Interactive Pattern Search in Time Series

Paolo Buono[a], Aleks Aris[bc], Catherine Plaisant[b], Amir Khella[bc], Ben Shneiderman[bc*]

[a] University of Bari, Bari, Italy;
[b]Human-Computer Interaction Laboratory, [c]Dept of Computer Science, University of Maryland, College Park, USA

ABSTRACT

The need for pattern discovery in long time series data led researchers to develop algorithms for similarity search. Most of the literature about time series focuses on algorithms that index time series and bring the data into the main storage, thus providing fast information retrieval on large time series. This paper reviews the state of the art in visualizing time series, and focuses on techniques that enable users to interactively query time series. Then it presents TimeSearcher 2, a tool that enables users to explore multidimensional data using coordinated tables and graphs with overview+detail, filter the time series data to reduce the scope of the search, select an existing pattern to find similar occurrences, and interactively adjust similarity parameters to narrow the result set. This tool is an extension of previous work, TimeSearcher 1, which uses graphical timeboxes to interactively query time series data.

Keywords: Time series, pattern search, visual interaction, information visualization, dynamic queries

1. INTRODUCTION

Time series are widely used in applications such as electrocardiograms (EKGs), seismographs, industrial processes, meteorology, and sound recordings. They consist of sequences of real numbers, representing the measurements or observations of a real variable at equal time intervals. From an algorithmic perspective, there is a long history on time series, originally grounded on statistical analysis. Today, with the aid of computers, users can analyze time series data using classical statistical models, and also explore their data using visualization tools. These interfaces enable users to see the data and apply their powerful perceptual abilities to identify trends or spot anomalies.

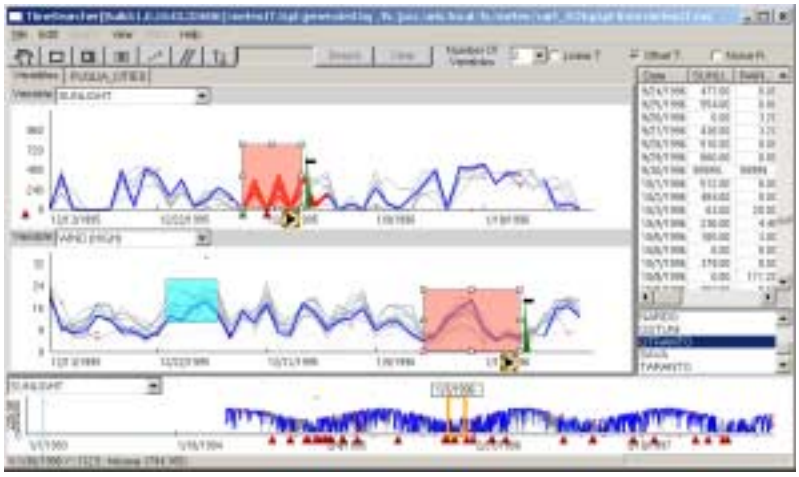

Figure 1: TimeSearcher 2

* Further author information: (Send correspondence to Paolo Buono)
P. Buono: E-mail: buono@di.uniba.it, Telephone: +390805442239
A. Aris: E-mail: aris@cs.umd.edu, Telephone:
Catherine Plaisant: E-mail: plaisant@cs.umd.edu, Telephone: +1-301-405-2768
Amir Khella: E-mail: plaisant@cs.umd.edu
Ben Shneiderman: E-mail: plaisant@cs.umd.edu, Telephone: +1-301-495-2680

The work presented in this paper (Figure 1) builds on previous work at the University of Maryland that explored the use of timeboxes to query time series data (Figure 2). Timeboxes are rectangular regions that are selected and directly manipulated on a timeline overview of the data. The boundary values of the timeboxes specify the relevant parameters of the query. This paper enhances the concept of timeboxes by differentiating two types of timeboxes. The first type (the original) is used to filter the data and reduce the scope of the search, whereas the second type is used to perform a specific pattern search anywhere in the remaining data. We also present enhancements to the general browsing interface of TimeSearcher, allowing users to deal with long time series of multiple heterogeneous variables. Our target users may be expert or intermittent users but our interfaces do not require any specialized analysis skills such as statistical knowledge. We aim to provide interfaces that extend the initial exploratory analysis. Users first gain an overview of the data, then filter and zoom on the data to spot patterns of interest. Next, a search can be performed to find similar patterns at other times and in other series helping users to make hypothesis about phenomena in the data (experts may use external tools that validate the hypothesis).

After a review of the literature, in Section 2, we describe the interface and propose a three-step interaction framework for interactive pattern search in time series in Section 3; then limitations and possible improvements are discussed in Section 4 and conclusions are given in Section 5.

