

V-Miner: Using Enhanced Parallel Coordinates to Mine Product Design and Test Data ¹

Kaidi Zhao, Bing Liu
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan St., Chicago, IL 60607
{kzhao, liub}@cs.uic.edu

Thomas M. Tirpak
Motorola Labs
1301 E. Algonquin Rd. Room 1014
Schaumburg, IL 60196
T.Tirpak@motorola.com

Andreas Schaller
Motorola Labs
Heinrich-Hertz-Str.1 65232
Taususstein, Germany
Andreas.Schaller@motorola.com

ABSTRACT

Analyzing data to find trends, correlations, and stable patterns is an important task in many industrial applications. This paper proposes a new technique based on parallel coordinate visualization. Previous work on parallel coordinate methods has shown that they are effective only when variables that are correlated and/or show similar patterns are displayed adjacently. Although current parallel coordinate tools allow the user to manually rearrange the order of variables, this process is very time-consuming when the number of variables is large. Automated assistance is required. This paper introduces an edit-distance based technique to rearrange variables so that interesting change patterns can be easily detected visually. The Visual Miner (V-Miner) software includes both automated methods for visualizing common patterns and a query tool that enables the user to describe specific target patterns to be mined or displayed by the system. In addition, the system can filter data according to rules sets imported from other data mining tools. This feature was found very helpful in practice, because it enables decision makers to visually identify interesting rules and data segments for further analysis or data mining. This paper begins with an introduction to the proposed techniques and the V-Miner system. Next, a case study illustrates how V-Miner has been used at Motorola to guide product design and test decisions.

Categories and Subject Descriptors

H.2.8 [Information Systems]: Database Management -- Data Mining; I.3.m [Computer Graphics]: Miscellaneous -- Visualization

General Terms

Design, Human Factors.

Keywords

Change patterns, parallel coordinate visualization, rules.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
KDD '04, August 22-25, 2004, Seattle, Washington, USA.
Copyright 2004 ACM 1-58113-888-1/04/0008...\$5.00.

1. INTRODUCTION

This paper describes a multi-variable visualization tool called V-Miner (for Visual Miner) designed for mining product design and test data. The goal is to discover useful or actionable knowledge from mobile phone testing data that can be provided as feedback to design engineers, who will use the knowledge to identify opportunities for improving both the product design and the product development process. In this way, the design cycle of new products can be shortened.

1.1 Design Process

At a high level, it is possible to characterize the typical design process for consumer electronic products, such as mobile phones, as follows:

1. Engineers, who are experts in mechanical, electrical, software, etc., aspects of mobile phones, design their specific sections of the phone, based on previous successful designs, new product specifications, design simulations, and general design guidelines.
2. After the mechanical, electrical and software designs are finalized, some prototypes are built.
3. A set of functional tests is performed on the prototypes to assure that the product fulfills the requirements. According to the performance of the product testing parameters, the engineers can verify whether the design meets the product requirements. If it does not meet the requirements, the design engineers have to modify the existing design, which leads to the next design cycle, i.e., returning to step 1.

The engineers need to repeat the above three steps until the design meets the complete specification. After that, the phone will be released to the New Product Introduction (NPI) Team, who will coordinate the release to volume manufacturing.

For existing product platforms, there is typically a large knowledge base available. Engineers have a good understanding

¹ We would like to thank Thorsten Hofer and Knut Moeller of the Motorola Personal Communications Sector in Flensburg, Germany, for performing the mobile phone tests, collecting and organizing the data sets used in this project, explaining the details of the test data, and providing us with feedback on the use of our system for their applications. We also thank Weimin Xiao for many insightful discussions, and Tom Babin for his review of this manuscript, MATC document 20041132M-19.

of potential design issues and possible ways to address them. However, for a new product platform, which incorporates fewer previously proven solutions, it may be necessary to coordinate a number of iterations of design revisions, prototype builds (also called *proto-builds*), and product tests. A large number of measurements are made for each proto-build, e.g., more than 100 variables are tested to characterize the electrical performance of a mobile phone. In order to reduce the engineering costs and cycle time associated with this design process, as well as to minimize the opportunities for design defects, we developed a method and software tool for mining useful knowledge from electrical test data that can be used to guide the decisions made by mobile phone designers.

