
- Line – a set of sequential points with the same general behavior
- Rise – a sustained increase in value in a set of sequential points
- Drop – a sustained decrease in value in a set of sequential points
- Plateau – a temporary increase in value in a set of sequential points
- Valley – a temporary decrease in value in a set of sequential points
- Gap – a specific type of valley where the values temporarily decrease to zero

Shape Attributes	
Spikes and Sinks	absolute height relative height angular height number of points
Lines	slope length volatility average value
Rises and Drops	magnitude of change length average value
Plateaus, Valleys, and Gaps	magnitude of change length average value

Table 1.1: This tables contains the set of shapes and their attributes, which are described in this thesis. The attributes associated with spikes and sinks are the absolute, relative, and angular height, as well as the number of points in the shape. Absolute height is the value of the peak point, the point at which the increasing edge meets the decreasing edge. Angular height is the value of the angle created at the peak point. Relative height is the value of the peak point relative to the other values in a time series. The number of points is a count of the points in the spike or sink. Slope, length, volatility, and average value are the attributes associated with line shapes. Slope is the rate of change and the length is the number of values in the shape. Volatility is a measure of the variability of the values in the line. Average value is an average of the values in the line. Rises, drops, plateaus,

Each shape will be assessed by a set of measurable attributes described in Table 1.1. For example, a line shape's primary attributes are its endpoints and slope. An attribute, such as the "spikiness" of a spike, may be manifested as one or more measurements of the shape's attributes. Each measurement or set of measurements represents a different behavior. The attributes are used to define a shape's behavior and compare and rank the shapes. A shape definition consists of one or more constrained attributes. For instance, a line with the slope constrained to be positive defines an increasing line. A shape can have many definitions that identify different behaviors of interest. A ranking metric is one or more attributes by which a shape is compared to other shapes of the same definition and ranked. A ranking metric results from one or more calculations performed over values associated with a particular variable. The shapes that will be discussed are not an exhaustive set of shapes, nor are the attributes. This thesis presents the idea of identifying behaviors of interest through shape identification, then ranking the shapes according to a set of attributes.

The shapes and attributes that will be discussed are simple, as are the measurements of the attributes. This work is not a replacement for pattern mining techniques used to identify a unique behavior in a data set. However, the work presents a way of thinking about an identified behavior of interest and how it is defined and can be compared to other behaviors.

This work focuses on the analysis of temporal data sets, which are a collection of items with value readings over a period of time. The value readings are taken at time points, which are equally spaced and consistent across all items in the data set.

An item and its complete set of values is a time series. For example, a collection of patients' EKG data is a temporal data set. If a reading was taken every thirty milliseconds, at each thirty millisecond interval a time point exists. The set of value readings from a single patient is a time series. The shapes are depicted in a time series and all analysis will be performed on the time series data.

A subset of the shapes and their multiple definitions were incorporated into TimeSearcher Shape Search Edition (SSE), an information visualization tool. SSE is built upon TimeSearcher 1[19], and allows for the exploration of temporal data sets by identifying shapes of interest and ranking them according to a ranking metric. SSE visualizes shapes and provides a numerical ranking metric, which allows the shapes to be compared. SSE can identify shapes like increasing, decreasing, and volatile lines, as well as spikes, sinks, rises, and drops. SSE has several definitions for each of the shapes to identify different types of behaviors.

Chapter 2

Literature Review

There is a substantial amount of research in the area of shape identification. The majority of the research falls into the area of pattern identification or pattern discovery. The sequence of values that make up a pattern define a shape, so for the purposes of this thesis shapes and patterns are the same. The primary difference between a pattern and a shape is a pattern normally suggests repetition, while a shape does not. However, most research in pattern identification identifies some sequence of values, a shape, and attempts to find similar shapes. Shape identification provides a foundation for the ideas presented in this thesis. Some of the research, such as Agrawal et al.'s shape definition language (SDL)[1] and Hochheiser, et al.'s timeboxes[18], focuses on allowing users to define shapes of interest and then identify them in a data set. Research in the area of pattern discovery has focused less on the definition of the pattern and more on the value of the identified pattern. Many of the papers on pattern discovery start to answer the question "How significant or interesting is the identified pattern?" Much of the work in this area takes an automated approach, examining sets of values in a data set and determining their value based on some function. The idea that patterns can be evaluated to estimate their value to the user is one of the ideas presented in this thesis. The following section will provide an overview of the current ideas about pattern identification and pattern evaluation and explain how this thesis uses and extends them.

2.1 Shape Definition

Some of the research in pattern identification has supplied users with an expressive efficient method of defining patterns, while others have focused on optimizing the definition to translate into an efficient data query. An expressive and efficient method of defining a pattern provides an equal balance between the complexity of the definition and the granularity of the pattern it can identify. An efficient data query is a query that can "quickly" identify value sequences that match a definition within a large data set. This thesis primarily focuses on providing an expressive language for identifying and comparing shapes, but the calculation of the attributes were kept simple to minimize the cost of identifying a defined shape. Agrawal et al.'s and Hochheiser et al.'s present two distinct methods of defining shapes. Both are expressive, but for different reasons. Agrawal et al.'s SDL provides a language consisting of an alphabet and a set of operators to define a shape; Hochheiser et al.'s research has focused on visual widgets to define shapes.

SDL provides a simple alphabet, Table 2.1, to describe point to point transitions in time series data. For example, the symbol "Up" may be used to define a significant increase in a stock price from one time point to the next. The definition for a symbol in the alphabet is

$$\text{(alphabet (symbol } \mathit{lb}\ \mathit{ub}\ \mathit{iv}\ \mathit{fv}\text{))}$$

Symbol	Description	*lb*	*ub*	*iv*	*fv*
up	slightly increasing transition	.05	.19	**anyvalue**	**anyvalue**
Up	highly increasing transition	.20	1.0	**anyvalue**	**anyvalue**
down	slightly decreasing transition	-.19	-.05	**anyvalue**	**anyvalue**
Down	highly decreasing transition	-1.0	-.20	**anyvalue**	**anyvalue**
appears	transition from a zero value to a non-zero value	0	1.0	zero	nonzero
disappears	transition from a non-zero to a zero value	-1.0	0	nonzero	zero
stable	the final value nearly equal to the initial value	-.04	.04	anyvalue	anyvalue
zero	both the initial and final values are zero	0	0	zero	zero

Table 2.1: This table contains a set of sample symbols, taken from *Querying shapes of histories*, defined using Agrawal et al. Shape Definition Languague (SDL). The symbols cover scaled variations between -1 and 1. This alphabet along with SDL's operators can describe simplified two dimensional shapes.

"Symbol" is the text label for the alphabet being defined. *lb* and *ub* are the upper and lower bounds for the variation allowed and *iv* and *fv* are the constraints placed on the initial and final values, respectively. The constraints can take the value of **anyvalue**, **nonzero**, or **zero**. In addition to the ability to define symbols in the alphabet, SDL provides a set of operators to describe the relationships between symbols. For example using the alphabet in Table 2.1 and the operators provided by SDL a spike could be defined as:

```
(shape spike(upcnt dncnt)
            (concat (exact upcnt (any up Up))
                    (exact dncnt (any down Down)))
```

This is a parametrized definition of a spike shape, *upcnt* and *dncnt* are the inputs.

The statement, (**exact** *upcnt*(**any** up Up)), says that exactly *upcnt* values must be defined as “up” or “Up” and the same is true for (**exact** *dncnt*(**any** down Down)). The **concat** operator indicates that the increasing points meeting the definition of the symbols “up” or “Up” must be followed by decreasing points meeting the definition of “down” or “Down”. SDL’s ability to define symbols combined with the provided operators gives users a lot of control over the expressiveness of the language. An alphabet could be defined to identify almost any shape. The “blurry matching,” bounded value ranges, instead of exact values, gives users the ability to identify a set of similar shapes. The expressiveness and “blurry matching” are desirable traits for any shape identification techniques. This method of defining shape places a large burden on the users to define a set of symbols that meet their needs.

TimeSearcher 1, an information visualization tool for exploring time series data, provides different techniques for defining shapes. The TimeSearcher tool uses timeboxes, Figure 2.1, and several other types of queries to allow users to visually define shapes. Timeboxes facilitate shape definition by allowing users to visually specify a range of values for the x and y coordinates of the data points within a shape. In addition to the timeboxes TimeSearcher 1 includes an angular query widget, Figure 2.1. The angular query widget allows users to define a range of slopes that are of interest. The timeboxes are a fairly course grain approach to defining shapes. However, the angular queries provide a much more granular approach. The ability to visually define and quickly identify the shape within a data set is valuable

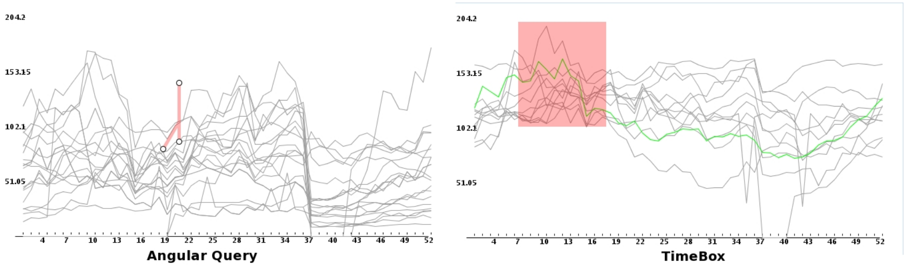


Figure 2.1: TimeSearcher 1 uses angular queries and timeboxes to graphically define shapes. The light red angular query widget, in the first image, can define shapes based on their slope. The vertical bar and the connected angled bar define a range of slope values. The white circles can be dragged to alter the value ranges. The time series that meet the defined shape are dynamically shown on the graph. The second image shows the timebox widget. The timebox widget defines shapes based on the a range of x and y values defined by the boundaries of the light red box.

in the knowledge discovery portion of data exploration, but the ability of the angular query to define shapes based on a measurable attribute is of more value to this thesis. The angular query is designed to only define slopes of interest, so it is limited in the complexity of the shapes it is able to identify, but it provides a foundation to build on. This thesis identifies several attributes like slope, which can be used to define shapes.

QueryLines[23] combines the point-to-point expressiveness of SDL and the dynamic visual query language of TimeSearcher 1. QueryLines is an information visualization tool that incorporates visual shape definition and user defined rankings to identify shapes of interest in temporal and ordered data sets. QueryLines defines three parameters, which are used to identify and rank shapes. These parameters are strength, penalty, and variability. Soft constraints and prefences make up the strength parameter. The soft constraints bound the x and y values with a set of line segments. The constraint is considered soft because a time series only has to meet one of the line segment constraints. The functionality provided by soft constraints are very similar to QuerySketch by Wattenberg[30]. Preferences are used to rank shapes. A preference consists of a set of contigous line segments that define a shape; identified shapes are ranked according to their similarity to the preference shape. The concept of evaluating identified shapes is important to this thesis and is discussed in Section 2.2. A penalty defines how to compare time points to a single line query. The penalties are minimum, maximum, goal, and trend, which identify points that are less than, greater than, exactly, and have the same slope as the query line. The final parameter variability is the user-defined dimensions in which a constraint can move. The variability parameters are fixed, y-flexible, x-flexible, and both-flexible.

SDL, TimeSearcher 1, and QueryLines enable users to define shapes of interest and locate their occurrences within a data set. SDL is an expressive solution that can be tailored to the needs of its users, but it could be hard to be used effectively by common users. On the other hand, TimeSearcher 1, is less expressive, but provides the users with the ability to define shapes in terms they understand (can see visually see). Keogh et al. extended timeboxes to create variable time timeboxes (VTT) to increase their expressiveness[19]. VTT allows a user to define a shape and then locate it over a range of values. Other research offers expressive ways of defining shapes over categorical data, such as temporal logic[21] and regular expressions[15], but the techniques do not easily transfer to temporal data sets. QueryLines has the expressiveness of SDL in a visual query tool, but it is unable to express higher level behaviors like anomalous spikes.

2.2 Shape Evaluation

In SDL and TimeSearcher 1, the significance of a shape is based strictly on whether the shape conforms to the definition or not. Although, the values used by the angular query widget could be used to define the significance of the identified shape, it is not an inherent capability of the tool. Since all shapes have the same significance they cannot be compared to one another. However, there is an area of research that is focused on evaluating the significance of a shape. A lot of this work falls into the area of pattern discovery[17, 16]. The ability to evaluate the significance of a shape implies that the identified shapes are comparable by some measurable attribute. For example, Dubinko et al.'s research in visualizing the evolution of social network tags defines "interestingness" as the likelihood of a tag occurring during a particular period of time[11]. "Interestingness," the frequency of a tags occurrence during a particular period of time provides a measurable attribute by which tags can be compared. Similarly, clustering techniques are used to identify patterns of interest. In this technique, similar patterns are grouped together into a cluster[14, 8]. Patterns identified using this technique can be compared based on the size of the cluster. The larger the cluster the more interesting the the pattern. Yang et al.'s STAMP algorithm uses statistics to measure the importance of identified patterns[31]. Each of these techniques provides a metric by which an identified pattern can be compared to another pattern. Unfortunately, these techniques are primarily associated with pattern discovery techniques and offer the user little control over what patterns are identified.

Garofalakis et al. recognized the “lack of user controlled focus in the pattern mining process” and introduced a set of algorithms called Sequential Pattern Mining with Regular Expression Constraints (SPIRIT)[15]. This research combines the ability to identify significance by using some measurable attributes, frequency, and regular expressions an expressive definition language. The regular expressions provide users with the ability to constrain the results returned by the pattern mining algorithm to just the patterns of interest to the users. The goals of this thesis are to provide capabilities similar to the SPIRIT algorithms, shape identification and ranking techniques using a user defined shape definition. Going beyond the SPIRIT algorithms this thesis presents techniques that allows users to define what is“interesting.” The majority of the research in the area of pattern discovery defines interesting as the frequency of the occurrence of a particular pattern. There are many novel techniques for identifying similar patterns, but few offer users the ability to direct the ranking of the results. The idea of ranking data according to user specified features is not new, Seo and Shneiderman’s created the rank-by-feature framework to assist users in selecting a feature that may interest them[24].

2.3 Summary

The research that has been done in the area of pattern definition and evaluation provides a foundation for the work presented in this thesis to build upon. SDL provides an expressive definition language that is able to accurately describe shapes given an appropriate alphabet. However, SDL's granular approach to defining an alphabet makes it difficult to express complex behaviors such as a data set with anomalous spikes. It also places a burden on the users to define the symbols. TimeSearcher 1 and 2 incorporate novel graphical query tools into an information visualization tool that allows users to explore time series data. It provides higher level tools with simple interfaces to do pattern definition, but it was designed to be a graphical exploration tool, and it is not as expressive as a language based solution. The research in shape evaluation has been focused on pattern discovery. The shapes are evaluated on what the perceived value will be to the user. Most of these techniques are not user guided and have inflexible evaluation functions. The shape definition ideas presented in this thesis balance expressiveness and complexity. The shapes identified can be ranked according to a user defined attribute.

Chapter 3

Shape Definitions

There are an infinite number of shapes; many of them are too complex to describe succinctly or create mathematical definitions to describe them. However, there are a set of simple shapes that are commonly used to describe a particular behavior. In the following sections several shapes will be described, as well as their attributes. These attributes will be used to provide examples of shape definitions and ranking metrics. Additionally, examples explaining how the shapes, their definitions and ranking metric may be used to answer different types of queries will be given. Line, spike, sink, rise, drop, plateau, valley and gap shapes will be discussed.

3.1 Line Shapes

The simplest shape, a line, is defined as one or more line segments created by a set of contiguous time points. In a 2D Cartesian plane, a geometric line can be defined using the equation, $y = mx + b$, where m is the slope, b is the y-intercept, and x is an independent variable. A line segment is a portion of a line defined by its endpoints. Line shapes are interesting because they can be used to describe any other shape, but they are most useful in describing consistent behaviors such as generally increasing, decreasing, stable, or volatile periods. For instance, a stock that consistently rises over a period of time can be described by an increasing line shape.