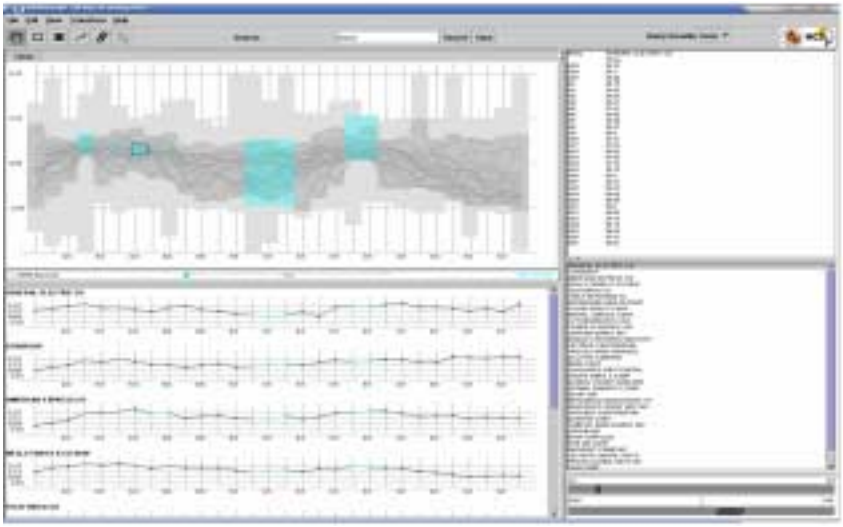

Figure 2: TimeSearcher 1

2. RELATED WORK

The characteristics of time series vary widely and are highly dependent on the application domain. Even data not originally thought of as time series can be transformed into time series. Examples include the mapping of a video stream as a time series, or the mapping of the human movement in 3D space to a time series (each coordinate becomes a single time series)[1]. There is an abundant literature focused on time series data. Some tools deal with a single time series and may look for patterns within that series[2], others deal with multiple time series and allow users to find the times series of interest[3]. Very few allow multiple variables to be browsed and searched[4]. Several papers propose the visualizations of time series but allow only limited interaction with the data, such as altering the order or accessing details. Interactive search facilities are rare and offer limited features[5,6,7]. Another direction is to provide predictions with visual results but the interaction is limited[8]. This section focuses on tools that include a graphical user interface allowing users to interact with the data by using simple widgets and without the need of extensive training or specialized skills such as statistical analysis expertise.

One of the first attempts to visualize and interact with time series was Diamond Fast[4]. Diamond Fast visualizes time series on the screen and permits users to move and resize them, hence compare more than two time series. It also includes some management of missing values. Diamond Fast was only capable of managing short time series which is not enough today, but remains an important inspiration for new tools and a good application in several domains. Other

newer tools like in ILOG[9] and in Personal Stock Monitor[10] permit a high level of interaction to browse the data with enhanced zoom features but they lack search capabilities and are limited to the visualization of a single time series. Brodbeck and Girardin[11] offer a semantic zoom implementation and the possibility to visualize very long time series. Tools that allow users to search for patterns vary in the way they let users specify the pattern, adjust the search algorithm, and browse the results. Some tools allow users to discover patterns interactively. For example, Carlis and Konstan[2] shows that by using a simple interaction technique (tightening or relaxing a spiral view) users can visually reveal periodic patterns in serial periodic data without the need of running specific search algorithms.

Timesearcher 1[3,12] allowed users to specify patterns by concatenating multiple boxes together to form a pattern at a particular time in the series. The series are filtered to show only those that pass through that set of time-specific Timeboxes. It includes useful tools to interactively search by slope, or permit the specification of queries that allow a range specification in the time axis, but there is no mechanism to search for a pattern occurring at different times in the dataset. Timesearcher 1 also started to explore the specification of queries on multiple variables and we continued this work.

Other tools permit users to specify a pattern of interest and then to see search results with similar patterns. Choratas[13] requires users to specify numerical parameters of the pattern, VizTree[14] asks users to specify the shape of the pattern by dividing the pattern into segments and specifying whether each segment belongs to a specific range. QuerySketch[15] allows users to directly sketch the shape of the pattern.

An interesting contribution is IPBC[16], that presents a 3D tool that allows users to select a pattern of interest in the data itself, and then initiate a search for similar patterns. Results are highlighted showing where the similar patterns are in the dataset. This tool is customized to display time series that have periodic characteristics (e.g. hours, days, weeks, months, etc.).

Visualization combined with interaction may be useful to generate hypotheses and to confirm analysis done using statistical algorithms. We believe that allowing users to select a pattern from the data itself is particularly useful in the exploratory phase of the analysis. During early data exploration, users do not know what pattern they may want to look for. They browse the data and when they see some anomalies or surprising shapes in the data they can zoom in, select the pattern, and then start the search for similar patterns, enabling them to see where and when similar behavior occurred. Of course, allowing multiple pattern specification methods is an even more powerful approach (see section 4). No tool that we know of other than TimeSearcher allows the specification of multiple patterns on multiple variables.

3. DESCRIPTION OF THE INTERFACE

TimeSearcher 1's basic browsing capability was extended to include multiple heterogeneous variables and tens of thousands of time points. In addition, TimeSearcher 2's new search interface combines both filter and pattern search capability, implementing a three-step approach that can be extended to a variety of time series search interfaces. Our work was primarily guided by petroleum industry production data (oil and gas well data) and meteorology data. The examples emphasize the meteorology data, as it is easier to explain.

3.1 Multivariable time series viewer

For exploratory data analysis users need an application that offers an overview of the time series and the possibility to zoom in and out. We found that in some cases the data had never been represented visually before and users were eager to explore their data first to see patterns, make hypotheses, understand outliers, and recognize that some data were missing.

TimeSearcher 2 allows up to eight multiple heterogeneous variables to be shown at once. Figure 3 shows an example using the meteorology data that refers to cities of Puglia, a region of Southern Italy. Three variables are shown: amount of sunlight, rainfall and average temperature, for a set of items – in this case Italian cities. On the right hand side, a scrollable table shows the numerical values and the list of cities. Approximately 5 years of data is shown and the seasonal patterns of most variables are visible. Users can highlight a specific time point in all variables by clicking on the background to produce with a light blue vertical line and a highlight in the table (upper right). Figure 4 shows all eight variables for all the items – which are drawn overlapped. Figure 5 provides an alternate view showing separated views of individual items (cities) for a single variable.