1.2 The Data

At the first stage, which coincides with the early prototype stages of a new mobile phone product at Motorola, engineers performed an extensive set of tests on a particular type of phone. After each design change, all the electrical test variables were measured. The resulting measurement data were mined. There are more than 100 test variables that characterize the performance of a mobile phone. Each variable takes numeric values and has the following characteristics:

1. It has an *upper limit* and a *lower limit*. Any value that exceeds either of the limits is considered unacceptable, i.e., the test fails for this variable. A design modification is needed to bring the failed variable to its acceptable range.
2. It also has an ideal value, which is called the *target value*. The closer the value of the variable is to its target value, the better it is.

Table 1 shows a sample test data set. *Value for change i* is the measured value for each variable after the *i*th component change.

Table 1. Sample electrical test data

Test variable	Target value	Lower limit	Upper limit	Value unit	Value for change 1
Variable 1	0	- 4	4	Voltage	0.9
Variable 2	50000	50000	55000	HZ	495830
.....

Understanding the sequence of values for incremental design changes 1, 2, ... is a significant part of our analyses. Note that tests were done after each component change, and that once a change was made to the design, it was not changed back to its original component. Thus, we can view each change of a component on the phone as a new design. Subsequent changes are based on earlier changes. Thus, the data can be treated as a sequential data set.

With the testing data, product designers are typically interested in the following:

- Prominent (or significant) changes in variable values after some design changes.
- The cause of these significant value changes.
- Stable variables whose values are not affected by design changes.
- Patterns of values, changes, and failures.

1.3 Using Traditional Rule Mining Systems

It is easy to think of ways to use classic data mining algorithms to mine patterns from the data. For example, it is possible to use association rule mining [1] to find associations or use a decision tree [11] to find failure patterns. However, these algorithms are inadequate for this task for the following reasons:

1. Due to the large number of variables (more than 100), association rule mining generates too many rules. We ran an association rule mining system and found more than 20,000 rules, which is too large for any human user to analyze and identify the interesting rules. It should be noted that in order to apply a rule mining technique, the variable values were discretized into intervals.
2. The decision tree approach was initially tried by Motorola engineers to find patterns in the test data. However, the problem with the decision tree method is that it does not find all the interesting patterns, but only a subset of the patterns that exist in the data. In many cases, the discovered patterns may not be the ones that are the most interesting to the decision makers, e.g., the product designers and test engineers. It was also tedious to run a decision tree program, because each failed variable has to be set as the class variable in order to find patterns related to its failure.

In our approach, we use parallel coordinates based visualization, which gives an intuitive view of the underlying data, thereby enabling the user to identify interesting patterns easily and quickly. Parallel coordinate visualization is suitable for the mobile phone test prototype application because, as explained later in this paper, interesting patterns can be easily seen from the visualization using various querying and sorting options. Feedback from the Motorola engineers, who carried out the data mining using our system, confirmed this.

By no means, we say that rule mining or other data mining techniques are not applicable. Actually, the engineers have some related but previous data mining results in the form of rules, which are helpful in this project. Our system also has the capability to visually filter data using rules, which allows the user to study the data records covered by certain rules. Detailed examples will be given in Section 4.

1.4 Parallel Coordinate

This section presents a brief review of the parallel coordinate method [7] for visualizing multi-variable data in a 2D space. In this approach, a separate vertical axis is assigned to each variable. Multiple vertical axes, i.e., one for each of the coordinates (variables) of the data set, are arranged at an equal distance from each other along the horizontal axis. For each data point (or record), a coordinate value is plotted along its respective vertical axis. The coordinate values for a given data point for two adjacent vertical axes are connected with straight lines. In this way, an *n*-variable data record can be visualized as a polygonal line drawn across the *n* parallel vertical axes with *n*-1 line segments. Figure 1 depicts a data set with nine 6-variable data records.

By viewing the arrangement of the lines on the visualization, the user can search for patterns visually. Those line segments that have a similar slope indicate that their corresponding data may have some correlations. For those similar data records, their corresponding lines will be visualized with similar shapes. In this

way, the task of searching for relations among multi-variable data is transformed into the problem of 2D pattern recognition. This shifts the computational load from numerical reasoning to visual reasoning, which is simpler for human beings.