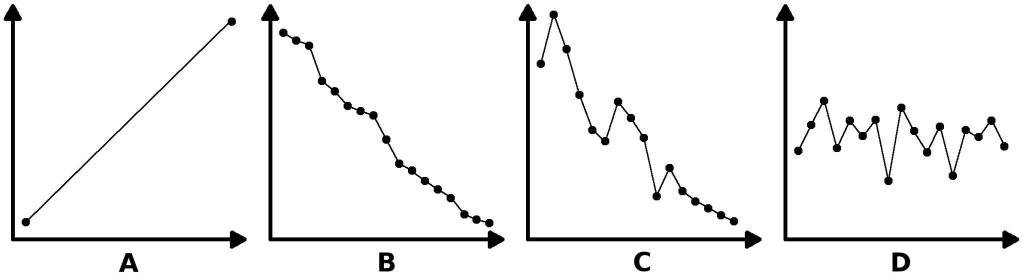


Figure 3.1: Graphs A through D show examples of line shapes. A shows a 2-point increasing line and B a multi-point constantly decreasing line. C is an example of a multi-point decreasing line that could be identified by a linear regression calculated using the values that compose it. The last graph, D, is an example of a volatile line, where volatility is a measure of the standard deviation of the values in the line.

A dieting person's weight can be described as a decreasing line shape. Depending on how its attributes are constrained, a line shape can be used to generalize the behavior of a set of time points or identify a specific behavior that is characterized by a limited range of value changes between time points. For example, a linear regression identifies a relationship between a set of variables that generalizes their behavior, but calculating the slope of each individual line segment can identify a specific behavior.

The attributes associated with line shapes are the length, slope, and volatility. The length attribute is the number of time points in the shape. The slope attribute is a measure of the rate at which the line shape is changing. The slope definition varies depending on whether the goal is to identify a particular behavior in the time series or to generalize the behavior of a set of time points. To identify a specific behavior, slope can be defined as the change in value between two time points.

This definition is identical to the definition of slope for a geometric line. Using this definition of slope any constraint applied to the slope must be consistent across every line segment in the line shape. For example, if one line segment is increasing, all line segments in the line shape must be increasing. On the other hand, if the goal is to generalize the behavior of a set of time points, the slope definition should consider all of the points together. For example, the slope of a line shape may be defined as:

- the amount of change between two time points that may or may not be contiguous
- the sum of the change of between all contiguous time points in the line shape
- the geometric slope of a linear regression computed over the time points in the shape.

These are examples of ways of calculating slopes. Figure 3.1C shows a line that could be identified using a linear regression, the set of values in the line have a decreasing trend. Each of these definitions describes a different behavior that may be of interest. Using different definitions for slope will result in different slope calculations for line shapes, therefore identifying different behaviors.

Volatility can refer to the relative rate at which a stock increases and decreases. A similar definition will be used to describe the volatility attribute of a line shape. The standard deviation of the values within a line shape can be used as a measure of a line's volatility. Figure 3.1D is an example of a volatile line. Other calculations may be more appropriate for measuring the volatility of line shape depending on the behavior of interest.

The slope, length, and volatility are attributes by which line shapes can be defined and ranked. Constraining the slope of a line shape to be a positive or negative value creates two definitions of line shapes, increasing and decreasing, respectively. According to the slope definition, an increasing line shape will characterize different behaviors. Constraining each individual line segment in a line shape to be negative creates a monotonically decreasing line, like the line shape in Figure 3.1B. Using the monotonically increasing line shape and ranking them according to their length, the question "Which stock has the longest period of constant growth?" could be answered. Understanding line shapes and a small set of attributes can be useful in answering such queries. In addition to constraining the slope of the line, the number of time points can also be constrained. The two point and multiple point lines are examples of definitions that are created by constraining the length attribute. Figure 3.1A is an example of a 2-point line shape; Figures 3.1B, 3.1C and 3.1D are examples of multiple point line shapes.

3.2 Spike and Sink Shapes

Spikes and sinks describe a temporal behavior in which a variable has a significant change over a period of time in one direction and then a significant change in the opposite direction. The point at which this change in direction occurs is the peak point. A spike, specifically, is a significant increase followed by a significant

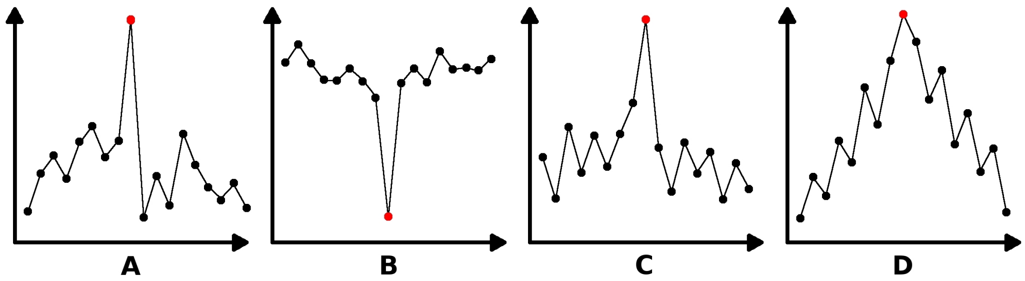


Figure 3.2: These graphs are examples of spike and sink shapes. The red dots are the peak points. Graph A, B, and C are graphs that may be ranked high based on its relative or angular height. The relative height is a measure of the difference between the peak point and average value of the remainder of the points. The angular height is the measure of the angle created by the two edges that meet at the peak point. An edge may consist of one or more points. Graph D is a spike that could be identified using a linear regression calculated over the points in the edges to the right and left of the peak point.

decrease. A sink is just the opposite, a decrease followed by an increase. These general definitions describe the behavior of spike and sink shapes. They are used in a diverse set of fields. For example, a stock market analyst may say a stock price spikes when a stock is rapidly bought for a period of time and then rapidly sold for a period of time. Similarly, a doctor would say when blood pressure spikes there is a rapid rise then fall in pressure. In order for an analyst or doctor to find a particular behavior, more detail must be added by identifying values of interest for a set of the attributes of spike and sink shapes.

The attributes associated with spike and sink shapes are the significance of the increase or decrease and their duration. The significance can be manifested in one or more attributes. The significance of the change can be measured by the absolute, relative, or angular height of the peak point. Absolute height is the absolute value of the peak point. Angular height is defined by the angle created at the peak point. Relative height is defined as the height of the peak point relative to all the other points in the time series. This definition will identify spikes and sinks whose behavior is significantly different then the rest of the the points in the time series. The relative height attribute characterizes that difference. For example, the equation, $|(max - mean)|/\sigma$ could be used to define the relative height of a spike or sink.

The relative height attribute of a spike or sink shape is affected by the behavior of all the time points in the time series. The absolute and angular height definitions have the ability to identify spikes and sinks in a volatile time series. Volatile time series are characterized by large changes in opposite directions between a set of consecutive time points. The relative height definition identifies spike and sink behaviors that differ from the behavior of the rest of the points in the time series. The duration attribute is given by the sum of time points contained in both edges plus the peak point. Constraining these attributes can identify a specific spike or sink shape within a time series.

The absolute, angular, and relative height attributes, as well as the duration and edge slope attributes can be constrained to define different spike and sink shape behaviors, and they can be used as a ranking metric to compare and rank the shapes. The duration attribute can be constrained to identify sink and spike shapes that occur over a specific period of time. For instance, a three point and multiple point definition could be defined. The three point shape contains exactly three time points, a peak point and a single point on each side. Three points is the smallest number of points that a spike or sink shape can contain. The multiple point shape contains more than three time points.

The peak height can be constrained to create a definition that will identify shapes which are greater or less than a particular height. The slope of the leading or trailing period of change can also be used to define behaviors of interest for spike and sink shapes. By using these attributes to create shape definitions and rank shapes, particular behaviors of interest can be identified in temporal data sets. For example, a doctor may want to identify patients who suffer from intense panic attacks that last longer than ten minutes, where intensity is a measure of a patient's heart rate. This behavior can be identified by using a ten point spike ranked according to its angular peak height. Correlations can also be made using shape identification. Using the same example, a doctor may want to know how the length of a panic attack is related to the intensity of the attack. This requires using the general definition of spikes and ranking them according to their angular height. Then just identify correlations between the spikes that are highly ranked and their duration.

3.3 Rise and Drop Shapes

Rise and drop shapes are used to describe a sustained change in the average value. These shapes can be divided into three distinct periods: a period of change that is preceded and followed by periods of stability, Figure 3.3C. The stable periods

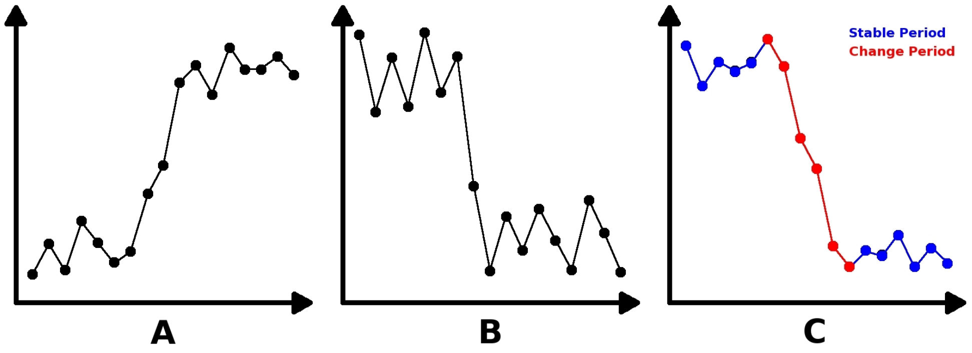


Figure 3.3: The graphs above are examples of rise and drop shape. Graph A is a rise. B and C are drops. Graph C shows the three periods of drop and rise shapes: the leading stable period, the change period, and the trailing stable period.

are drawn in blue and the period of change in red. A rise shape has a change period that increases in value, while a drop shape decreases in value, as seen in Figures 3.3A and 3.3B respectively. Each period must consist of one or more time points; there is a single transition point between each period; and the time points in the shape must be contiguous. Drop and rise shapes contain a minimum of five points. The periods of stability separate these shapes from spikes, sinks and lines.

Stable time points have very low volatility, which could be measured by the standard deviation of the points or some other definition. In drops and rises if a set of time points is not stable, it is changing. A rise and drop shape describes a person's heart rate at the start and conclusion of an aerobic workout, respectively. At the start of a workout a healthy person's heart rate will transition from a resting rate of approximately 65 beats per minute (bpm) to 140 bpm. During the period prior to and after the transition the active and resting heart rate will be stable until the conclusion of the workout. This is the type of behavior a rise or drop shape could identify.

The length of the periods, change significance, and average value of stable periods are some of the attributes associated with rise and drop shapes. The length of a period is defined by the number of time points contained within that period. The change significance, like the previous shapes, can be defined by the slope of that period, and the slope can be defined in several different ways based on the behavior of interest. The average value of the stable period is the mean of the points in the period.

Period length is the most intuitive attribute to constrain when creating shape definitions for rise and drop shapes. A definition that limits the length of the change period to just two points is useful in identifying rapid change. Using the workout example, constraining the length of the trailing stable period to be greater than 15 would identify workouts of longer than 15 time points. Consider a data set containing serveral connections per minute. By constraining the length of the change period to two and ranking the time series according to the average value of the trailing stable period, an information technology (IT) specialists would be able to start to identify anomalous behavior of traffic within their network.

3.4 Plateaus, Valleys and Gaps

Plateaus, valleys, and gaps are used to describe temporary changes in variable. They differ from spikes and sinks because the temporary value is sustained for a measurable period of time. These shapes consist of leading, intermediate, and

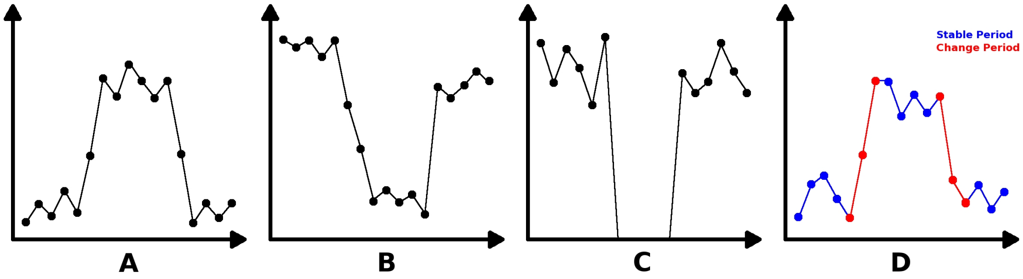


Figure 3.4: Graphs A, B and C show a plateau, valley and gap shape, respectively. Graph D shows the periods associated with plateau, valley and gap shapes.

trailing stable periods, as well as departing and returning change periods as shown in Figure 3.4D. A plateau has an intermediate stable period, whose average value is greater than the leading and trailing stable periods (Figure 3.4A), while a valley has an intermediate period, whose average value is less than the average value of the other two stable periods (Figure 3.4B). A gap is a specific type of valley where the intermediate period's values are zero (Figure 3.4C). Using the workout example, a plateau describes a person's heart rate during his or her entire workout. Prior to the beginning and after the end of the workout, the heart rate is stable at a resting rate of 65 bpm. At the start of the workout, the heart rate leaves the resting rate and rises to approximately 140 bpm. This heart rate is maintained throughout the workout. At the conclusion of the workout, the heart rate returns to the resting heart rate and remains there. Plateaus, valleys, and gaps are very similar to drops and rises with one important difference. Drops and rises do not define the behavior that occurs after the trailing stable period. Therefore, several ranking metrics, such as the length of the intermediate stable period (the trailing stable period in the drop and rise shape) have a different meaning in plateaus, valleys and gaps than in drop and rise shapes.

The ranking metrics are similar to the ranking metrics for drops and rises, but they are calculated over the additional portions of the plateaus, valleys, and gaps. Although the calculations are same, the meanings are different. For example, using the workout example, the difference between the mean of leading and trailing stable periods in plateau shapes may signify a strengthening of the heart. On the other hand, the difference between the leading and trailing periods in a rise shape signifies a more strenuous workout.

Definitions that constrain the length of the stable periods are useful when examining plateau, valley and gap shapes. By limiting the length of a particular period, shapes with a specific duration can be identified. Definitions that measure the difference between the leading and trailing stable periods can also be useful. For instance, the blood cell counts of a chemotherapy patient will depict a valley when graphed over time. Some chemotherapy drugs can cause cells to stop dividing, causing a drop in the cell count. Therefore, a patient's blood cell count should ideally have the same average value before and after chemotherapy. By ranking valleys according to the difference between the leading and trailing stable periods, a health professional may be able to start to question whether the treatment had an adverse effect on a patient's cell count or not.

3.5 Summary

Lines, spikes, sinks, rises, drops, plateaus, valleys, and gaps are shapes that can be used to describes the behavior of the values of a variable over a period of time. The attributes associated with each of these shapes characterize specific portions of the shape. The attributes can be used in combination to describe an exact behavior. For example, a stock that has risen consistently, from year to year, for a period of seven is defined by it slope and its length. These simple shapes and attributes offer a good balance between expressiveness and complexity.

Chapter 4

Interface Design

TimeSearcher Shape Searcher Edition (SSE), an extension of TimeSearcher 1, is a visualization tool that identifies and ranks shapes in temporal data sets. It uses the graphic routines provided by TimeSearcher 1, which are created using Piccolo [4]. It also maintains a similar window layout. Figure 4.1 shows the TimeSearcher SSE graphical user interface (GUI). In TimeSearcher the window in the upper left corner is used to create visual queries. This window is not used in TimeSearcher SSE and it can be hidden by clicking the on the arrow on the bar below the window.

TimeSearcher SSE consists of four primary windows. The shapes window on the left side contains time series graphs displaying each of the identified shapes. The tabbed window on the upper right side shows a details view, the time points and associated data values, of a time series in the details tab and the current shape definition in the definitions tab. The rankings window is on the right side in the center. This window displays the ranking metric for an individual shape and the label for the time series in which it is located. The shapes, details and rankings windows are tightly connected. Scrolling in the shapes window causes the rankings window to scroll, so that the first item in the rankings window is the same as the first graph in the shapes window. Selecting an item in the ranking window will cause the details for that time series to be shown in the details window and graph containing the shape to be the first one shown in the shapes window. Similarly, mousing over a graph in the shapes window will cause the details of the graph to be shown. The window on the lower right hand side contains range sliders which filter the identified shapes based on its endpoints and the value of the ranking metric.

The graphs in the shapes window are a visual representation of a time series. These graphs make it easy to identify the shapes created by plotting the values in a time series. The graph's y-axis is labeled with the range of values that the variable takes on throughout the entire data set. The x-axis is labelled with the time points. The axes are drawn in black, while the time series is plotted in gray. Each time point is represented by a small gray dot and each consecutive dot is connected by a gray line. Each shape is shown in its own graph; if a time series has more than one unique occurrence of a shape, then the graph of the time series will appear more than once. Each shape is labeled in the graph with red lines instead of gray; points of interest in the shape are marked by large red dots. A significant point may be the peak point in a spike or sink shape or the change period in a rise or drop shape.