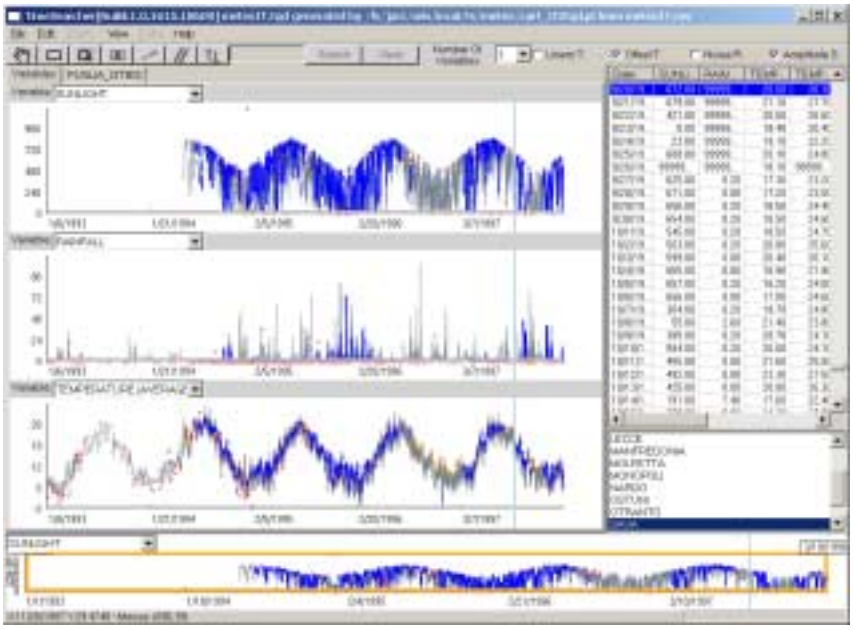

Figure 3: TimeSearcher 2 main data-browsing interface.

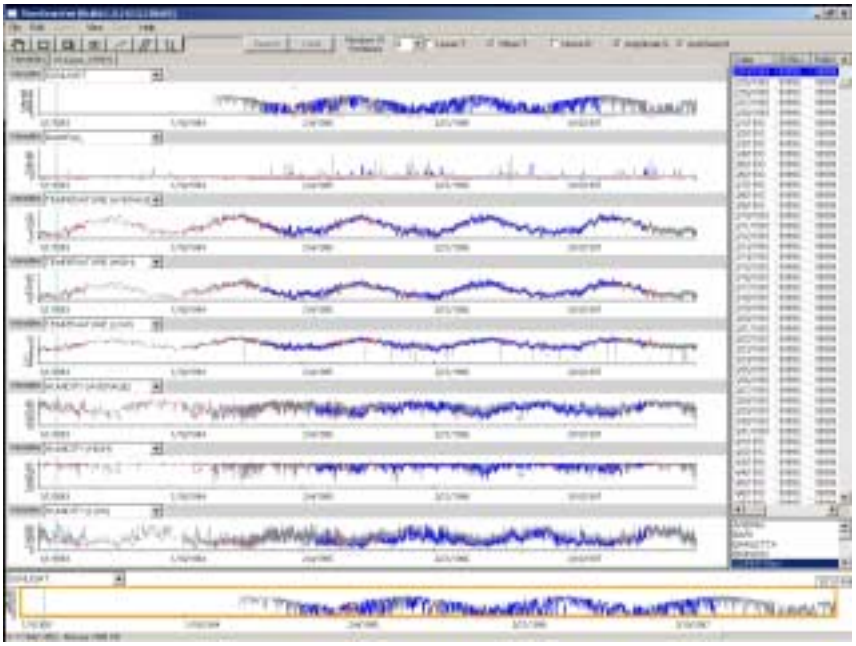

Figure 4: Many variables can be shown at once, here 8 variables over 5 years, for a complete overview of the data available for a single city.

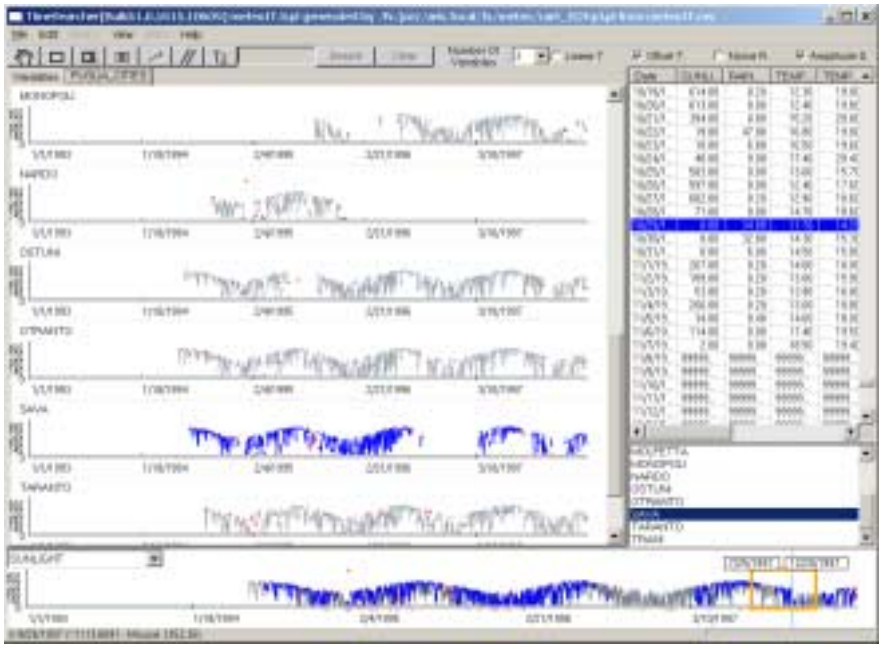

Figure 5: This view clearly shows the periods where data is missing. Missing values are indicated with red circles. All items are ordinarily shown together overlapping. Alternatively, items (here cities) can be shown on separate lines for a single variable (here amount of sunlight).