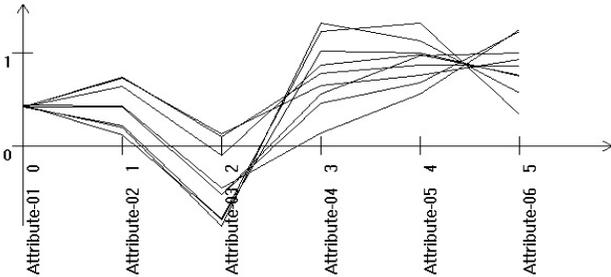


Figure 1. An illustration of parallel coordinates

Using the traditional parallel coordinate technique directly, however, is not sufficient for our application for two reasons:

1. The traditional parallel coordinate method does not consider the sequence in which the data were generated, and the significance thereof. The data sequence may be important, as it contains information about changes in a product’s electrical performance after the first, second, etc., prototype cycle. Thus, we have added a sequence component to the traditional parallel coordinate visualization.
2. The traditional parallel coordinate visualization does not consider the ordering of the variables. Thus, vertical axes are ordered in an arbitrary manner. In the V-Miner software, an edit distance based querying and sorting tool is implemented, to allow the user to issue queries that subsequently rearrange the axes according to the results of the queries. In our application, this definitely facilitated the discovery of interesting patterns and correlations.

Details of these two enhancements will be discussed in Section 3 of this paper, and their respective benefits will be presented in a case study in Section 4.

2. RELATED WORK

Parallel coordinate techniques are widely used for multi-variable visualization. In [6], a parallel coordinate method helps the user understand how a certain design compares to other designs. It allows the user to investigate correlations between variables (dimensions) in the data by manually selecting a “driving” variable to color the lines, or by manually re-arranging the ordering of variables. This is different from our V-Miner system as we perform this task automatically or semi-automatically based on the user’s interests.

In [9], the WinViz system was described, which is an enhanced parallel coordinate technique. In this system, each polygonal line in the visualization may represent several data n -tuples satisfying the attribute values specified by the user. Group bars are used in place of attribute values on each vertical axis, with the size of the bars indicating some related information such as population size. This helps to reduce the complexity of the visualization when there are many data points. The system also supports visual querying, in which the user can formulate simple AND and OR

type of queries. The system highlights the specified conditions in the visualization. This is different from our approximate/similarity queries, as we will explain in Section 3.

Another parallel coordinate approach is described in [3]. The PARCOVI software tool follows the original idea of parallel coordinates, yet it aims to be a generic data visualization system, and allows the user to manually select and re-arrange variables in the visualization. It also incorporates scatter plots for the purpose of cross-checking and verification. VisDB [8] is also a generic visualization tool, which allows exploration of large databases using visualization techniques, such as parallel coordinates and pixel-oriented techniques.

The above systems aid in information discovery. Each has its advantages. However, these applications are based on an implicit assumption about the data. Namely, the data records are independent of each other, and thus the results obtained from data mining are independent of the order in which the records are fed into the data mining system. In our application, where the goal is to characterize a set of sequential prototypes of a mobile phone design, this assumption is not true.

Our work is also related to shape querying [2] and axes sorting. The method presented in [2] includes a shape definition language for retrieving objects based on shapes contained in the histories of these objects. The history of an object is represented by its values at each point in time. Shapes are expressed with a set of primitives, e.g., “up”, “down”, “stable”, etc. Given a shape query, the system will return all the objects (variables) whose histories match the query. This approach, however, does not allow approximate matching, which is very important in our work, since not all test values are available for all prototype designs. Furthermore, [2] is not concerned with visualization. In our work, we use edit distance [5] in the similarity matching. In our visualization, we allow the user to specify a query shape either by indicating an example data point or by explicitly specifying the shape of interest. The system can then sort the variables according to edit distance results.