The upper panel, Figure 4.2, shows the seven buttons labeled with the shapes that TimeSearcher SSE can identify and rank. Rolling over the buttons will cause a popup to be shown indicating the name of the shape. Each shape has several definitions that can be chosen from the definitions drop down box to the right of the shape buttons. Some of the shape definitions require user input, such as the number of time points in the shape. This allows more fine grain specification of the exact behavior of a particular shape. This field will only be enabled if the definitions require user input. The search button initiates the search for the selected shape of interest with the user defined information, if applicable. Only a subset of the shapes and definitions that were explained in this thesis were implemented in TimeSearcher SSE. These shapes are spikes, sinks, lines, rises, and drops.

4.1 Spikes and Sinks

Spike and sink shape buttons, the first and second buttons in Figure 4.2, identify several definitions of spike and sink shapes. These shapes are ranked according to their relative and angular heights. By clicking on the spike or sink button the definitions below will populate the drop down box to the right.

3 Point Angular	3 Point Relative
5 Point Angular	5 Point Relative
7 Point Angular	7 Point Relative
Multi-Point Angular	Multi-Point Relative

The ranking attribute is denoted as “Angular” or “Relative”. The multi-point definitions require user input. The user is required to indicate how many points should appear in the spike or sink shape.

4.2 Lines

Increasing and decreasing line shape buttons, the third and fourth buttons in Figure 4.2, enable the identification of line shapes. The increasing line button constrains the slope of the line to be positive, and the decreasing line button constrains it to be negative. Clicking the buttons populates the definitions drop down box with the four definitions listed below.

2 Point Slope

Multi-Point Slope

Longest Monotonic Slope

Monotonic Slope w/ Length > x

The two point line shape is ranked according to the geometric slope of the line segment, while the multiple point line shape is ranked according to the geometric slope of the linear regression calculated over the points in the shape. The multiple point line shape definition requires a user to specify the number of points in the line, including the endpoints. The final two definitions are line shapes with monotonically increasing or decreasing slope. The longest monotonic slope is ranked according to its length. The longest monotonic slope definition has no constraint on the number of points in the line. However the monotonic slope with length greater than x constrains the minimum length to x. This definition requires the user to input a minimum length for the identified line shapes. The shapes are identified by red lines and small red dots on the graph.

The volatility button, the last button in Figure 4.2, identifies line shapes with a high volatility. These shapes are ranked according to their standard deviation and the sum of the variation. By clicking the volatility button the definitions drop down box will be populated with ranking attributes. Because the entire line is considered the shape, the dynamic filter that constrains the start and end points does not work, and each time series only appears once.

4.3 Rises and Drops

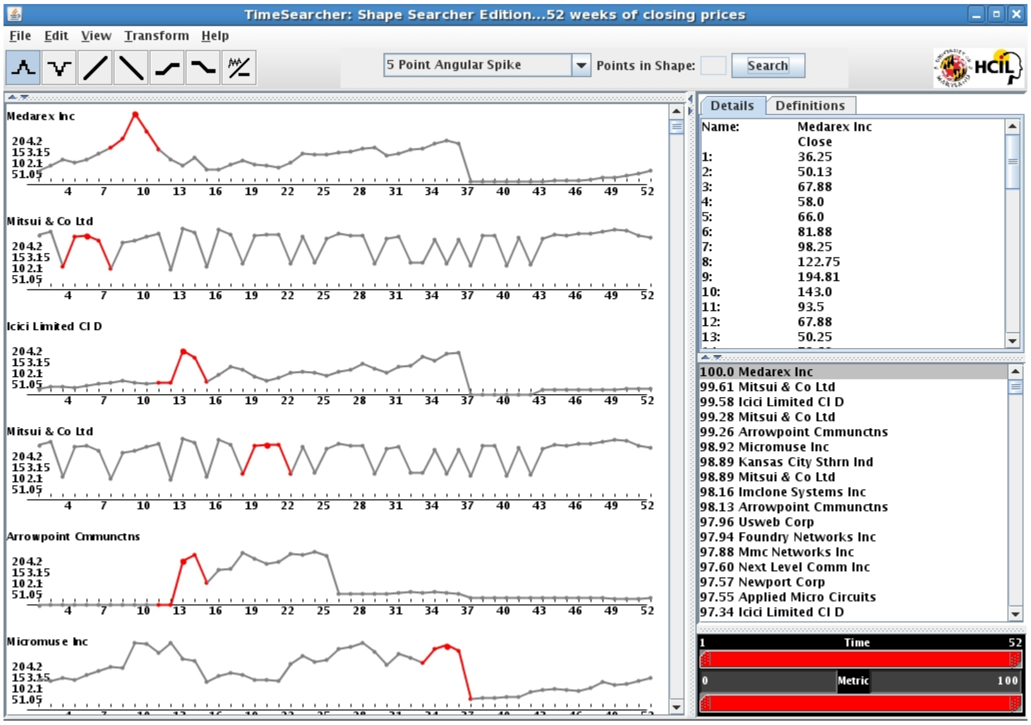
The rise and drop shape buttons, the fifth and sixth buttons in Figure 4.2, enable the identification of those shapes. The rise button constrains the period of change to be positive, and the drop button constrains it to be negative. Clicking on the drop or rise buttons will populate the definitions drop down box with the following definitions.

Slope

Length

Slope with Stable Length > x

The slope definition ranks the shapes according to the slope of the period of change. The length definition ranks them according to the total length of the shape. The "slope with stable length $> x$" definition constrains the length of the stable period to be greater than x, where x is defined by the user. The beginning and end of the change periods of drop and rise shapes are marked by large red dots.



Name:	Medarex Inc
	Close
1:	36.25
2:	50.13
3:	67.88
4:	58.0
5:	66.0
6:	81.88
7:	98.25
8:	122.75
9:	194.81
10:	143.0
11:	93.5
12:	67.88
13:	50.25

100.0 Medarex Inc
99.61 Mitsui & Co Ltd
99.58 Icici Limited Cl D
99.28 Mitsui & Co Ltd
99.26 Arrowpoint Cmmunctns
98.92 Micromuse Inc
98.89 Kansas City Sthrn Ind
98.89 Mitsui & Co Ltd
98.16 Imclone Systems Inc
98.13 Arrowpoint Cmmunctns
97.96 Usweb Corp
97.94 Foundry Networks Inc
97.88 Mmc Networks Inc
97.60 Next Level Comm Inc
97.57 Newport Corp
97.55 Applied Micro Circuits
97.34 Icici Limited Cl D

Figure 4.1: This is a screenshot of TimeSearcher Shape Searcher Edition (SSE). The upper panel shows the seven buttons labeled with the shapes that TimeSearcher SSE can identify and rank. Each shape has several definitions that can be selected from the drop down box to the right of the shape buttons. Some of the shape definitions require user defined input, such as the number of time points in the shape. The left side contains the shapes window, which displays the currently identified shapes for the loaded data set. The window in the upper right contains the details and definitions tab. The details tab displays the time points and values of a particular time series. The definition tab displays an explanation of the selected shape definition. The window in the left center is the rankings window. Once a shape and definition have been chosen from the upper panel the ranking metric value and label for each shape will be shown in this window. The lower right corner contains the dynamic query bars. These bars allow the shapes to be filtered based on the ranking metric and the endpoints associated with a shape.

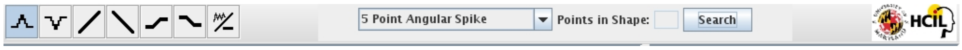


Figure 4.2: The TimeSearcher SSE button panel, shown above, enables spike, spike, increasing, decreasing, rise, drop, and volatility shape searches, respectively. Clicking on any of buttons will populate the definitions drop down box, to the right of the buttons, with a set of definitions for the selected shape. The input field is used to take user input for certain definitions, and the search button populates the shapes window and rankings windows with the identified shapes.

Chapter 5

Implementation

TimeSearcher Shape Search Edition (SSE) was implemented as an extension of TimeSearcher 1(TimeSearcher) by Harry Hochheiser. It is implemented in Java 1.6. The user interface is the same as TimeSearcher, and the original code base for TimeSearcher was left unchanged, so that the TimeSearcher functionality would still work. TimeSearcher's timeboxes, graphs, and data envelopes are implemented in Piccolo[4], a zooming toolkit. TimeSearcher SSE does not use the timeboxes, nor the query and data envelopes. SSE is the result of approximately six months of work and is currently in a beta development stage. The following sections will walk through an overview of the changes made to TimeSearcher to create SSE, SSE's implementation, and the lessons learned from the development of the tool.

5.1 Overview

TimeSearcher required several changes to support shape identification and ranking. This section offers an overview of the purpose of each of the packages and the changes that were made to to create SSE.

- **edu.umd.cs.temporalquery** – This package contains the TQCore, TQMain, CmdTable and TQMenuBar classes. The TQCore class provides the core functionality for TimeSearcher and SSE. The TQCore class in combination with the TQMenuBar and CmdTable provides an interface to the SSE functionality. TQCore contains the functionality that enables the shape search buttons triggered by the "Shape Search Buttons" item in the view menu. TQMain starts the application and no modifications were made to its functionality.

- **edu.umd.cs.temporalquery.data** – This package contains the DataSet class, the primary data model for the TimeSearcher and SSE. The DataSet object is composed of Entity objects, which contain all the information relating to an individual time series, such as the values and statistical information associated with the time series. SSE uses the Entity class to store all of the shapes associated with a particular time series. Shapes that require no user input are identified when the DataSet object is created and are stored in the appropriate Entity object. SSE also added a mean and standard deviation calculation to the Entity class, because their values are used by several of the shape definitions. This package also contain four utility classes to help deal with the different types of data that TimeSearcher and SSE must handle.

- **edu.umd.cs.temporalquery.event** – This package contains a single class that handles information relating to query modification. This class was not altered, because the query infrastructure is not used by SSE.

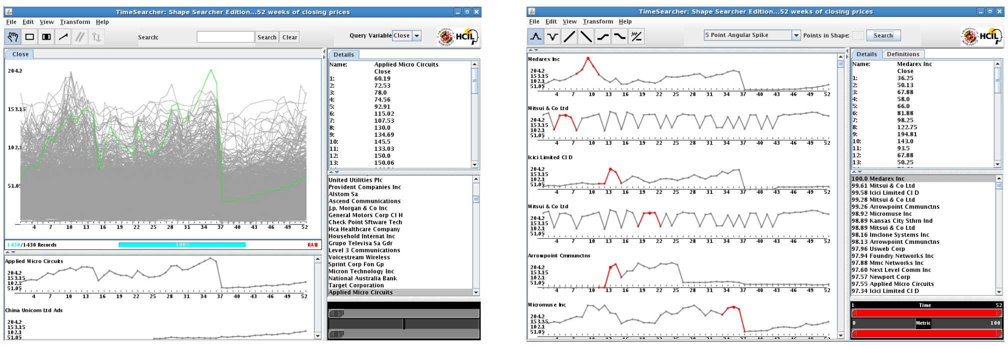
- **edu.umd.cs.temporalquery.graph** – In TimeSearcher this package is responsible for displaying the items that match the current query; in SSE it is responsible for displaying the shapes that match the current definition. This required a fundamental change in the GraphSet class. In TimeSearcher an Entity represents a single graph, but in SSE the Entity contains multiple shape graphs. The GraphSet class was altered to plot multiple shape graphs instead of a single time series. This class relies on other classes to do the actual drawing of the graphs; those classes were also changed.

- **edu.umd.cs.temporalquery.images** – This package contains all the images used by TimeSearcher and SSE, such as those used for the buttons, logo, and splash screen.
- **edu.umd.cs.temporalquery.piccolo** – This package contains all the extensions to the Piccolo classes that are used to provide graphic support for the query space and rankings window. The DataAxis class is responsible for the actual drawing of the graph containing the shapes. The drawing routines were changed to understand how to draw a TShape instead of an Entity. The classes supporting the query space functionality in TimeSearcher were left unaltered.
- **edu.umd.cs.temporalquery.pwindow** – This package contains classes that support the Piccolo functionality within the query window. This functionality is not used by SSE, so no modifications were made.
- **edu.umd.cs.temporalquery.query** – This package contains several classes for maintaining a set of active queries. This functionality is not used by SSE, so no modifications were made.
- **edu.umd.cs.temporalquery.rangeslider** – The IntRangeSlider and FloatRangeSlider classes support querying of the shapes based on the ranking

metric and the beginning and ending time points of the shape. These classes were written in such a way that no modifications were needed.

- **edu.umd.cs.temporalquery.shapes** – This package contains all of TimeSearcher SSE's core functionality. The TShape object is used to represent individual shapes. The TShapeQuery class serves as the primary interface by which shapes are accessed. The IncDec, SpikeSink, RiseDrop and VolatileStable classes contain the shape definitions. This package will be explained in detail later in the paper.

- **edu.umd.cs.temporalquery.util** – This package contains a variety of utility classes that support threading, logging, file selection filters, pop-up menus, and widgets specific to TimeSearcher. No modifications were required for any of these classes.



Name: Applied Micro Circuits
Close
1: 60.19
2: 72.53
3: 78.0
4: 74.56
5: 92.91
6: 115.02
7: 107.53
8: 130.0
9: 134.69
10: 145.5
11: 133.03
12: 150.0
13: 150.06

United Utilities Plc
Provident Companies Inc
Alstom Sa
Ascend Communications
J.p. Morgan & Co Inc
General Motors Corp Cl H
Check Point Sftware Tech
Hca Healthcare Company
Household Internat Inc
Grupo Televisa Sa Gdr
Level 3 Communications
Voicestream Wireless
Sprint Corp Fon Gp
Micron Technology Inc
National Australia Bank
Target Corporation
Applied Micro Circuits

Name: Medarex Inc
Close
1: 36.25
2: 50.13
3: 67.88
4: 58.0
5: 66.0
6: 81.88
7: 98.25
8: 122.75
9: 194.81
10: 143.0
11: 93.5
12: 67.88
13: 50.25

100.0 Medarex Inc
99.61 Mitsui & Co Ltd
99.58 Icici Limited Cl D
99.28 Mitsui & Co Ltd
99.26 Arrowpoint Cmmunctns
98.92 Micromuse Inc
98.89 Kansas City Sthrn Ind
98.89 Mitsui & Co Ltd
98.16 Imclone Systems Inc
98.13 Arrowpoint Cmmunctns
97.96 Usweb Corp
97.94 Foundry Networks Inc
97.88 Mmc Networks Inc
97.60 Next Level Comm Inc
97.57 Newport Corp
97.55 Applied Micro Circuits
97.34 Icici Limited Cl D

TimeSearcher 1 **TimeSearcher SSE**

Figure 5.1: This shows the original TimeSearch 1 interface and the TimeSearcher SSE interface.

- **edu.umd.cs.temporalquery.windows** – This package contains the classes used to build the TimeSearcher (GUI). Many of the classes were modified to support SSE functionality. Figure 5.1 shows the differences between TimeSearcher 1 and TimeSearcher SSE interfaces. Modifications were made to the TQDetails to create a tabbed pane that could display details about the time series and a detailed explanation of the currently selected shape definition. In TimeSearcher the TQItemList class was used to display the static labels for each of the time series, but in SSE, the TQItemList displays both the calculated ranking metric for a shape and the label for the time series in which that shape appears.

5.2 Data Representation

The data TimeSearcher SSE uses to identify and rank shapes is stored in a DataSet object. The DataSet object contains all of the time series data. Each time series and all information relating to an individual time series is stored within an Entity object, which contains the time series values, identified shapes that require no user input, maximum and minimum values, standard deviation, and mean. The data is read in the form of an input file. The input file format is described below.

The input file for TimeSearcher SSE is a plain text of the following format:

1. Title: a descriptive title for the data file
2. Static attributes: static information for each data item, such as a label associated with a time series. The static attributes are given in the form, “Name, Type.” The “Name” is the name of the attribute, and the “Type” is the data

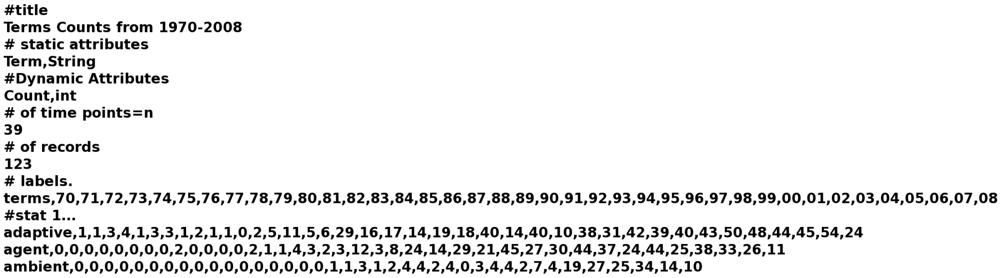
type of the attribute. The supported data types are string, float, and int.