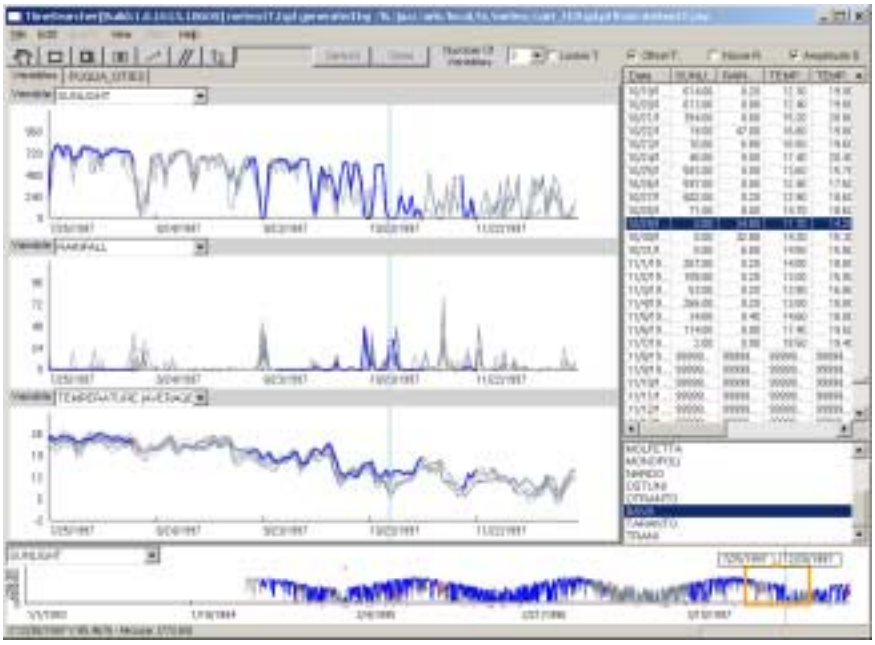

Figure 6: To see more details users can zoom on the timeline by narrowing the orange field-of-view box drawn on the overview shown at the bottom of the screen. The overview represents the complete 5 years of data for one variable, while the detail view is zoomed on the second half of 1997.

The use of a detail+overview browser to provide access to details is shown in Figure 6. Users may want to look at the second half of 1997, which they can achieve by moving and resizing the orange field of view box on the bottom (also delimited by dates), as it determines the range of time interval displayed on the screen. Users often need to compare

different time periods, so the detail view can be split into 2 panes each showing a different time period, controlled by independent field of view boxes in the overview[17]. Users can use the rapid browsing capabilities of Timesearcher 2 to monitor data. They can visually spot interesting patterns, and quickly browse recent data in search of other instances of the pattern. Nevertheless, this process becomes cumbersome when looking at extensive archives of data, and search capabilities become necessary. They are described in the next section.

3.2 Three-step interactive search

One of the innovations in Timesearcher 2 is the capability to perform pattern search. The first goal was to create a simple widget that enables searching for a pattern in the time series. Once the pattern is specified, the search for similar patterns may be started. Pattern matching may slow-down the application for a real-time interaction. To solve this problem, we propose a three-step framework to interactively search the data. In Step 1, users reduce the scope of the query by drawing timeboxes in one or more detail views, as shown in Figure 7. In Step 2, users specify a pattern and get a large number of results with an approximate search. Step 3, users refine the query to narrow down the result set by dynamically manipulating the parameters of the search.

In Step 1, the goal is to reduce the size of the dataset to be searched with simple filtering operations. This action will remove the uninteresting time series from the search process. A boolean AND operation is performed between all time boxes. Filtered out items appear light grey on the screen (and can be removed from the display if needed). The immediate feedback given to users clarifies the effect of users' actions and reveals the logic of the underlying process due to timeboxes. In Figure 7, the list of the remaining 3 wells (A, H, and I) appears in the lower right list box. The smaller number of items will allow the next search steps to be performed more rapidly.