In [10], an algorithm for ordering categorical values in parallel coordinates is proposed with the following three steps: 1) constructing natural clusters of categorical values based on domain semantics; 2) ordering the clusters; 3) ordering the categorical values within each cluster. As all the variables for our application are numeric, the technique described in [10] is not applicable. In [4], an ordering algorithm is proposed for numerical variables. It shows that the best arrangement of coordinates/variables is NP-complete. Thus, some heuristic algorithms are proposed for the variable arrangement problem. The basic idea is to make sure that the most similar variables are placed next to each other after the rearrangement. This is an important and useful idea. Our work is somewhat different, though, as we wish to allow the user to issue approximate queries. We believe that performing a single sort for the best global result is not flexible and not always effective because users always have different interests, and their interests also change with time.

3. THE ENHANCED PARALLEL COORDINATES

We have extended the basic parallel coordinate technique in two major ways, namely, by adding trend figures and enabling

querying by approximate matching. We now discuss them in turn.

3.1 Trend Figures

As mentioned in Section 2, the classic parallel coordinate visualization does not consider the sequence in which the data records were generated or collected. Thus, it assumes that the order of the data records is of no significance. However, in product testing applications, the sequence in which the values are observed is very important and may reveal some sequence-dependent relations or cause-and-effect relations.

In this work, we extend the classical parallel coordinate method by adding an additional graph for each variable above its coordinate. In the supplemental graph, as shown at the top of Figure 2, the horizontal axis reflects the sequence of the data record, and the vertical axis shows its value in each data record. We call these graphs trend figures. They make it possible to quickly see variables that change in similar ways, by noting their similar trend figures.

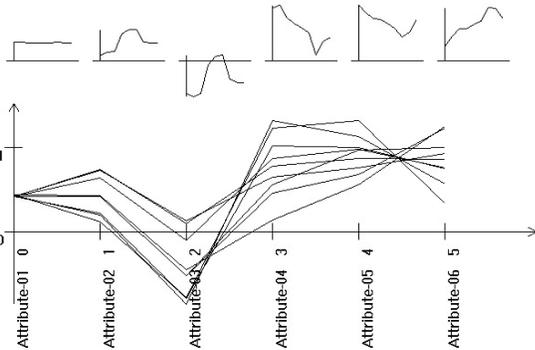


Figure 2. Parallel coordinates with trend figures.

Trend figures are very useful for sequential types of data and do not crowd the space, i.e.,

1. Each added trend figure sits above its coordinate. It uses the space which would otherwise usually be empty. It does not affect the main parallel coordinate visualization and is also space-efficient.
2. In classic parallel coordinate visualization, the overlapping problem of lines significantly hinders the visualization. As the user cannot distinguish one line from another, it is hard to see the changes of the variables. With trend figures, it is clear and easy.

This extension is thus generic and readily applicable to other applications.

3.2 Edit Distance Based Querying and Sorting

Our second major enhancement to traditional parallel coordinate visualization is to allow the user to query shapes based on approximate pattern matching. After the matching is completed, sorting of variables is performed, which enables the user to view the most interesting patterns in nearby sections of the horizontal axis.

Two important types of patterns are the *value change pattern* and the *failure pattern*. For our mobile phone design application, the value change pattern of a variable shows how the variable's value

changes over different design changes. After a design change, if a variable value increases compared to its previous value, we say its value is "up", and we denote it with the character "3" in its value change pattern. If its value decreases after a design change, we say it is "down" and use the character "1" to represent it. If its value remains the same after a design change, we use "2" (stable) to represent it in the value change pattern. With these representations, the behavior history of the variable can be summarized using a value change pattern string. For example, the string "331" means that the value becomes larger after the first and the second changes, and then goes down after the third change. Queries can be issued using such value change strings.

The failure pattern of a variable shows whether or not its value falls outside the upper or lower limit after a design modification. If the variable value is outside the acceptable range, we say that it fails. The letter "F" denotes the failure. If the value is within the acceptable range, we mark it with an "O" (OK). An example of a failure pattern is "OOOFFF", which means that the variable is within the acceptable range for the first three design changes, but fails from the fourth design change onward. The V-Miner system allows the user to query failure patterns.

The value change pattern and failure pattern convert the numerical comparison task to string comparison, which is more convenient and intuitive for human users. Our system allows the user to issue queries by supplying the above two types of string patterns. We employ the edit distance [5] for string comparison. Ordering of variables in parallel coordinate visualization is done according to the comparison results. The query can be formed either by indicating an example data point or by specifying the shape of interest explicitly.