3. Number of time points: the number of values in each time series

4. Number of items: the number of time series in the data set

5. Time point labels: the text labels that are associated with each time point.

6. Individual items: a comma separated list of the time series' values preceded by the static attribute. Each item is separated by a new line character.



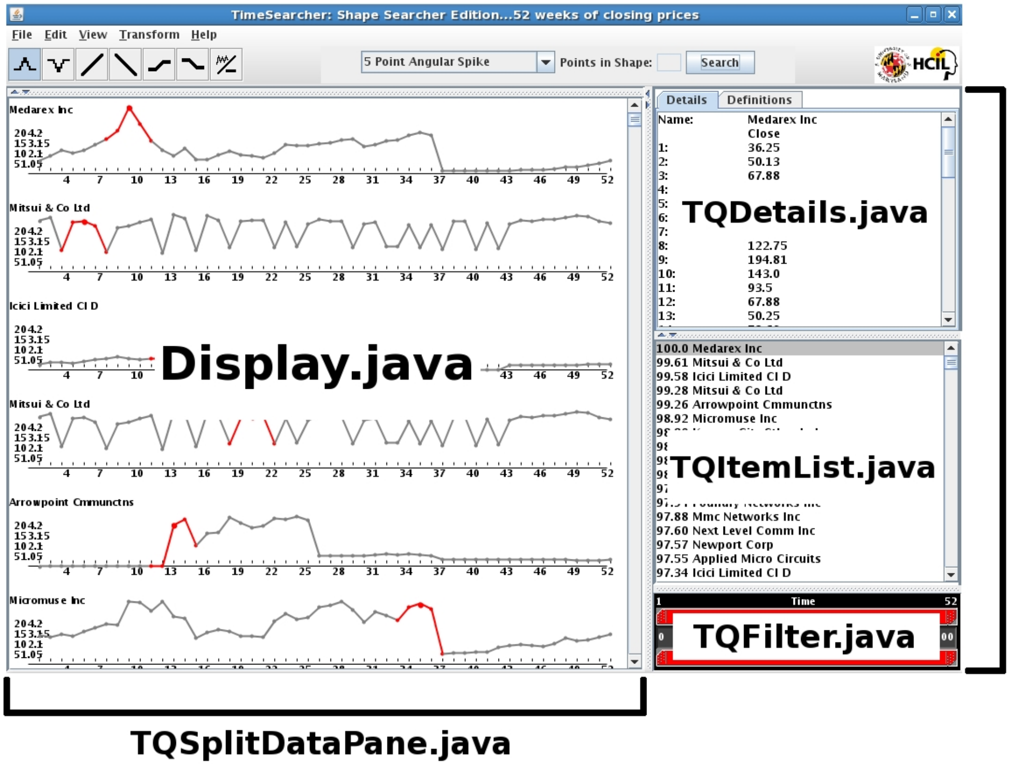
```
#title
Terms Counts from 1970-2008
# static attributes
Term,String
#Dynamic Attributes
Count,int
# of time points=n
39
# of records
123
# labels.
terms,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,00,01,02,03,04,05,06,07,08
#stat 1...
adaptive,1,1,3,4,1,3,3,1,2,1,1,0,2,5,11,5,6,29,16,17,14,19,18,40,14,40,10,38,31,42,39,40,43,50,48,44,45,54,24
agent,0,0,0,0,0,0,0,0,2,0,0,0,0,2,1,1,4,3,2,3,12,3,8,24,14,29,21,45,27,30,44,37,24,44,25,38,33,26,11
ambient,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,3,1,2,4,4,2,4,0,3,4,4,2,7,4,19,27,25,34,14,10
```

Figure 5.2: Excerpt of a TimeSearcher input file

Once the data has been read, preprocessed and stored, a user can begin to explore data through the GUI. The GUI allows the user to select a shape and a definition of interest. The next section describes the classes that create the graphical user interface.

5.3 Graphical User Interface

The GUI consists of six classes: TQDetails, TQItemList, TQFilter, TQControl, Display, and TQSplitDataPane. TQDetails, TQItemList, TQFilter, TQCon-



```
Name:     Medarex Inc
          Close
1:        36.25
2:        50.13
3:        67.88
4:
5:
6:
7:
8:        122.75
9:        194.81
10:       143.0
11:       93.5
12:       67.88
13:       50.25
```

```
100.0 Medarex Inc
99.61 Mitsui & Co Ltd
99.58 Icici Limited Cl D
99.28 Mitsui & Co Ltd
99.26 Arrowpoint Cmmunctns
98.92 Micromuse Inc
97.88 Mmc Networks Inc
97.60 Next Level Comm Inc
97.57 Newport Corp
97.55 Applied Micro Circuits
97.34 Icici Limited Cl D
```

Figure 5.3: This figure is the TimeSearcher SSE interface with labels applied to show the Java files that are responsible for creating each window.

trol, and TQSplitDataPane are located in the edu.umd.cs.temporalquery.windows package, while Display is a part of the edu.umd.cs.temporalquery.pwindows package. Figure 5.3 shows which Java file creates each of the labeled windows. The TQControl class contains the TQDetails, TQItemList and TQFilter classes. The TQSplitDataPane class contains the Display class. Below are descriptions of each of these classes, how they work, and how they interface with the rest of the code.

- **TQControl** – The TQControl contains the TQDetails, TQItemList, and TQFilter classes. It uses the JSplitPane class to allow either the details and definitions tabbed pane or the ranking window to be expanded to fill the entire panel.

- **TQControl** – The TQDetails class is a tabbed pane. One tab contains the details view, while the other tab contains an explanation of the current selected shape definition. The MouseListener interface is used to synchronize the details tab, TQItemList, and Display. On a mouse over event from the Display object the details of the graph the mouse is over are displayed. On a mouse click event from the TQItemList the details about the item clicked are displayed.

- **TQItemList** – The TQItemList object creates the rankings window in which the ranking metric and the static attributes of the shapes are displayed. The AdjustmentListener interface is used to synchronized the rankings window and the shapes window. When scrolling takes place in the shapes window, an adjustment event occurs. The TQItemList receives this event, and scrolls to the proper item in the list based on the visible shapes in the shapes window.

- **TQFilter** – The TQFilter object creates the filter window that allows shapes to be filtered based on their start and end points and the ranking metric. The filter window contains two dual slider bars to facilitate the filtering. The code is based on code by Ben Bederson and Jon Meyer.

- **TQSplitDataPane** – This class contains the Query and Display classes. TimeSearcher SSE does not use the window created by the Query class. The Query window can be hidden because the TQSplitDataPane extends the JSplitPane class. The AdjustmentListener interface is used to synchronize the shapes window with the rankings window. When the TQSplitDataPane receives an adjustment event from the TQItemList object, the scroll pane containing the Display object synchronizes the graphs displayed with the items in the rankings window.

- **Display** – The Display class creates the shapes window. The Display object displays the graphs in the GraphSet object. The GraphSet object contains the graphs that match the current shape definition. These graphs are drawn by the DataAxis class, which uses the Piccolo library to create the individual shape graphs.

5.4 Shape Identification

The primary package of TimeSearcher SSE is edu.umd.cs.temporalquery.shapes. This package contains, the IncDec, SpikeSink, RiseDrop and VolatileStable classes. These classes contain the definitions for each of the shapes in TimeSearcher SSE. The TShape class contains the fundamental shape representation for TimeSearcher SSE. TSQuery is the primary interface for accessing the shapes. The ShapeUtil class contains utility functions used by the rest of the package. Shapes with the same attributes are located in the same file, such as spikes and sinks and increasing and decreasing lines. Below, each of the classes will be described. The **definitions** will be in bold and the *ranking metrics* in italics.

5.4.1 SpikeSink Class

The SpikeSink class contains three definitions for both spike and sink shapes. A spike shape is defined as an increasing edge followed by a decreasing edge that meet at a single point, while a sink shape is a decreasing edge followed by an increasing edge. An edge is a set of time points. The definitions define a three, five, and seven point spike and sink and each of these shapes can be ranked according to its relative and angular height. Each of the shapes and its ranking metrics are described below:

- **3-Point Spike/ Sink** – a spike or sink shape containing exactly three time points with a single time point on both sides of the peak point.

- **5-Point Spike/ Sink** – a spike or sink shape containing exactly five time points with two time points on both sides of the peak point.

- **7-Point Spike/ Sink** – a spike or sink shape containing exactly seven time points with three time points on both sides of the peak point.

- *Angular Height* – the measure of the angle created at the point where the edges meet. Figure 5.4A shows the component's angular height calculation. Using the trigonometric function $cos(\alpha+\beta) = cos(\alpha)*cos(\beta) - sin(\alpha)*sin(\beta)$ the angle created by the edges of the spike is equal to $cos(\alpha + \beta) = (dy1 * dy2 - 1)/\sqrt{(1 + dy1^2) * (1 + dy2^2)}$ where $dy1 = |y1 - y2|$ and $dy2 = |y2 - y3|$. A linear regression calculated over the points to the right and left of the peak point defines the increasing and decreasing edges for the 5 and 7-point spikes.

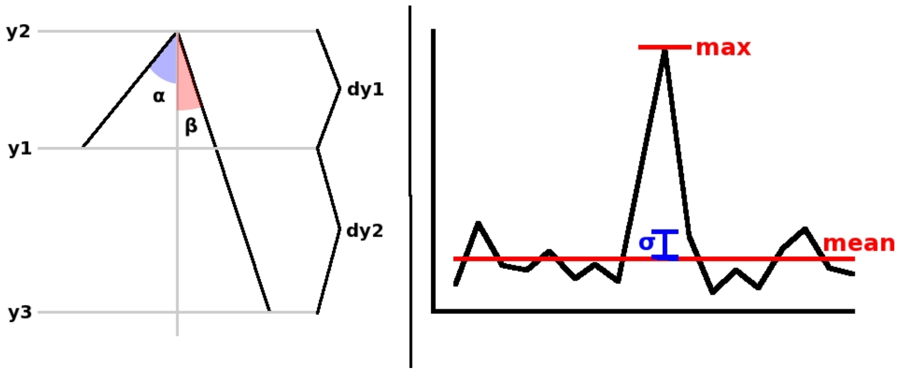


Figure 5.4: The diagrams above show how the angular and relative height attributes are calculated. The first image shows the components of the angular height equation, $cos(\alpha + \beta) = (dy1 * dy2 - 1)/\sqrt{(1 + dy1^2) * (1 + dy2^2)}$. The angular height a measure of the angle created where the two edge of spikes and sinks meet. The second image shows the components of the relative height equation, $|max - mean|/\sigma$. The relative height is the height of a spike or sink relative to the rest of the shape.

- *Relative Height* – the height of the peak point from the mean of the time series measured in standard deviations. The relative height is given by the equation $|max - mean|/\sigma$. Figure 5.4B shows the values of the relative height calculation.

All of the definitions and ranking metrics are static, and require no input from the user. Each shape is computed when the data is loaded. Values such as the mean and standard deviation are only calculated once and stored within the internal representation of a time series, an Entity object. The function that identifies the spikes and sinks takes a parameter that defines how many points will be in a spike or sink. These shapes are identified simultaneously. The class attempts to identify shapes as efficiently as possible, by only passing through the data once. Figures 5.5 – 5.7

show examples of spikes and sinks identified by TimeSearcher SSE's SpikeSink class.

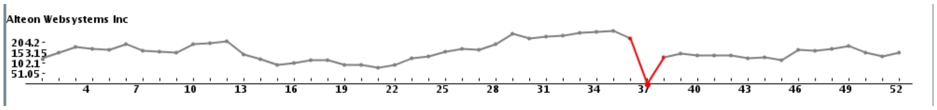


Figure 5.5: An example of a three point sink ranked according to its angular height. This sink identifies a missing value in this stock market data.

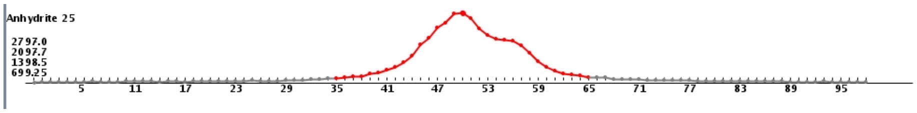


Figure 5.6: An example of a 31 point spike in X-ray diffraction data ranked according to its angular height.

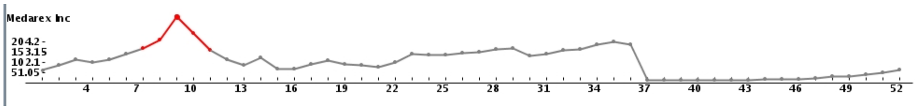


Figure 5.7: An example of a five point spike in a stock price that is highly ranked according to its angular height.

5.4.2 IncDec Class

The IncDec class contains four definitions for both increasing and decreasing line shapes. The first three shape definitions are two point, multiple point, and monotonic slope line shapes. The fourth definition is a monotonic slope line shape with a constraint placed on the minimum length. The two point and multiple point definitions are ranked according to their slope, while the monotonic slope definition is ranked according to its length and slope. The definitions for each shape and ranking metrics are listed below:

- **2-Point Line** – a line shape that contains only two time points. An increasing line has a positive slope, while a decreasing line's slope is negative.
- **Multiple Point Line** – a line shape that contains multiple time points. An increasing line has a positive geometric slope, while a decreasing line's slope is negative. There are several ways to measure the slope which are discussed below.
- **Monotonic Slope Line** – a line shape where each line segment's geometric slope has the same sign, positive or negative.
- *Slope* – the geometric slope is given by the equation, $(y2-y1)/(x2-x1)$. The slope of a two point line shape or a line segment can be calculated using the geometric slope equation. A multiple point line's slope is a measure of the geometric slope of the linear regression calculated over the points in the line shape. The slope of a monotonic slope line is calculated in the same fashion.
- *Length* – the number of time points contained in the line shape.

This class contains functions to identify multiple point lines and lines with monotonic slopes. Both functions require the user to identify multiple point lines of a particular length and monotonic slope line greater than a particular length. This allows the user to specify a minimum length for the monotonic slope lines, eliminating the two point lines, which are always monotonic. Figures 5.8 – 5.9 shows examples of increasing and decreasing lines identified by the IncDec class.

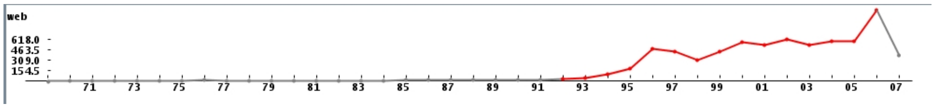


Figure 5.8: An example of a fifteen point increasing line ranked according to slope. This line shows the term “web” increasing over a fifteen year period.

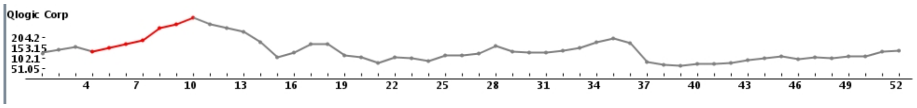


Figure 5.9: An example of a monotonically increasing line in stock market data ranked highly due to its slope.

5.4.3 RiseDrop Class

The RiseDrop class contains three definitions for both rise and drop shapes. These definitions are ranked according to their slope and the length of their stable periods. The definitions defined in the RiseDrop class are general definitions described in Chapter 3, and the same definition except the length attribute of the stable periods is constrained to be a minimum length. Listed below are the definitions:

- **Rise or Drop** – a sustained change in values. These shapes consist of three distinct time periods: a stable period, followed by a period of change, concluding with another stable period.

- **Drop or Rise with Multiple Point Stable Period** – a rise or drop shape that contains multiple points in each of its stable periods.

- *Slope* – the geometric slope of the period of change. The slope of the period of change and a line shape are calculated the same way.

- *Length of the Stable Periods* – the lowest number of time points between the two stable periods.

A point is stable if it lies within one standard deviation of the mean of the other points within the stable period. If a point is not stable, then it is changing. Figures 5.10 – 5.11 are examples of rise and drop shapes.

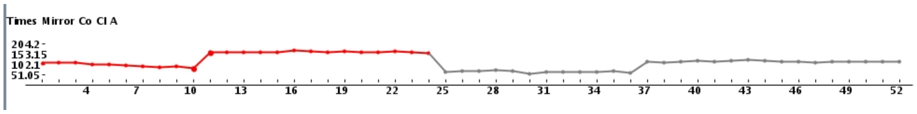


Figure 5.10: An example of a rise shape in stock market data. The shape is highly ranked according to the length of its stable periods.