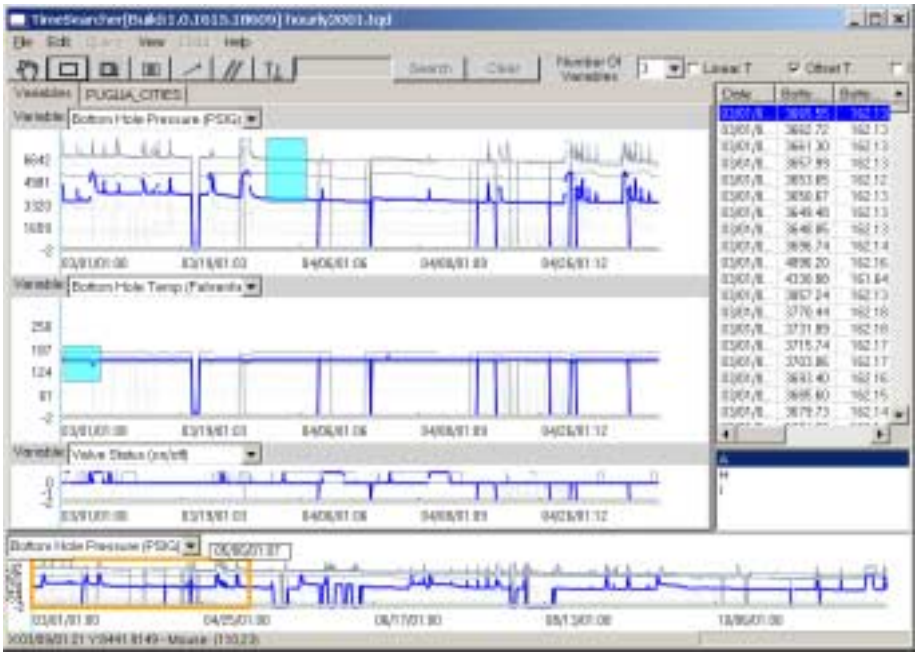

Figure 7: Example of filtering over multiple variables, using an oil production dataset. Each item is a well, for which hourly data is available (pressure, temperature, etc.). Users have narrowed down the list of wells down to three (A, H, and I) by filtering the series to keep only the wells with high "bottom hole pressure" early April and high "bottom hole temperature" at a different period (early March).

Step 2 corresponds to the selection of the specific pattern, and the initial search for similar patterns within the scope specified in Step 1. Users select the pattern in the dataset itself by selecting an item – therefore highlighting it – then drawing a box enclosing the pattern. Figures 9 to 12 show an example with only one variable to keep the figures small. After drawing the box (Figure 8), an adjustable tolerance slider appears, allowing users to roughly set the tolerance of the match between the selected pattern and the search results.

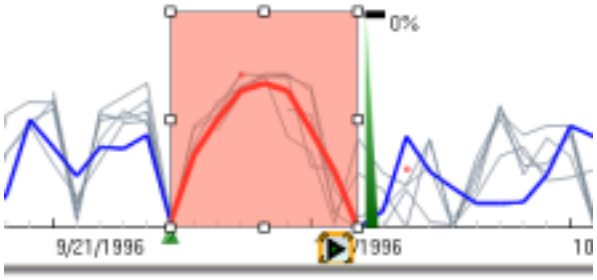

Figure 8: Users select a pattern by first selecting a line (i.e. a time series) and drawing a box around the pattern of interest. The default tolerance is set to zero. The reference pattern is tagged by a green triangle on the detail and on the overview.

The initial search is triggered explicitly by clicking on the arrow-shaped button located below the box. The result of the search is displayed with red triangle markers under the horizontal axes on the overview - and in the detail view(s) when applicable (Figure 9). Together with the triangles, the matched patterns appear with a different color (red) in the detail view and in the overview, when applicable. This step might become slow when the dataset is large but it may need to be performed only once (see discussion section).

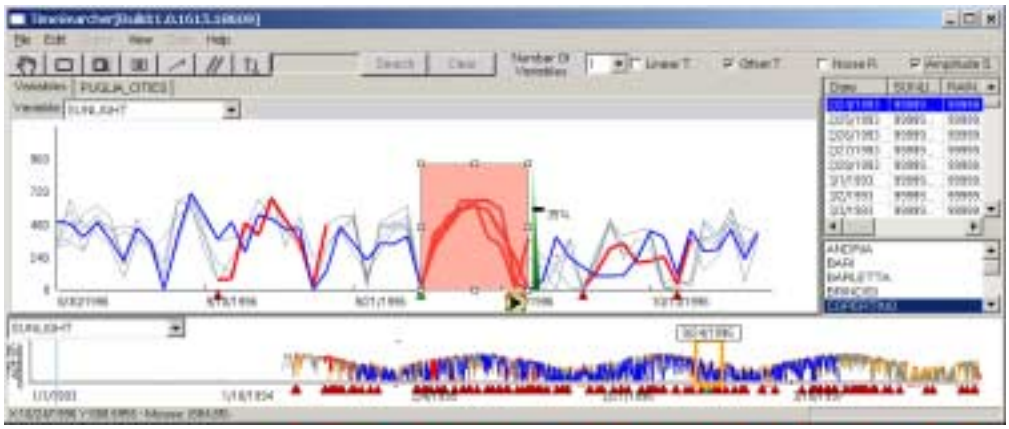

Figure 9: The slider is first adjusted to an arbitrary high level of tolerance and the search is initiated by clicking on the arrow shaped button below the pattern. The start of each match is indicated by a red triangle.

Finally, in Step 3, users reduce the tolerance and adjust search parameters, incrementally. In order to rapidly adjust the size of the result set, the users benefit from having immediate feedback about the effect of their action on the results. Users can adjust the tolerance level (Figure 10), or modify parameters of the search (Figure 11) and immediately see the effect on the cardinality of the result set. Finally, they can use all the basic features presented in the previous subsection 3.1 to browse the results: overview and details, coordinated table to view numerical values, multiple detail views to compare patterns, etc.

Users can repeat the operation by selecting patterns in multiple variables (Figure 1). At this stage our implementation handles the variables independently, i.e. results of the pattern matching in one variable are highlighted in that variable view only and no boolean operation is performed. The overview shows the positions of all patterns matched over the entire time series. Users are then able to browse the results to see how patterns found in separate variables match in time.