4. PRODUCT TEST APPLICATION

In this section, we describe how our enhanced parallel coordinate visualization tool V-Miner has been used to discover useful knowledge from phone testing data that can be fed back to design engineers, who can use the knowledge to anticipate problems in the design process so as to reduce the number of design errors and to speed up the design process. The tool also allows the user to use data mining rules as a way to filter the data in the visualization, which turns out to be quite useful in practice.

4.1 The Need for Data Mining

The overall goal for the Motorola application is to enable the engineers who design and prototype new products to identify the following:

- Test variables that show prominent changes in their values after some design changes.
- The causes of these significant value changes, i.e., the component changes that have resulted in these large value changes.
- Those stable test variables that are not affected by current design changes.
- Failure patterns of those variables that have failed after certain design changes.
- Test variables that have similar value change patterns.
- Existing rules mined from previous or related projects that

can be used to filter data further data mining.

V-Mine is also integrated with other tools at Motorola so that the user can perform all the tasks in a data mining cycle. Mining can also be done recursively through data filtering.

4.2 Data Normalization

As with most data mining and visualization applications, the first task is to normalize the raw data. Normally, the value range of each variable is made within a fixed range, e.g., -1 to 1. However, this method is not suitable for our data because:

1. It does not consider any user-specified target value, which is very important in our data. If one uses a fixed range, it will be difficult to see the target value of each variable and how far the actual value is from its target value.
2. It does not consider the lower and upper limits. Thus, it will not be easy for the user to see whether a variable fails after some component is changed.

We have designed several normalization methods that clearly separate values within the normal range from those outside the normal range. Variables whose values are out of range will be normalized to either larger than 1 or less than -1. Thus, the normalized values close to 0 are the ones that are close to the target values. The following is one example of them:

```
Procedure normalization (value, min, max, target)
  // return value stores in: normalized_value
  if ((value >= min) && (value <= max)) then
    normalized_value = (value - target) / (max - min);
  else
    if (value > max) then
      normalized_value = (value - target) / (max - min) + 1;
    else // value < min
      normalized_value = (value - target) / (max - min) - 1;
    end-if
  end-if
```

This normalization makes it very easy for the user to visualize important patterns. The above method is only one of the possible ways to perform normalization. V-Miner offers the user a number of normalization methods from which to select. The user can also switch from one to another at any time to obtain better visualization for different datasets and/or pending decisions. In our interviews with Motorola engineers who had used V-Miner, we found that this intuitive normalization was one of their favorite features of the software.

4.3 A Typical Scenario

We now present a typical knowledge discovery scenario. After normalizing the data and loading it into V-Miner, the user is presented with the visualization as shown in Figure 3. Due to confidentiality concerns regarding the actual product data, the test variable (attribute) names have been replaced with generic names "Test-Attribute-*i*" in all the figures.

In Figure 3, the main window on the left displays the parallel coordinate visualization of the data. Most of the user interactions with the system are performed in this window. The horizontal axis shows all the test variables. Their names are displayed below the horizontal axis. The vertical axis displays the normalized value of

each variable after every component or design change.

The information window on the right displays the detailed information as the user moves the mouse cursor over points in the main window.

The key features of the visualization include:

1. Data from different designs (component changes) are visualized using different colors. The same color scheme is also used on the right information window so that the user can easily relate the visual cues found in the left window to the detailed information shown in the information window on the right.
2. For each test variable, a trend figure is drawn at the top of the screen. These small figures complement the main visualization in that they show the correlation of component changes and value changes of each test variable. Those test variables which have similar change patterns will have similar figures.
3. There are two dashed lines on the vertical axes at $Y = 1$ and $Y = -1$ for both the parallel coordinate diagram and the trend figures above. Due to the way that the data are normalized, these two lines enable the user to instantly identify the out-of-range (failure) values.
4. The querying mechanism discussed in Section 3 allows the user to sort the variables in order to see interesting patterns and facts conveniently. We will give some examples below.