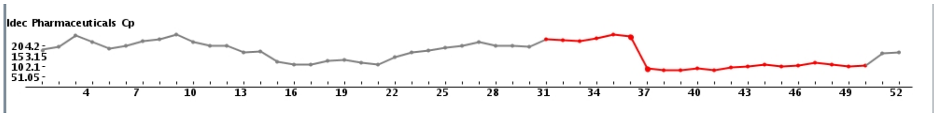


Figure 5.11: An example of a drop shape in stock market data. This drop was identified using the "stable period greater than x" definition which is ranked according to the slope of the change period.

5.4.4 VolatileStable Class

The VolatileStable class contains just a single definition and ranking. The definition ranks the time series according to its standard deviation. The time series with the greatest standard deviation is the most volatile, while the time series with the smallest standard deviation is the most stable. Figures 5.12 – 5.13 are examples of volatile lines identified by the VolatileStable class.

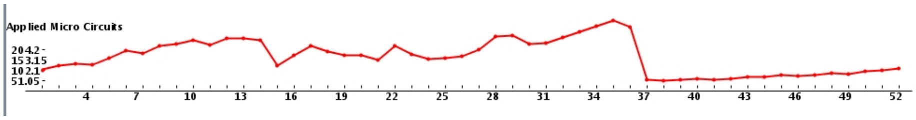


Figure 5.12: An example of a volatile line shape ranked highly according to its standard deviation.

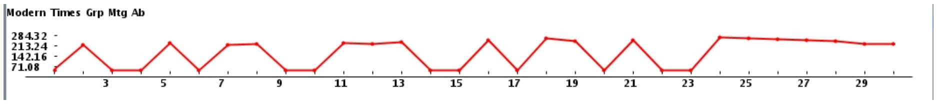


Figure 5.13: An example of a volatile line shape ranked highly according to its standard deviation.

Chapter 6

TimeSearcher SSE Case Study

TimeSearcher SSE was given to a single user to explore data sets within his expertise where shape identification and ranking would be useful. The user participated in 4 one hour user sessions. In each of the sessions, he looked at a single data set commenting on discoveries he made and problems with the interface. In this case study, he examined a network traffic data set and three x-ray diffraction data sets.

6.1 Network Traffic Data Set

The user is currently a Computer Science Researcher focusing on computer network defense. He has spent 2 years developing information visualization tools for anomaly detection in network traffic data. He believed that shape detection would be beneficial in detecting anomalies in network traffic data. He was unsure whether it would be more beneficial then current techniques, but he was willing to use TimeSearcher SSE to explore some data sets and see if any undetected anomalies were found.

In the first session he was interested in looking at network traffic data. He was interested in using SSE's ability to identify and rank shapes to find anomalous behavior in network traffic. It was his belief that spikes and rises could be associated with particular network events like botnet denial of service (DoS) attacks and sustained traffic increases, respectively.

The IP traffic data set consisted of a year of connection data for a specific set of IP addresses. The data were organized to show the number of connections by each IP address over twenty four hour periods. This data set was loaded into TimeSearcher SSE. The data set was very sparse; a particular IP often only visited once or twice per day. TimeSearcher SSE enabled the user to find several spikes, but none of the them were of any interest to the analyst. After using the tool for an hour and not getting the results he expected, he expressed the idea that the tool was better suited for X-ray diffraction data and/or spectroscopic data, which he had dealt with in a position earlier in his career.

6.2 X-Ray Diffraction

The user was a former research physicist specializing in non-destructive testing for explosives detection in aviation security. The user had 5 years experience in computed tomography and X-ray diffraction, both angular and energy dispersive. X-ray diffraction was used for materials identification and materials research; spectrum matching was accomplished using intensity peak heights and location. Typically JADE, software for powder diffraction data analysis, is used to match peak heights with angular positions. In angular dispersive X-ray diffraction (ADXRD) peaks are very sharp as opposed to energy dispersive X-ray diffraction (EDXRD), where peaks are much broader. In EDXRD it is much harder to correctly identify materials, but the technique is much faster for data acquisition. It is his opinion that shape detection could be used to improve material identification via X-ray diffraction and in other areas of spectroscopic techniques.

6.2.1 Session 2

In the second session, the user explored a second data set that consisted of several X-ray diffraction samples. X-ray diffraction is used to observe properties of materials, such as their chemical composition or a specific physical property by shining an x-ray beam on a material across a range of angles and measuring the scattered intensity of the x-ray as a function of the incident and scattered angle, polarization, and wavelength. The intensity readings produced by this process can be used as a fingerprint for a material. The fingerprint consists of intensity readings at various angles. For example, the element copper (Cu) may produce high intensity readings at angles 19.65, 23.0, and 33.5. Similarly, materials containing copper such as covellite (CuS) would produce high intensity readings at similar angles. Current techniques attempt to match the angular position and peak height of a known material with an unknown material. For example, Figure 6.1 contains the graphs of three materials containing iron (Fe). Each graph shows the angular positions 30 – 35. Each material has a peak between positions 33 – 34.5. A hypothesis can be formed that these peaks were created by the presence of Fe in the material. The goal of this case session of the case study was to load the X-ray diffraction data into TimeSearcher SSE and see if a common material could be identified. The identification of a common material would be similar to the process of identifying a component in an unknown material.

The data were obtained from an online database[22], because the user had no access to an apparatus performing the X-ray diffraction process. Three materials having at least one element in common were chosen. The materials were Arsenopyrite (FeAsS), Berthierite ($FeSb_2S_4$), and Awaruite (Ni_3Fe), which all contain Fe. Each sample contained approximately 8500 intensity readings over a large range of angles . Each of the samples was obtained from a different source, and the X-ray diffraction data was taken by a different machine. This was representative of an actual scenario, but caused several problems.

TimeSearcher SSE was not designed to handle such a large number of time points in a single time series, so each sample was divided into 85 separate samples with 100 time points in each sample. Each file contained a set of angular positions. The raw data values were used to maintain the integrity of the data. The data values could have been averaged to reduce the size of data set. Splitting the sample caused a problem that limited TimeSearcher SSE's effectiveness. The effectiveness of its dynamic sliders was reduced because the angular positions were spread over several files. Therefore, the user was unable to query the shapes based on their angular position. This was complicated by the fact that the samples were acquired from different machines. The machines that perform the X-ray diffraction process are very sensitive and have to be calibrated to perform optimally. Different machines may produce spikes at different positions based on their calibration. Splitting the data into multiple time series eliminated TimeSearcher SSE's ability to identify some spikes, because the spikes were spread over multiple time series. Figure 6.2 is an example of a spike that spans two time series. It also limited the user's ability to match the angular position with the spike.

In spite of the problems, the user was still able to identify some spikes with matching intensities and angular positions. By experimenting with spikes containing varying number of points, the user was able to see that Arsenopyrite and Berthierite both had spikes at angular position 33 and 34, respectively. A ten point spike definition was used and the results were ranked according to their angular height. The Arsenopyrite and Berthierite spikes were ranked consecutively with normalized values of 97.00 and 97.34, respectively, as shown in Figure 6.3. The third material, Awarite, did not have a spike ranked at the 33^{rd} or 34^{th} angular positions. But using an eight point spike definition ranked according to the angular height, Awarite and Arsenopyrite appear in the ranking window close together. The angular height of Awarite's spike at position 33 has a normalized value of 98.3, and the Arsenopyrite has a value of 96.3, as shown in Figures 6.4. Although a definition and ranking metric that ranked the spikes in similar position for all three materials together was not found, a correlation could be drawn from the results. However, the user was not satisfied with these results, and based on his experience the user suggested a more controlled data set, which was used in the third session.

6.2.2 Session 3

In the third session, a data set was created based on the criteria given by the user; all samples should come from the same machine and silicon should be a common element in all of the materials. The same machine criteria is based on the possibility that variations in spike position and intensity may be caused by differences in machine calibrations. The second criteria stating all samples should contain silicon is due to silicon's well known properties. In fact, silicon is commonly used as a calibration element because it produces a set of well defined peaks at known locations. The only material data available meeting these criteria was for Cristobalite (SO_2), Forsterite (Mg_2SiO_4), and Opal (SiO_2nH_2O), Figure 6.5. First, the data set was explored with no guidance and no discoveries were made. After identifying the positions of interest, 21-22, from the graphs of the raw data, the times containing the angular positions of 21 through 22 were examined. The Cristobalite spiked in the 21 – 22 position range, but the spike spanned two time series, Figure 6.2. Forsterite and Opal also spike within that range, but the spikes were not similar by any metric supported by TimeSearcher SSE.

6.2.3 Session 4

ADXRD is a very precise technique, but has drawbacks in sample preparation and time. Therefore, the user suggested loading infrared (IR) diffraction data into TimeSearcher SSE because he felt it would represent a more accurate portrayal of the type of data typical of EDXRD. Spikes in the IR data are broader and the data were less noisy. In past work, research has shown that a properly calibrated ADXRD data generally produced very sharp peaks, and depending on the angular step, could produce high noise if not properly calibrated and then normalized. Based on the fact that ADXRD data are noisy, the user believed that the IR data more accurately depicted data in fields of study where he thought shape detection would be beneficial. IR diffraction data was collected for Anhydrite ($CaSO_4$) and Baryte ($BaSO_4$) , Figure 6.6. Spikes at similar wavelength with similar intensities were identified in each material, as shown in Figure 6.7. The spikes had normalized values of 99.97 in both the Anhydrite and Baryte samples.

Two additional features were implemented during the case study to help the user more easily navigate the data sets, and the user suggested an improvement to the overall usability of the tool. The first problem the user encountered was that the shape definitions supported by TimeSearcher SSE did not fit the data set. Each of the data sets were very large, and each sample consisted of 8500 data points. Spikes sometimes consist of 20 or more points. However, the large spike definition supported by the tool was only seven points. To make the spike identification with the tool more flexible multiple point parametrized spike definitions were added to the interface. The second problem was that many of the shapes that were identified were only slight variations of the small set of points. For example, two seven point spikes in the same time series that differ by one point would be identified and have a similar ranking. This would create clutter and obscure the interesting shapes. This problem was corrected by choosing the shape with ranking value and removing all overlapping shapes.

The user also offered two suggestions that he believed would make the tool more useful. First, he thought the interface was too rigid and did not allow hime to use shape identification and ranking features the way he would like. A design for a new interface is discussed in Chapter 9. He also suggested that resolution of the data be considered when calculating shapes. This is an interesting point; the resolution is currently hidden with shape definitions, but it might simplify the definition if the resolution of the shapes was exposed as an attribute. Ideas about adjusting the resolution of data sets are explored in Berry et al. BinX, information visualization tool[5]. Incorporating ideas about resolution and shape identification will be left for future work.

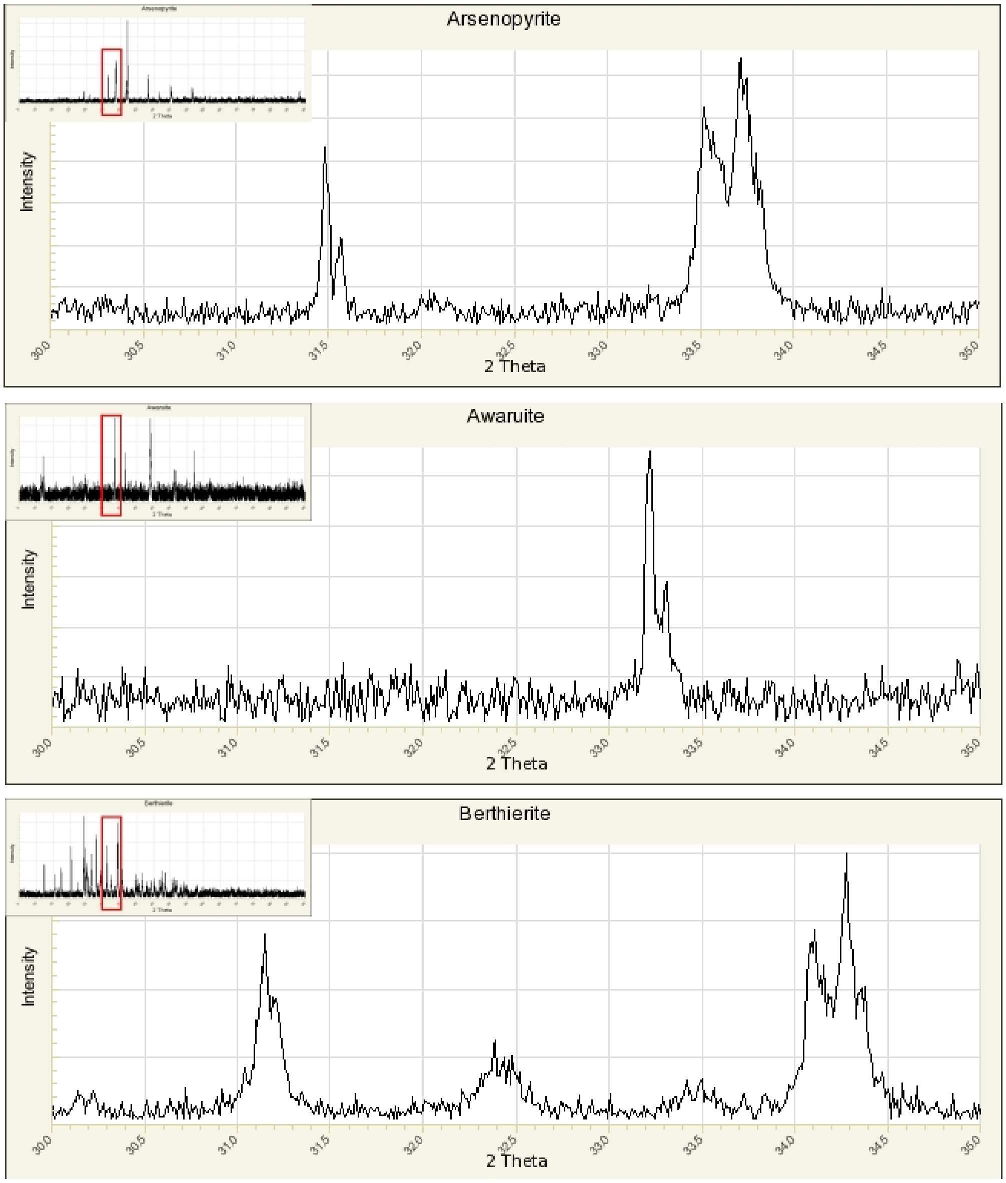
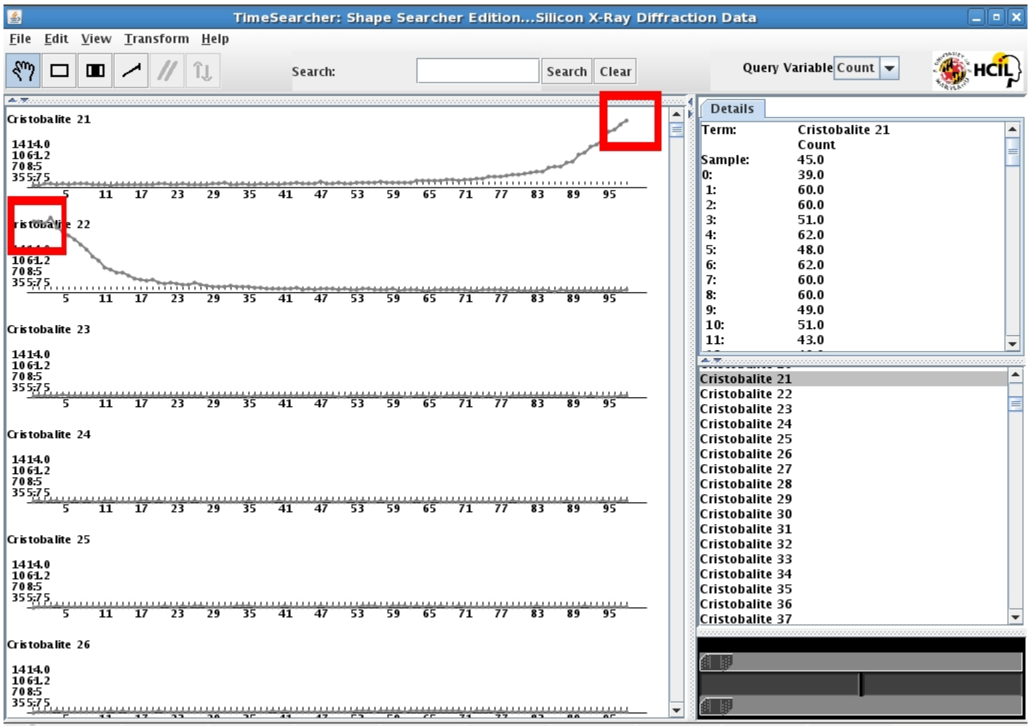