Even though the standard setup shows an overview of the entire dataset at the bottom of the screen on a single variable, users can stretch the field-of-view box (the box in the overview panel) to cover the entire time span of the dataset to see an overview of all variables at once in the area generally reserved for details.

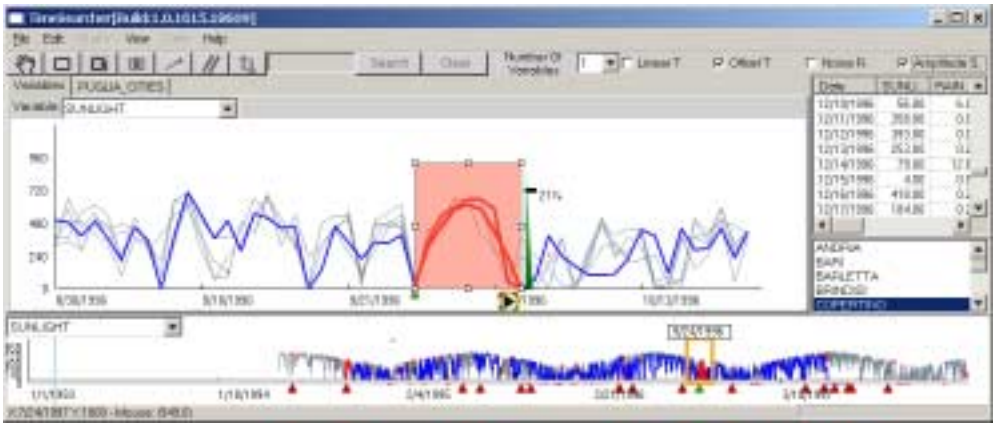

Figure 10: The tolerance level is reduced iteratively until the result set is small enough to be examined.

3.3 Search algorithm used

Our focus was first on demonstrating a general interactive framework for search interfaces, so we started by implementing a simple Euclidean distance algorithm that has the following formula $D(Q,C) \equiv \sqrt{\sum_{i=1}^{n}(Q_i - C_i)^2}$ where Q and C are two time series with n points and D the distance function for two time series. To improve performance we remove the square root, which is acceptable since the Euclidean distance is monotonic, so the formula becomes: $D(Q,C) \equiv \sum_{i=1}^{n}(Q_i - C_i)^2$. It is still possible to compute the actual distance value after removing the square root. In Figure 13, a graphical example of the computation of the distance between two time series is shown.

This simple search algorithm was chosen because the data provided by our users were not extremely large, had no periodicity, and because our users could not predict the size of the pattern that would need to be searched, which meant that existing advanced search algorithms could not be used directly (see discussion in section 4).

Euclidean distance alone may not be enough to detect two similar looking patterns as similar. We implemented four transformations: offset translation, magnitude scaling, linear trend removal and noise reduction[1]. TimeSearcher 2 allows users to specify those transformations or disable some of them, depending on their needs (using the checkboxes always available in the top-right area of the application). For instance, users may turn off the magnitude scaling transformation when it is important to preserve proportions (Figures 11 and 12).

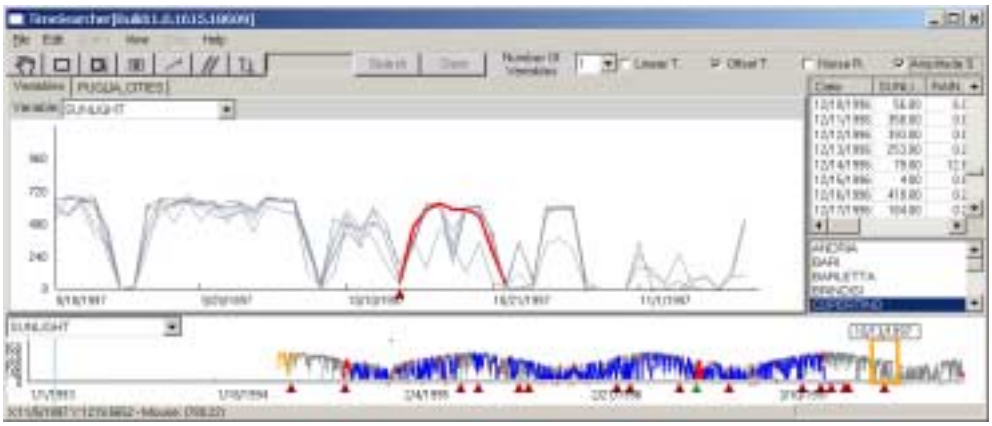

Figure 11: Exploring the results by moving the field-of-view box.