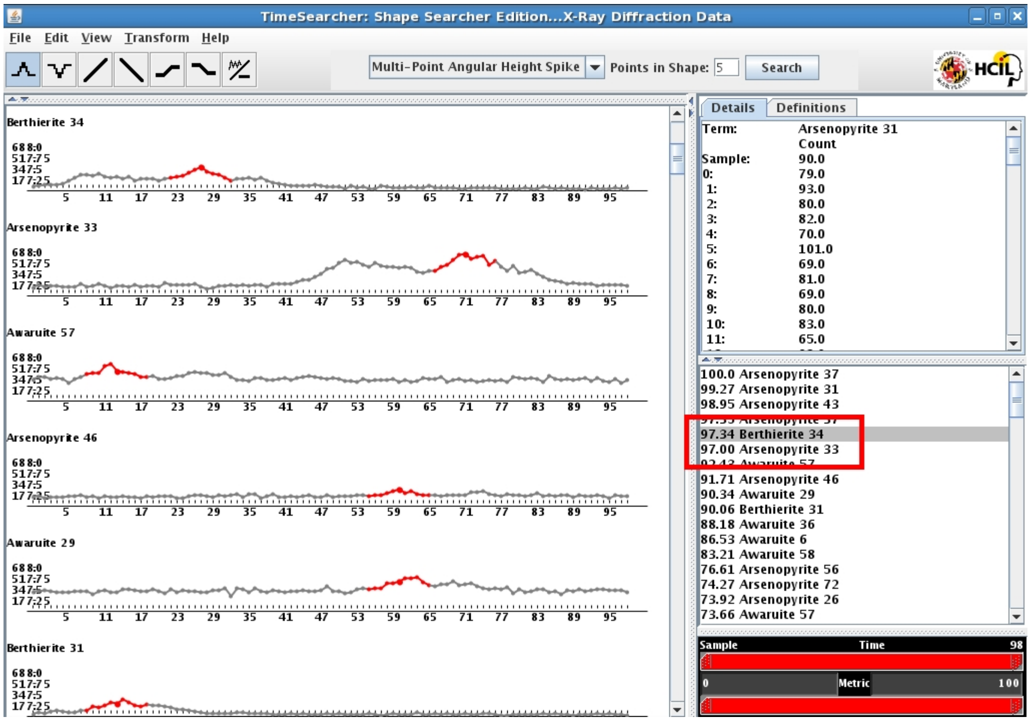
Figure 6.1: These graphs show raw powder X-ray diffraction data for Arsenopyrite (FeAsS), Berthierite ($FeSb_2S_4$), and Awaruite (Ni_3Fe). Each graph shows the intensity values for angular positions 30 – 35, while their upper left corners show the intensity readings for angular positions 5 – 85. Each of the samples contains Fe. The spikes around angular positions 33 and 34 might be caused by Fe. TimeSearcher SSE was used to attempt to identify this spike and others while exploring the data that was used to create these graphs.



Term:	Cristobalite 21 Count
Sample:	45.0
0:	39.0
1:	60.0
2:	60.0
3:	51.0
4:	62.0
5:	48.0
6:	62.0
7:	60.0
8:	60.0
9:	49.0
10:	51.0
11:	43.0

Cristobalite 21
Cristobalite 22
Cristobalite 23
Cristobalite 24
Cristobalite 25
Cristobalite 26
Cristobalite 27
Cristobalite 28
Cristobalite 29
Cristobalite 30
Cristobalite 31
Cristobalite 32
Cristobalite 33
Cristobalite 34
Cristobalite 35
Cristobalite 36
Cristobalite 37

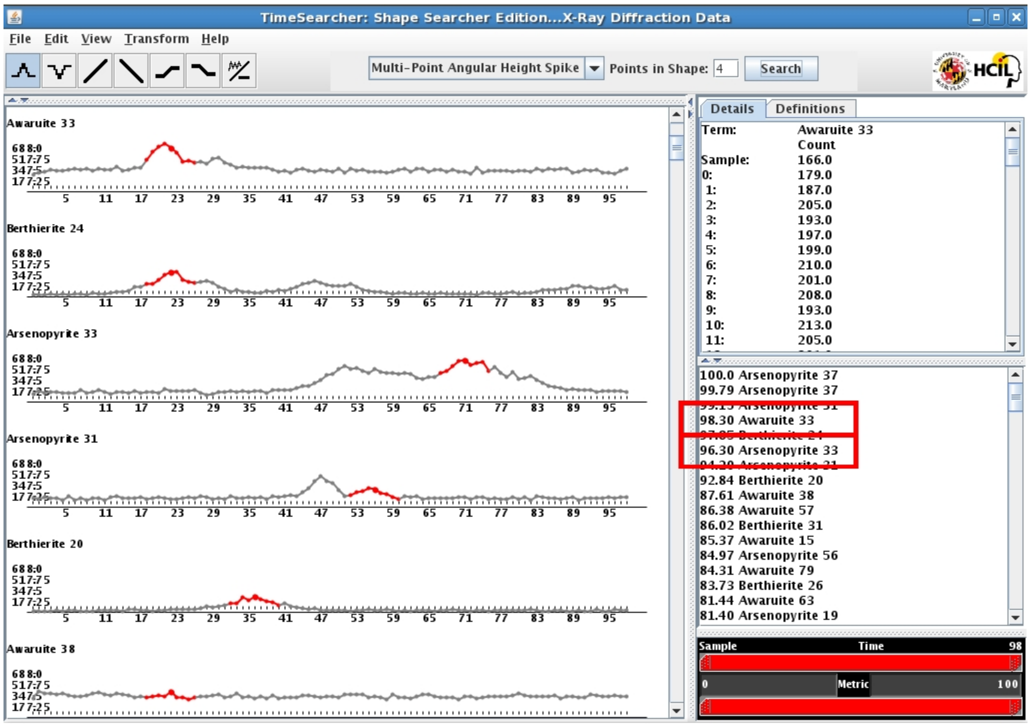
Figure 6.2: The X-ray diffraction data for a single sample was too large to read directly into TimeSearcher SSE, so the data were split into multiple time series causing some spikes to be split. The red square indicate the points at which the spike was split. The split spike could not be identified by TimeSearcher SSE.



Term:	Arsenopyrite 31
	Count
Sample:	90.0
0:	79.0
1:	93.0
2:	80.0
3:	82.0
4:	70.0
5:	101.0
6:	69.0
7:	81.0
8:	69.0
9:	80.0
10:	83.0
11:	65.0

100.0	Arsenopyrite 37
99.27	Arsenopyrite 31
98.95	Arsenopyrite 43
97.55	Arsenopyrite 37
97.34	Berthierite 34
97.00	Arsenopyrite 33
92.43	Awaruite 57
91.71	Arsenopyrite 46
90.34	Awaruite 29
90.06	Berthierite 31
88.18	Awaruite 36
86.53	Awaruite 6
83.21	Awaruite 58
76.61	Arsenopyrite 56
74.27	Arsenopyrite 72
73.92	Arsenopyrite 26
73.66	Awaruite 57

Figure 6.3: The user in the case study was able to identify spikes in Beritherite and Arsenopyrite at similar angular positions with TimeSearcher SSE. The top two graphs show the matched spikes. The element name and range of angular positions are in the upper left corner. TimeSearcher SSE was loaded with raw powder X-ray diffraction data for Beritherite, Awarite, and Arsenopyrite, materials all containing Fe. A ten point spike ranked according to its angular height was used, to identify these spikes. The angular height value is squared in red in the ranking window. This discovery indicates that spikes at this position may be caused by the presence of Fe in the materials.



Term:	Awaruite 33
	Count
Sample:	166.0
0:	179.0
1:	187.0
2:	205.0
3:	193.0
4:	197.0
5:	199.0
6:	210.0
7:	201.0
8:	208.0
9:	193.0
10:	213.0
11:	205.0

```
100.0 Arsenopyrite 37
99.79 Arsenopyrite 37
99.15 Arsenopyrite 31
98.30 Awaruite 33
97.85 Berthierite 24
96.30 Arsenopyrite 33
94.20 Arsenopyrite 31
92.84 Berthierite 20
87.61 Awaruite 38
86.38 Awaruite 57
86.02 Berthierite 31
85.37 Awaruite 15
84.97 Arsenopyrite 56
84.31 Awaruite 79
83.73 Berthierite 26
81.44 Awaruite 63
81.40 Arsenopyrite 19
```

Figure 6.4: The user in the case study was able to identify spikes in Awarite and Arsenopyrite at similar angular positions with TimeSearcher SSE. The first and third graphs show the similar spikes. The material name and range of angular positions are located in the upper left corner of the graph. TimeSearcher SSE was loaded with raw powder X-ray diffraction data for Beritherite, Awarite, and Arsenopyrite, material all containing Fe. An eight point spike ranked according to its angular height was used to identify these spikes. The angular height value is squared in red in the ranking window. This discovery indicates that spikes at this position may be caused by the presence of Fe in the materials.

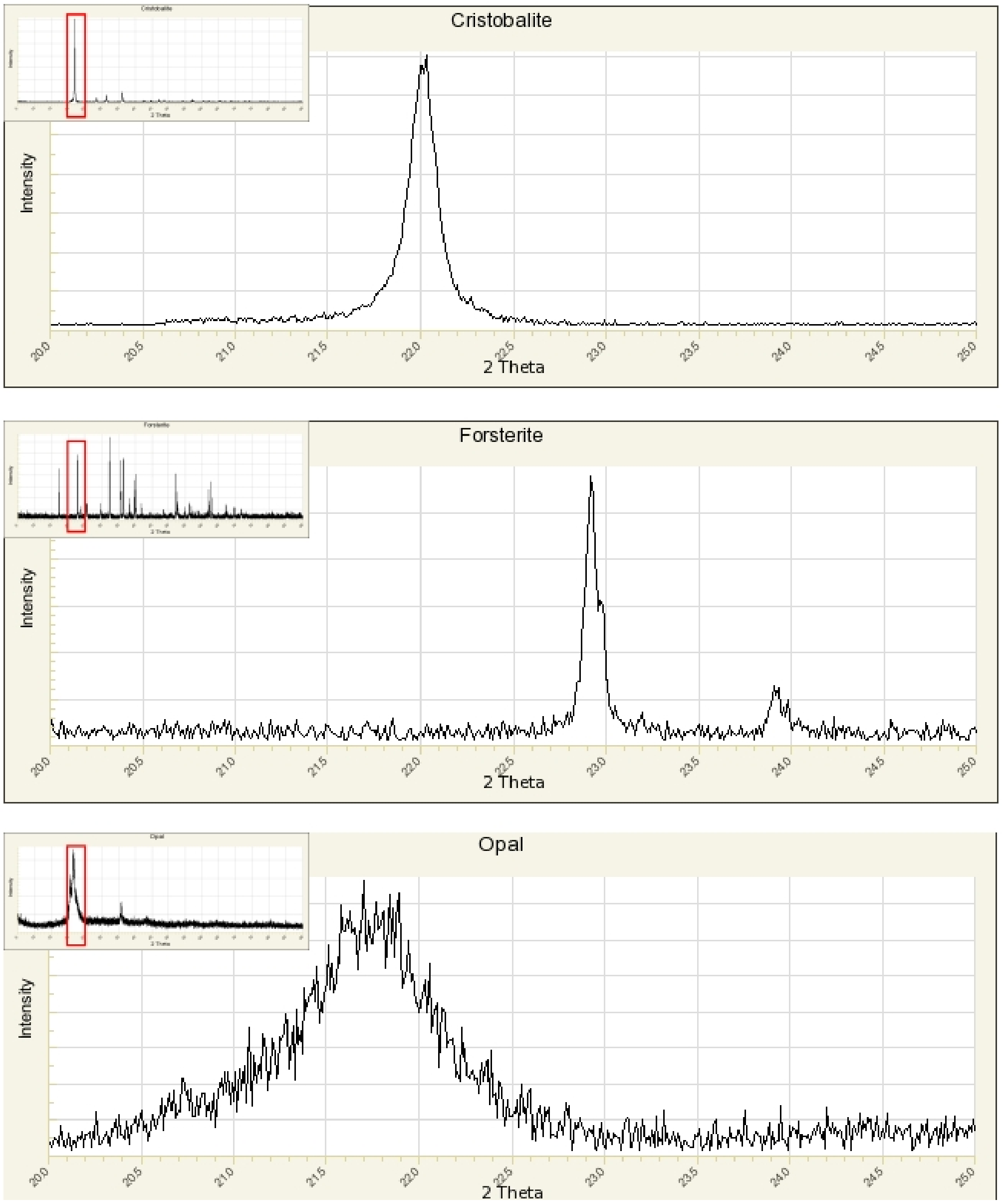


Figure 6.5: These are the graphs for the materials containing silicon,Cristobalite (SO_2), Forsterite (Mg_2SiO_4), and Opal (SiO_2nH_2O). The graphs display the angular positions 20 – 25. Each sample shares a common spike in the 22 – 23 angular position range. The upper left corner shows all the intensity readings for angular positions 5 – 85. The red square indicates where the enlarged graph came from.

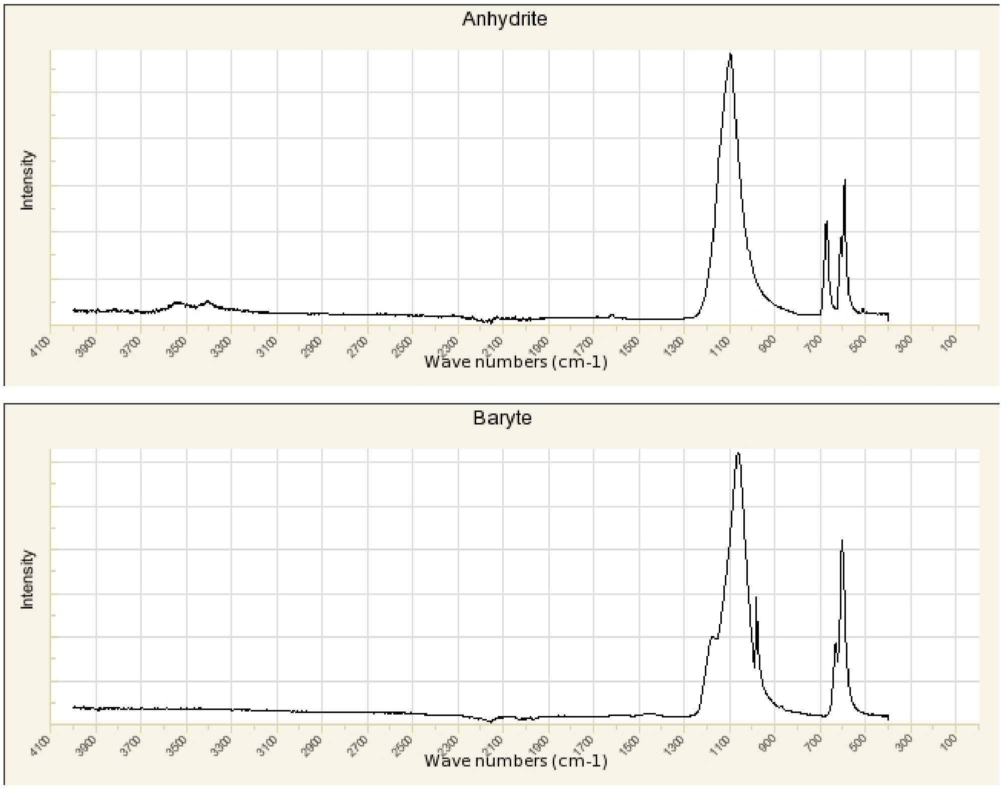
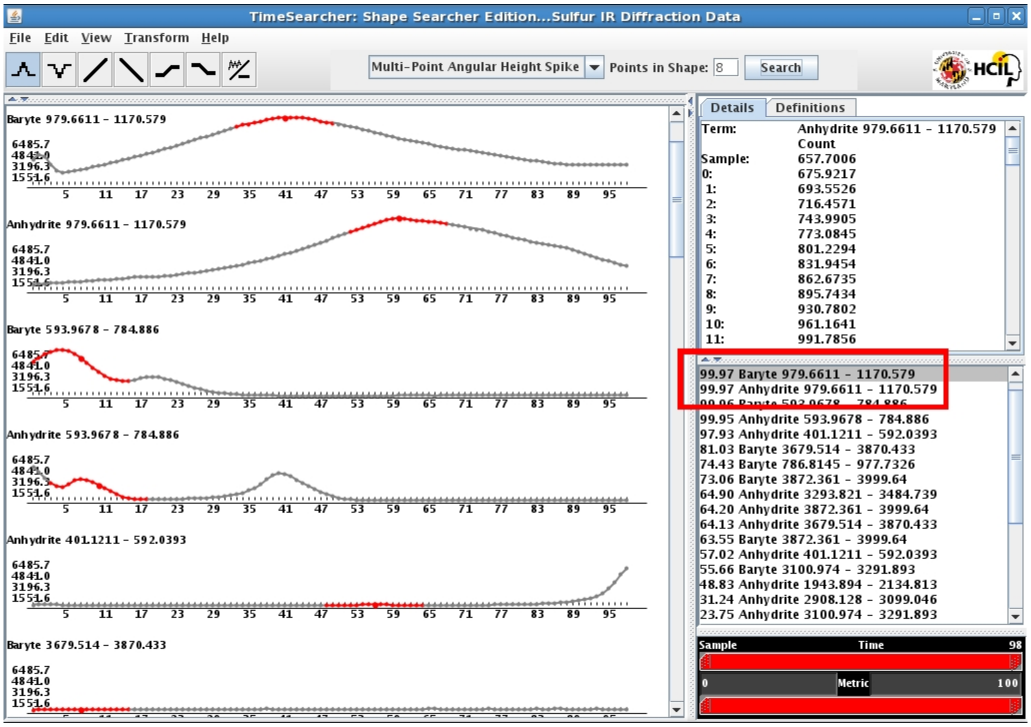


Figure 6.6: The graphs show the infrared (IR) spectroscopy data for two materials, Anhydrite ($CaSO_4$) and Baryte ($BaSO_4$, which contain sulfur, S. The data has two really clean spikes, and the data set does not contain much noise.