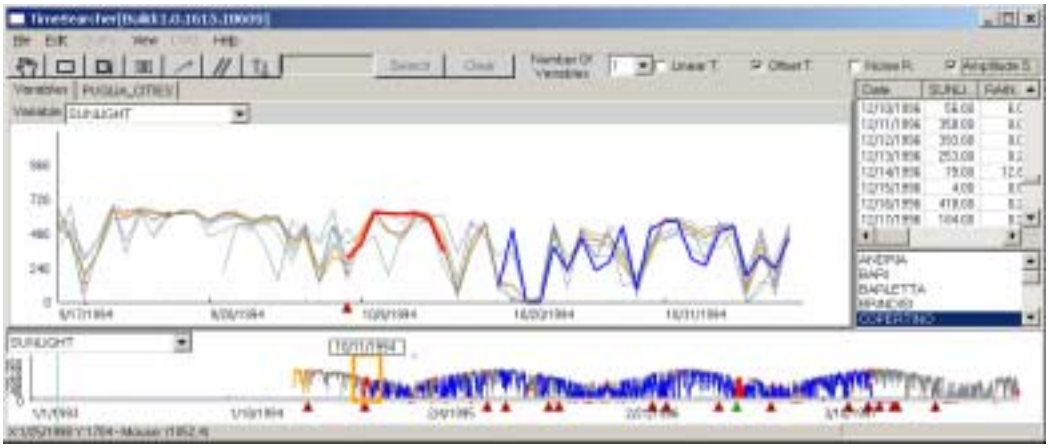

Figure 12: When a search result is found unacceptable, users can adjust the algorithm parameters. Here users realize that they care about the amplitude of the pattern so they decide to turn off the "amplitude scaling" option at the top right of the screen

The meteorological dataset contains 2000 time points (daily means of temperatures, humidity, etc.). Each time point has 8 variables and it is possible to perform the search on 15 items (cities in our case). All queries in this reasonably sized dataset can be performed in real time because we can load all the data in memory, therefore achieving dynamic queries[18]. Dynamic queries imply rapid, incremental, and reversible actions; and the immediate display of feedback (less than 100 milliseconds). The oil production data sample has approximately 10,000 time points and 7 variables for 10 wells, and dynamic queries were not achieved at this point.

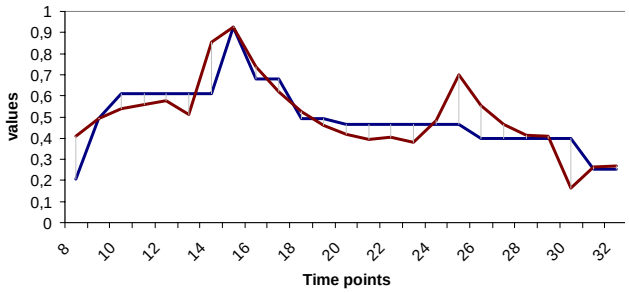

Figure 13: Euclidean distance between two time series. Distance is computed by matching the corresponding points along the horizontal axis.

4. DISCUSSION AND FUTURE WORK

The tool still has many opportunities for improvement. Main issues include dealing with larger datasets, improving the interface, and dealing with missing data.

Dealing with larger datasets: Our algorithms can be optimized but for much larger datasets further development is needed to handle each step separately so that slower steps can be triggered explicitly (by pressing a button) while others are triggered implicitly to achieve dynamic queries. One technique consists of using Step 2 as another scope reducing step. Once users have run the search once, with a high tolerance value of X, the result set R can be used as the new reduced scope of refinement queries using tolerance Y<X (i.e. when dragging the slider up). This makes the operation of reducing the tolerance faster as the tolerance level is reduced). Increasing the tolerance would require running a search on the entire data again, unless the algorithm keeps track of the value for which a pattern was dropped off the result list. And if the value is greater than X, or when transformations are checked or unchecked, then a complete search on the

entire scope defined in Step 1 has to be performed again. Another possibility is to index the time series in step 2 so the dynamic queries may be applied also for a bigger data set. When the data (indexed or not) is larger than the high-speed storage capacity, the three-step framework can be applied as well. The initial search of Step 2 may be slow, but Step 3 can become interactive when the Step 2 result set is small enough to fit into the high-speed storage.

When the size of the pattern is identified in advance, simple search can be replaced by faster indexed searches that find a small set of results then apply the Euclidian distance to quickly refine this set. This is the case when seasonal cycles are present in the data, or in general, when the data are cyclic. For example, for EKGs the cycle is on the order of a second, and weekly cycles are likely to be present in time series of human work activities. Those cycles are important to index the data for pattern search because very efficient indexing techniques are available when the duration of the patterns to be searched is known in advance [21, 22]. When the length of the search pattern cannot be known in advance but only approximated to fall within a fixed range, then multiple indices can be created for a reasonable set of different pattern lengths, having an index for each length.

Improving interaction: Our early feedback highlighted the benefits of good browsing capabilities. Exploring the data visually is extremely important, and providing access to the numerical values needs to be supported with coordinated views of the graphs, tables and lists. Multiple presentations of the data are useful (for example overlapping versus sequential views of the items). User suggestions included providing multiple methods to specify the pattern and the options of the search algorithm. For example, one alternative to specifying tolerance would be to allow users to interactively modify the selected pattern or to draw boundaries of tolerance on the pattern by interacting with the line using direct manipulation. When the users know the pattern they are looking for and it is not easy to find an existing similar one, pure sketching may help[15]. The selection of patterns in existing data would also be complemented by a pure sketching option. This could easily be done with the current software. Alternatives can be offered as well for the selection of search parameters. Our early user feedback indicates that some users are confused by the tolerance slider label, because the tolerance measure has no real meaning to them. We initially chose to display a number to allow users to return to a previous value that had been found useful. Hopefully, the dynamic query behavior will help users quickly understand how to use the tolerance slider. An option might be to avoid displaying a numerical value and only provide + and – controls. Another natural suggestion is to extend the scope-setting Step 1 by allowing users to limit the time range where the search should be performed. Currently, nothing requires users to start with Step 1, and they can start with Step 2 and 3. Nevertheless it might be good to encourage users to reduce the scope of the search with the filter boxes when the size of the data doesn't allow the use of dynamic queries.