TimeSearcher: Shape Searcher Edition...Sulfur IR Diffraction Data
File Edit View Transform Help
Multi-Point Angular Height Spike
Points in Shape: 8
Search
HCIL
Baryte 979.6611 – 1170.579
Anhydrite 979.6611 – 1170.579
Baryte 593.9678 – 784.886
Anhydrite 593.9678 – 784.886
Anhydrite 401.1211 – 592.0393
Baryte 3679.514 – 3870.433
6485.7
4841.0
3196.3
1551.6
5 11 17 23 29 35 41 47 53 59 65 71 77 83 89 95
Details
Definitions
Sample
Time
98
0
Metric
100

Term: Anhydrite 979.6611 – 1170.579
Count
Sample: 657.7006
0: 675.9217
1: 693.5526
2: 716.4571
3: 743.9905
4: 773.0845
5: 801.2294
6: 831.9454
7: 862.6735
8: 895.7434
9: 930.7802
10: 961.1641
11: 991.7856

```
99.97 Baryte 979.6611 – 1170.579
99.97 Anhydrite 979.6611 – 1170.579
99.96 Baryte 593.9678 – 784.886
99.95 Anhydrite 593.9678 – 784.886
97.93 Anhydrite 401.1211 – 592.0393
81.03 Baryte 3679.514 – 3870.433
74.43 Baryte 786.8145 – 977.7326
73.06 Baryte 3872.361 – 3999.64
64.90 Anhydrite 3293.821 – 3484.739
64.20 Anhydrite 3872.361 – 3999.64
64.13 Anhydrite 3679.514 – 3870.433
63.55 Baryte 3872.361 – 3999.64
57.02 Anhydrite 401.1211 – 592.0393
55.66 Baryte 3100.974 – 3291.893
48.83 Anhydrite 1943.894 – 2134.813
31.24 Anhydrite 2908.128 – 3099.046
23.75 Anhydrite 3100.974 – 3291.893
```

Figure 6.7: TimeSearcher SSE was loaded with infrared spectroscopy data for Anhydrite ($CaSO_4$) and Baryte ($BaSO_4$). A user was attempting to identify spikes with similar intensity at the same wavelength. The user was able to do this with a sixteen point spike ranked according to its angular height. The values of the ranking metrics are shown in the ranking window inside the red square.

Chapter 7

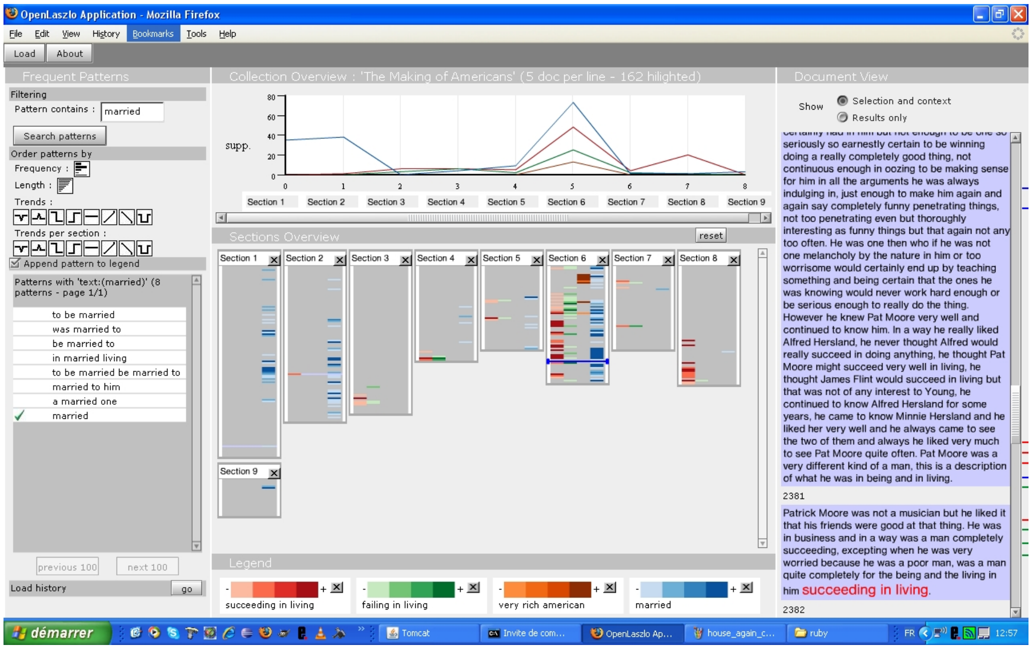
Other Examples of Shape Search

7.1 Overview

In addition to the case study described in Chapter 6 their are several other examples of shape search being used to perform analysis on several data sets. Before TimeSearcher SSE, many of the ideas about shape identification and ranking were explored through a class project called FeatureLens[10]. FeatureLens is a web based information visualization tool with the ability to do shape search as well many other things. FeatureLens was used by several people to explore word usage in the book, *The Making of Americans*, and the State of the Union Addresses George W. Bush gave during his presidency. TimeSearcher SSE was also used by the developers of the tool to explore the history of the Human and Computer Interaction (HCI) field through a database of the key words and abstracts of published papers over the past 40 years. Below are examples of the discoveries made using search in FeatureLens and TimeSearcher SSE.

7.2 FeatureLens

FeatureLens, shown in Figure 7.1, was originally designed for literary analysis. Tanya Clement, a doctoral student in the University of Maryland English Depart-



Pattern contains : married

☑ Append pattern to legend

Patterns with 'text:(married)' (8 patterns - page 1/1)

- to be married
- was married to
- be married to
- in married living
- to be married be married to
- married to him
- a married one
- ✓ married

Show ● Selection and context ○ Results only

seriously so earnestly certain to be winning doing a really completely good thing, not continuous enough in oozing to be making sense for him in all the arguments he was always indulging in, just enough to make him again and again say completely funny penetrating things, not too penetrating even but thoroughly interesting as funny things but that again not any too often. He was one then who if he was not one melancholy by the nature in him or too worrisome would certainly end up by teaching something and being certain that the ones he was knowing would never work hard enough or be serious enough to really do the thing. However he knew Pat Moore very well and continued to know him. In a way he really liked Alfred Hersland, he never thought Alfred would really succeed in doing anything, he thought Pat Moore might succeed very well in living, he thought James Flint would succeed in living but that was not of any interest to Young, he continued to know Alfred Hersland for some years, he came to know Minnie Hersland and he liked her very well and he always came to see the two of them and always he liked very much to see Pat Moore quite often. Pat Moore was a very different kind of a man, this is a description of what he was in being and in living.

2381

Patrick Moore was not a musician but he liked it that his friends were good at that thing. He was in business and in a way was a man completely succeeding, excepting when he was very worried because he was a poor man, was a man quite completely for the being and the living in him succeeding in living.

2382

Figure 7.1: FeatureLens. The left panel contains a set of controls used to search for a shape within the collection. When a button is selected, the ranked shapes appear in a list below the button panel. The center panel shows the distribution of frequency of a particular word or n-gram across the collection. The right panel shows one occurrence of a word or n-gram in the context where it appears in the text.

ment, was the primary user of FeatureLens. She was also very influential in the design of FeatureLens. Her work deals with the study of *The Making of Americans* by Gertrude Stein, which consists of 517,207 words - 5,329 of them unique. In comparison, Herman Melville's Moby Dick consists of only 220,254 word - 14,512 of them unique. Stein's extensive use of repetitions renders *The Making of Americans* one of the most difficult books to read and interpret. Literary scholars are developing interpretive hypotheses about the purpose of this text's repetitions. Using FeatureLens for this analysis provides invaluable insights to the benefits and limitations of

FeatureLens and shape identification and ranking. The tool allowed the user to find new ideas, appearing in a random manner across the book, which were meaningful to someone who is acquainted with the book. One of the findings that shape identification allowed is depicted in Figure 7.1. It showed that the chapters associated with domestic terms were also the ones where the only female child appeared, whereas these terms did not appear in the chapters associated with the other two male children. The tool also allowed the discovery of the usage of the concepts of failure and success in the book, which appeared not to be associated with business but with marriage. This is shown by the n-grams, “succeeding in living,” “failing in living,” “very rich American,” and “married”, all having spikes in Chapter 5 that are highly ranked.

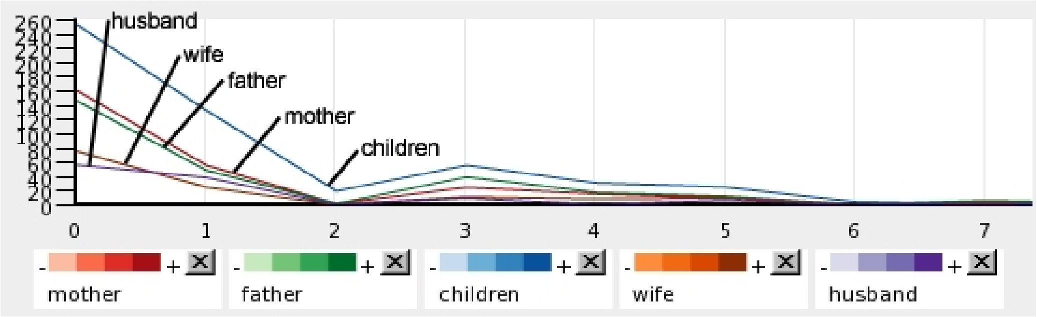


Figure 7.2: Examples of decreasing slope from *The Making of Americans.* The decreasing trend suggests that as the book progress the topic of family decreases.

Figure 7.2 shows several n-grams whose time series contain decreasing lines from *The Making of Americans.* The n-grams are “husband,” “wife,” “father,” “mother,” and “children.” All these terms refer to family, so the conclusion was made that the family diminishes as a topic as the book progresses. These terms were found by ranking decreasing line shapes by their slope.

In addition to performing analysis on the Making of Americans book FeatureLens was used to analyze the State of the Union addresses that were given during George W. Bush's presidency. A study was conducted with eight participants, who had an advanced degree or were pursuing an advanced degree. As they used the tool they were asked to comment aloud. The study consisted of a fact finding session where the users were asked to answer specific questions using the FeatureLens tool and a free exploration session. The fact finding session was focused on making sure the users were able to use the tool and understood the different portions of the screen. The interesting results came when the users were able to freely explore the data set. They were able to find several trends using the shape search feature of FeatureLens. Figure 7.3 shows a steady usage of the words, budget, citizens, america, and world. Figure 7.4 show a spike in the usage of freedom, good, war, and terror in 2002. This is likely due to the start of the war on terrorism following 9/11.

Although FeatureLens is not as robust as TimeSearcher SSE, a user was able to use the tool's shape identification and ranking ability to make analytical discoveries in a temporal data set of interest. The ranking feature was used to identify n-grams that had similar shapes; upon further inspection, the n-grams were found to be related.

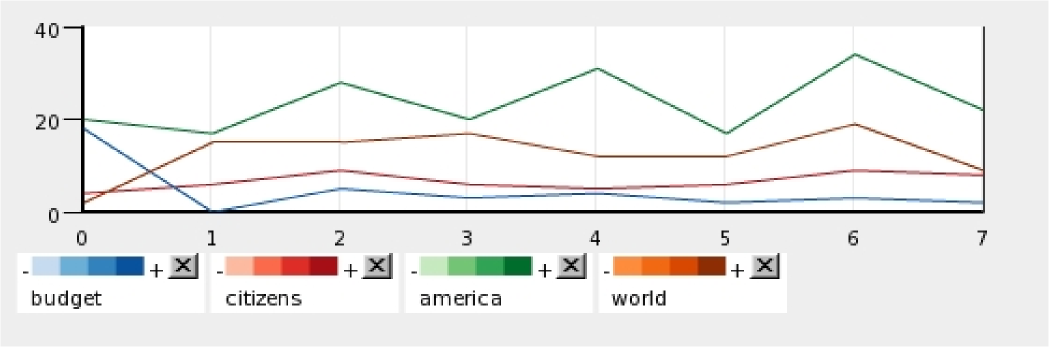


Figure 7.3: This figure shows the word who usage remained constant throughout all of the State of the Union addresses given by George W. Bush. The frequency of the usage of the words, “budget”, “citizens”, “america”, and “world” remained constant throughout George W. Bush’s presidency.

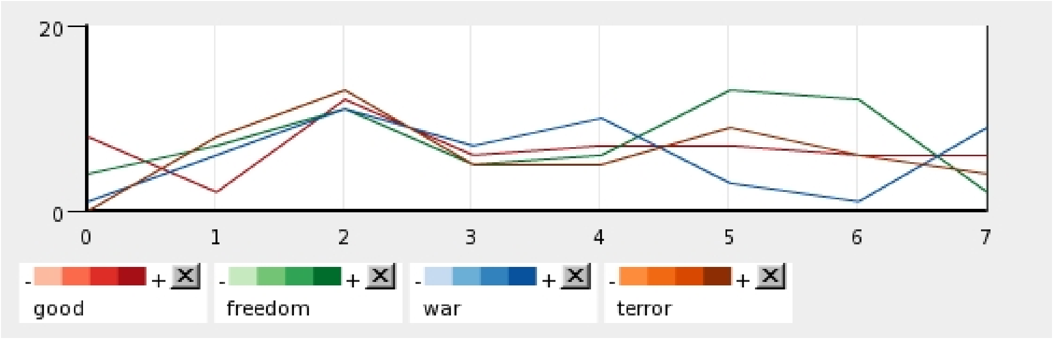


Figure 7.4: This figure shows a correlated spike in the usage of the words, “good”, “freedom”, “war”, and “terror” following the terrorist attacks that occur September 11, 2001.

7.3 TimeSearcher SSE

As TimeSearcher SSE was being developed different data sets were loaded into the tool to test its functionality. As the data sets were explored some interesting discoveries were made. One of the data sets contained the frequency of occurrences of “interesting” words in the abstracts of published papers in the Human and Computer

Interaction (HCI) field over the last 40 years. This data was obtained by crawling the HCI bibliography website, http://hcibib.org. An interesting discovery was made while exploring this data set. In 2003 there was a significant relative spike in the use of the word "universal" as shown in Figure 7.5. It was weird that a topic would occur frequently and then practically disappear. Upon further inspection it was determine that most of the abstracts were from the journal *Universal Access in the Information Society* which published over 300 papers that year, more than 250 more than any other year. Figure 7.6 shows the increasing trend of abstracts containing the word "web."

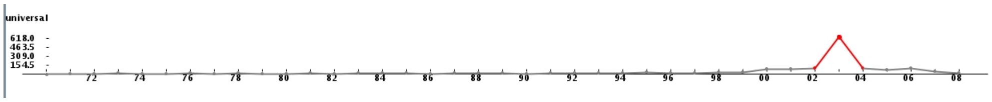


Figure 7.5: This figure shows a TimeSearcher SSE plot of the number of abstracts that contain the word "universal." The spike was caused by collecting data from the Human and Computer Interaction International conference is that was only collected that year. TimeSearcher SSE is well suited for finding anomalies in data.

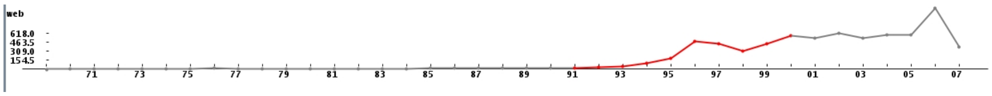


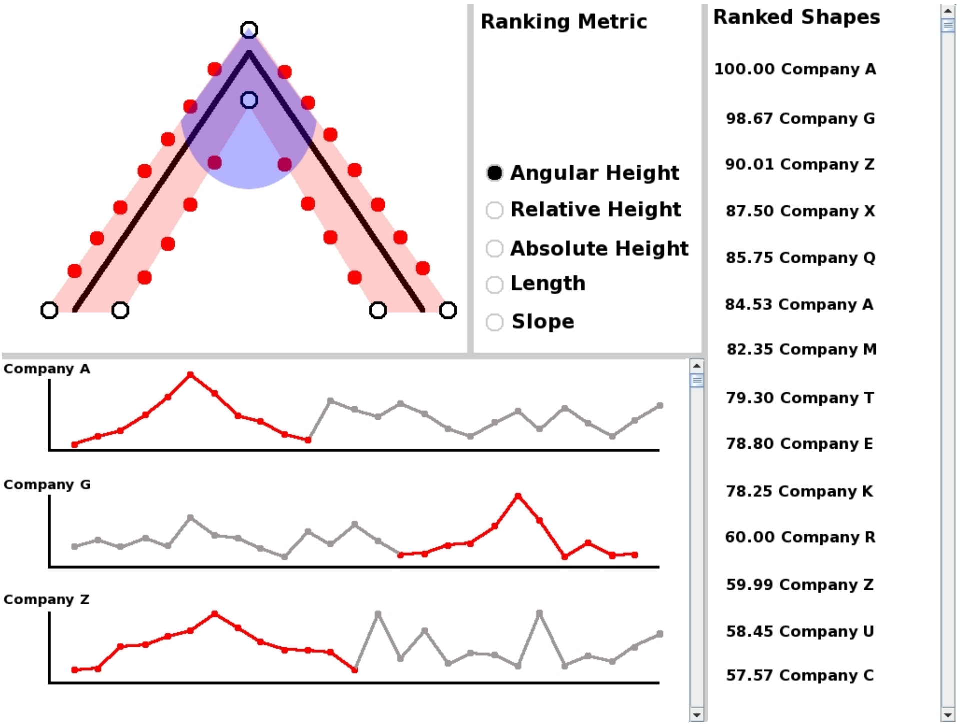
Figure 7.6: This figure shows an increase in HCI papers relating to the web.

Chapter 7

Future Work

Extending TimeSearcher 1 to create TimeSearcher SSE allowed for quick prototyping of a tool capable of exploring different types of definitions of shapes and ranking them according to a metric of interest. But the tool was not meant for shape identification and ranking, so the ability to dynamically define and rank shapes is not present. Figure 7.1 shows a mock up of an interface very similar to TimeSearcher 1, but it is better suited for shape defining, ranking, and helping users to explore shapes and their definitions, more intuitively. The proposed interface will be able to adhere more closely to the information visualization mantra of "overview first, zoom and filter, then details-on-demand[25]."

The interface contains four primary windows, as shown in Figure 7.1. The ranking window on the right side of the screen and the shapes window on the lower left side are identical to the windows in TimeSearcher SSE. The definition and ranking metrics windows are new, allowing users to dynamically define shapes of interest and choose the metrics by which the shapes are ranked. The shape definition window is located in the top left hand corner, and the ranking metric selection window is located in the top center. The ranking metric window allows the user to select how the shape will be ranked. The upper portion of this window will display a simple text definition for the ranking metric. The attribute or attributes


Ranking Metric

● Angular Height
○ Relative Height
○ Absolute Height
○ Length
○ Slope

Ranked Shapes

100.00 Company A
98.67 Company G
90.01 Company Z
87.50 Company X
85.75 Company Q
84.53 Company A
82.35 Company M
79.30 Company T
78.80 Company E
78.25 Company K
60.00 Company R
59.99 Company Z
58.45 Company U
57.57 Company C

Figure 7.1: A mock up of a new interface for dynamically creating shape definition.

that are measured by the ranking metric are highlighted in blue on the shape in the shape definition window.

The shape definition window is the primary window in the interface. It allows the user to dynamically define a particular shape of interest. This is important because users may not know exactly what behavior they are interested in or may not understand the attributes associated with a particular shape. The shape definition window allows a user to explore the shape's attributes by directly manipulating the shape and dynamically creating definitions. The definitions are created by dragging the open circles to define ranges of values that a particular attribute can take on. Figure 7.2 shows three examples of the shape definition window, in addition to the window shown in Figure 7.1.

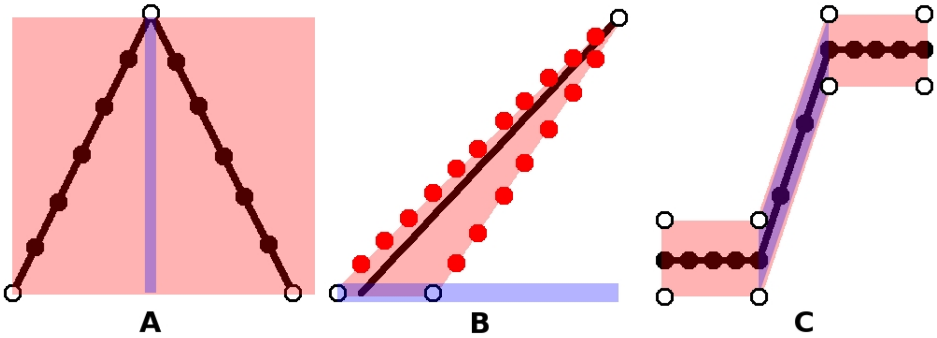


Figure 7.2: These are three examples of how a shape's definition can be dynamically created. Figure A is a spike shape. The red box indicates the values that the height and slope attributes can take. The black dots indicate that the shape is a 13 point spike. The blue rectangle in the center of the spike indicates that the spike will be ranked according to its height. Figure B is a multiple point increasing line with a range of values for the slope and length attributes, as indicated by the red triangle and dots. The blue rectangle across the bottom indicates this shape will be ranked according to its length. Finally, Figure C shows a rise shape. The number of points in the stable periods is shown by the black dots. The red transparent rectangles constrain the values of the stable and change periods, and the blue rhombus identifies the slope as the ranking metric.

The shape definition window in Figure 7.1 shows a spike shape. The transparent quarter circle indicates that the identified shapes are ranked according to the angular height. The red transparent spike shape defines a range of values that several attributes can take. The slope, length, and angular height are all constrained. Some attributes may change independently of the others, while other attributes directly affect each other. For example, dragging the left bottom edge between open circles along the slope (as shown by arrow A in Figure 7.3) will increase length, without

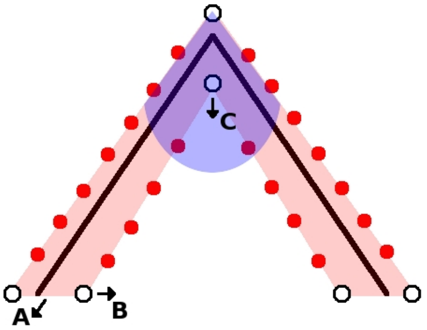


Figure 7.3: This is an example of how a spike shape can be manipulated to dynamically create shape definitions.

changing the range of slope values. On the other hand, dragging the open circle on the left side of the bottom left edge to the right (as shown by arrow B in Figure 7.3) will change the ranges for the length and slope attributes. By dragging the open circles and the edges between the circles, shape definitions can be dynamically created and shapes within the data set explored.

Figure 7.2 has three shape definitions. Figure 7.2A shows a six point spike with no defined value ranges for the slope and angle. The shapes are ranked according to the height. The transparent blue box in the center gives no indication of how the height will be measured, only that it will be measured. The ranking metric selection window shows how the height will be measured. The same is true for the slope. The user will specify a slope definition through a preferences menu. Figure 7.2B shows an increasing line of variable length and slope that is ranked according to the length. Figure 7.2C is a rise shape with stable periods containing five time points and a change period of two points. The range of values for each attribute is specified by the red transparent box. The rise shape in Figure 7.2C is ranked according to its slope.

Chapter 9

Conclusion

In this thesis, a set of common shapes were examined and their attributes defined. Each shape defines a behavior; the attributes define an aspect of that behavior. The attributes provide an expressive way of defining a shape. By ranking the shapes according to an attribute of interest, users can tailor the results of the shape identification process to the shapes that interest them. Lines, spikes, sinks, rises, drops, plateaus, valleys, and gaps where discussed.

Research has examined useful ways of defining and identifying shapes, as well as evaluating the usefulness. SDL provides an expressive language for describing shapes; TimeSearcher 1 and 2 perform shape definition and identification via graphical widgets. They are both powerful in their own right, but they do not help users to understand how the identified shapes relate to the original definition, nor how they compare to each other. Pattern discovery provides a starting point to understanding how a shape can be evaluated and compared to other shapes. This thesis presented ideas that combine shape definition and evaluation, by examining the attributes that characterize a shape.

Lines are simple shapes that consist of a set of line segments. Lines are used to describe a consistent behavior. The attributes associated with line shapes are slope and length. Slope is the measurement of the general direction values are going. For example, increasing values would be indicated by a positive slope and decreasing values by a negative slope. The length of a line shape is a measurement of the duration of the behavior

A spike or sink suggests a significant, but temporary change in value. Three height attributes for spikes and sinks were identified. Angular height identifies how much the values changed in each direction, while relative height measures how different the behavior is from the rest of the points in the time series. In addition to the height attributes, the slope of edges that make up spike or sink shapes are attributes that measure the rate of change. The absolute height is the value of the peak point. The number of points identifies the duration of the spikes.

Rises and drops identify a sustained change in value. These shapes consist of three periods: a leading stable period, a period of change, and a trailing stable period. The stable periods can be characterized by the attributes of average value, while the period of change is described by its slope. Both of the periods share a common attribute of length, which is a measurement of the duration of the shape.

Plateaus, valleys, and gaps consist of five distinct periods, three stable periods, and two periods of change. The leading stable period is followed by departing period of change. The intermediate stable period separates these shapes from spikes and sinks. The intermediate stable period is followed by the returning period of change and the trailing stable period. The attributes that were identified for drops and rises are the same for plateaus, valleys, and gaps, except they have a different meaning. In addition to those attributes, differences in the average value of the leading and/or trailing period and the intermediate period may be used to identify a shape of interest.

Examples of these shape definitions were incorporated into TimeSearcher to create TimeSearcher SSE. Several attributes were chosen to rank the shapes identified by TimeSearcher SSE. A case study was used to understand the usefulness of TimeSearcher SSE and its method of shape identification and ranking, as well as its limitations. A researcher used the tool to identify spikes of the same intensities and angular position in X-ray diffraction data. The tool successfully performed the task in one of the data sets, but it was limited by inflexibility in the other tasks.

The user case study pointed out some limitations of the TimeSearcher SSE interface that prevent exploration of the true power of this method of shape identification and ranking. To overcome these limitations, this thesis put forth a new unique interface to allow users to dynamically create shape definitions and rank the identified shapes according to an attribute of their choice.

Bibliography

[1] R Agrawal, G Psaila, E L Wimmers, and M Zaot. Querying shapes of histories. In *Proceeding 21st International Conference on Very Large Databases*, pages 502–514. Morgan Kaufmann Publishers, Inc, 1995.

[2] H Andre-Jonsson and D Badal. Using signature files for querying time-series data. In *First European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 211–220, 1997.

[3] K Balog, G Mishne, and M Rijke. Rijke. why are they excited? identifying and explaining spikes in blog mood levels. In *Proceedings 11th Meeting of the European Chapter of the Association for Computational Linguistics*, 2006.

[4] B Bederson, J Grosjean, and J Meyer. Toolkit design for interactive structured graphics. *IEEE Transactions on Software Engineering*, 30:535–546, 2004.

[5] L Berry and T Munzner. Binx: Dynamic exploration of time series datasets across aggregation levels. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 215–217, 2004.

[6] P Buono, A Aris, C Plaisant, A Khella, and B Shneiderman. Interactive pattern search in time series. In *Proceeding of the Visualization and Data Analysis Conference*, pages 175–185, 2005.

[7] G Chen, J Cho, and M Hansen. On the brink:searching for drops in sensor data. In *Proceeding of the 11th International Conference on Extending Database Technology*, 2008.

[8] G Das, K Lin, H Mannila, G Renganathan, and P Smyth. Rule discovery from time series. In *Proceeding of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1998.

[9] H Dettki and G Ericsson. Screening radiolocation datasets for movement strategies with time series segmentation. *Journal of Wildlife Management*, 72:535–542, 2008.

[10] A Don, E Zheleva, M Gregory, S Tarkan, L Auvil, T Clement, B Shneiderman, and C Plaisant. Discovering interesting usage patterns in text collections: Integrating text mining with visualization. In *Proceeding of the 16th Conference on Information and Knowledge Management*, pages 213–222. ACM Press, New York, 2007.

[11] M Dubinko, R Kumar, J Magnani, J Novak, P Raghavan, and A Tomkins. Visualizing tags over time. In *Proceeding of the 15th International WWW Conference*, 2006.

[12] J A Fails, A Karlson, L Shahamat, and B Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *in Proceeding of the IEEE Symposium on Visual Analytics Science and Technology*, pages 167–174. IEEE Press, Piscataway, NJ, 2006.

[13] Featurelens website. http://www.cs.umd.edu/hcil/textvis/featurelens/.

[14] T C Fu, F L Chung, V Ng, and R Luk. Pattern discovery for stock time series using self-organizing maps. In *Workshop on Temporal Data Mining, 7th International Conference on Knowledge Discovery and Data Mining*, pages 27–37. ACM Press, 2001.

[15] M N Garofalakis, R Rastogi, and Shim. Spirit: Sequential pattern mining with regular expression constraints. In *Proceeding of the 25th International Conference on Very Large Databases*, pages 223–234, 1999.

[16] V Guralnik and J Srivastava. Event detection from time series data. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pages 33–42, 1999.

[17] J Han, G Dong, and Y Yin. Efficient mining of partial periodic patterns in time series database. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 214–218. AAAI Press, 1998.

[18] H Hochheiser. *Visual Queries for finding patterns in time series data.* PhD thesis, University of Maryland Computer Science Dept, 2002.

[19] E Keogh, H Hochheiser, and B Shneiderman. An augmented visual query mechanism for finding patterns in time series data. In *Proceedings of the 5th International Conference on Flexible Query Answering Systems*, pages 240–250. Springer, LNAI, 2002.

[20] J Lin, E Keogh, S Lonardi, J Lankford, and D Nystrom. Viztree: Visually mining and monitoring massive time series. In *Proceeding of the 10th International Conference on Knowledge Discovery and Data Mining*, pages 460–469. ACM Press, 2004.

[21] B Padmanabhan and A Tuzhilin. Pattern discovery in temporal databases: A temporal logic approach. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[22] Rruff project database. http://rruff.info/.

[23] Kathy Ryall, Neal Lesh, Hiroaki Miyashita, Shigeru Makino, Tom Lanning, Tom Lanning, Darren Leigh, and Darren Leigh. Querylines: approximate query for visual browsing. In *Extended Abstracts of the Conference on Human Factors in Computing Systems*, pages 1765–1768. ACM Press, 2005.

[24] J Seo and B Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *Proceeding of the IEEE Symposium on Information Visualization*, pages 65–72. IEEE Press, 2004.

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

[26] A Silberschatz and A Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Knowledge Discovery and Data Mining*, pages 275–281, 1995.

[27] Timesearcher website. http://www.cs.umd.edu/hcil/timesearcher/.

[28] Value line website. http://www.valueline.com/.

[29] H Wang, W Wang, J Yang, and P Yu. Clustering by pattern similarity in large data sets, 2002.

[30] M Wattenberg. Sketching a graph to query a time series database. In *Proceedings of the 2001 Conference Human Factors in Computing Systems, Extended Abstracts*, pages 381–382. ACM Press, 2001.

[31] J Yang, W Wang, and P S Yu. Stamp: Discovery of statistically important pattern repeats in a long sequence. In *Proceedings of the 3rd SIAM International Conference on Data Mining*, pages 224–238. SIAM, 2003.