Dealing with missing data: Missing data is another common problem that needs to be carefully addressed. Time series make it easy to inform users of the fact that data is missing[19] and TimeSearcher 2 shows the location of the beginning of missing data (Figure 5). Nevertheless, standard annotation mechanisms are needed to inform users about the default method used to handle missing data in the search (is the missing data ignored? Is it considered a perfect match? Is it replaced by an estimated value?), and users should be able to specify what method is to be used. Our experience suggests that providing annotation mechanisms will also be important to document the reasons for the absence of the data and to use the exploratory tool as a way to gather the anecdotal knowledge explaining some of the data.

Evaluation: The formal evaluation of exploratory tools such as TimeSearcher remains a challenge[20]. Formative usability studies will allow us to improve the interface but, more importantly, we will continue working with users to identify and report on case studies, that will help us understand the range of data type and application for which TimeSearcher 2's pattern search can be effective.

5. CONCLUSIONS

Our review of the literature and the work we presented suggest that there are many ways interactive exploration of time series can be improved. This is a lively field and users will benefit greatly from systems that combine both improved algorithms and empowering interactive interfaces. This work illustrates how traditional features such as overview and details combined with an interactive search interface can help users perform exploratory analysis on time series. We believe that our proposed three step framework can naturally be expanded to environments using advanced algorithms on indexed data, thereby providing interactive access to much larger data archives with a combination of explicit and implicit queries.

ACKNOWLEDGMENTS

A significant part of this work was conducted while Paolo Buono visited the University of Maryland Human-Computer Interaction Laboratory in the Spring of 2004. We thank Eamonn Keogh, Harry Hochheiser and Maria Francesca Costabile for their feedback and suggestions. Partial support for this research was provided by Chevron Texaco.

REFERENCES

1. Keogh, E., Data mining and machine learning of time series data: a tutorial, *Proc.14th European Conference on Machine Learning (ECML) and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Dubrovnik, Croatia, (September 23, 2003).
2. Carlis, J. V., and Konstan, J. A., Interactive visualization of serial periodic data, *ACM Symposium on User Interface Software and Technology*, ACM Press, New York (1998), 29-38.
3. Hochheiser, H., and Shneiderman, B. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration, *Information Visualization*, 3, 1 (2004), 1-18.
4. Unwin, A. and Wills, G., Exploring time series graphically, *Statistical Computing and Graphics Newsletter*, 2, (1999), 13-15.
5. Keim. D. A., Pixel-oriented visualization techniques for exploring very large databases, *Journal of Computational and Graphical Statistics*, 5, 1 (1999), 58-77.
6. Keller, P., and Keller, M., *Visual Cues: Practical Data Visualization*, IEEE Computer Society Press, Los Alamitos, CA (1994).
7. Embedded Component History Object, http://www.echohistorian.com, (Accessed October 25, 2004).
8. SAS Institute, http://www.sas.com, (Accessed October 25, 2004).
9. ILOG - JViews, http://www.ilog.com, (Accessed October 25, 2004).
10. Personal StockMonitor, http://www.personalstockmonitor.com, (Accessed October 25, 2004).
11. Brodbeck, D., and Girardin, L., Interactive poster: trend analysis in large time series of high-throughput screening data using a distortion-oriented lens with semantic zooming, *IEEE Symposium on Information Visualization (InfoVis 2003)*, IEEE Press, Piscataway, NJ (2003).
12. Hochheiser, H., *Interactive Graphical Querying of Time Series and Linear Sequence Data Sets*, Ph.D. dissertation, University of Maryland Dept. of Computer Science (April 2003).
13. Chortaras, A., *Efficient Storage Retrieval and Indexing of Time Series Data*, MSc Thesis, Imperial College, London (2002).
14. Lin, J., Keogh, E., Lonardi, S., Lankford, J. P., and Nystrom, D. M., Visually mining and monitoring massive time series, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, New York (2004), 460-469.
15. Wattenberg, M., Sketching a graph to query a time-series database, *CHI '01 Extended Abstracts on Human factors in Computing Systems*, ACM Press, New York (2001), 381-382.
16. Chittaro L., Combi C., and Trapasso G., Data mining on temporal data: a visual approach and its clinical application to hemodialysis, *Journal of Visual Languages and Computing*, 14, 6 (December 2003), 591-620.
17. Jerding, D., and Stasko. J., The Information Mural: A technique for displaying and navigating large information spaces. *Proceedings IEEE Symposium on Information Visualization*, IEEE Press, Piscataway, NJ (1995), 43-50.
18. Shneiderman, B., Dynamic queries for visual information seeking, *IEEE Software*, 11, 6 (1994), 70–77.
19. Eaton, C., Plaisant, C., Drizd, T., The challenge of missing and uncertain data, Poster Summary in the *Adjunct Proceedings of IEEE Visualization Conference*, Vis03 (2003), 40-41.
20. Plaisant, C., The challenge of information visualization evaluation, *Proc. of Conference on Advanced Visual Interfaces* (AVI 2004) ACM Press, New York (2004), 109-116.
21. Keogh, E., and Kasetty, S. (2002), On the need for time series data mining benchmarks: A survey and empirical demonstration, *Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, New York (2002), 102-111.
22. Keogh, E., Chakrabarti, K., Pazzani, M. and Mehrotra, S., Locally adaptive dimensionality reduction for indexing large time series databases, *Proc.of ACM SIGMOD Conference on Management of Data*, ACM Press, New York (2001), 151-162.