After loading data into V-Miner, the user can identify some significant characteristics from the visualization (Figure 3):

1. The user can easily identify which values of a variable are out of the range and which are within range. Only those values that are between -1 and 1 are within the normal range of a variable, e.g., test variables 19, 20, etc.
2. From the trend figures on top of each variable, it is possible to see that some variables behave similarly, e.g., variables 33 and 34. This suggests that there are some correlations among these variables for the given sequence of design changes.
3. Some variables have stable values over all the tests.

In classic parallel coordinate visualization, the overlapping lines significantly hinder the visualization. Here, we observe that the trend figures mitigate this problem to a great extent, as the users can easily see the changes and trends of variables from the figures.

4.4 Example Findings

Next, we present some example findings from our system. Suppose that at first, the user is interested in identifying stable variables, whose values do not change a great deal over different design changes. It is possible to issue a "222..." query on the value change pattern to obtain this information. Figure 4 shows the query result based on the value change pattern. The test variables on the horizontal axis are ordered in such a way that stable variables appear first on the left side of the visualization.

Note that V-Miner also allows the user to hide variables if they are deemed unnecessary for further analysis. This reduces the

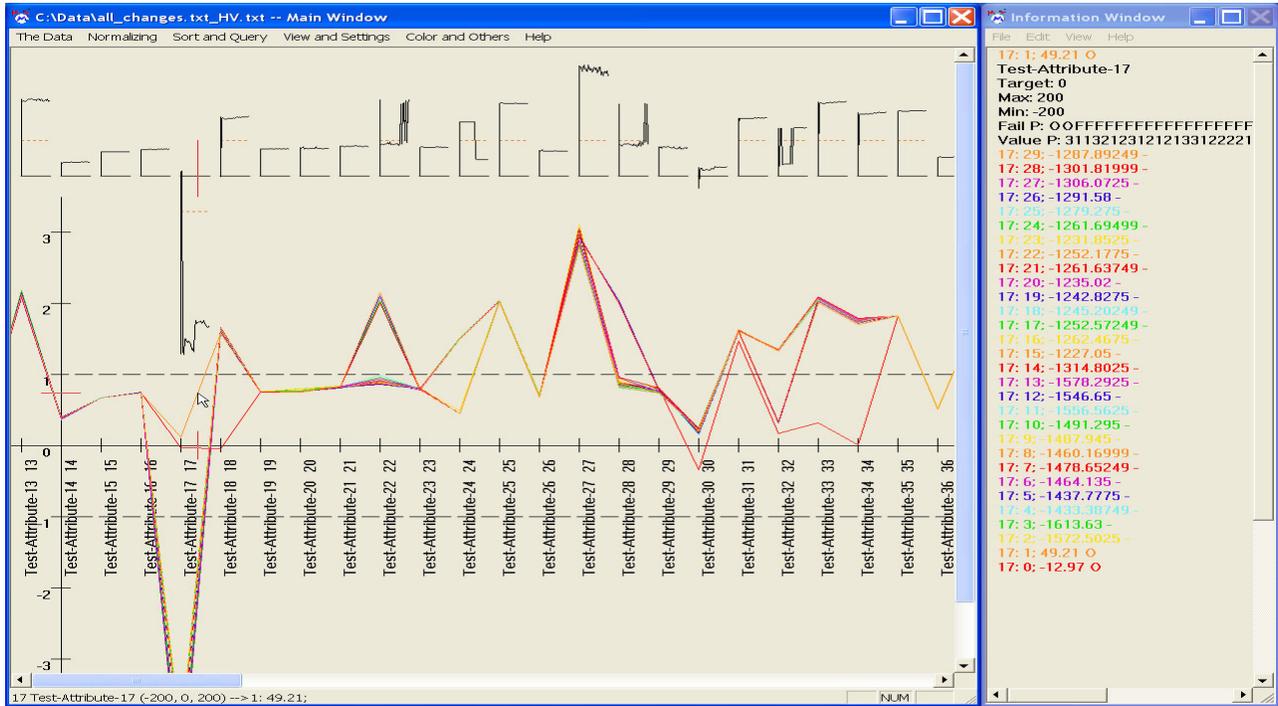


Figure 3. Initial Display of the Test Data

complexity of the visualization and enables the user to better focus on the remaining variables.

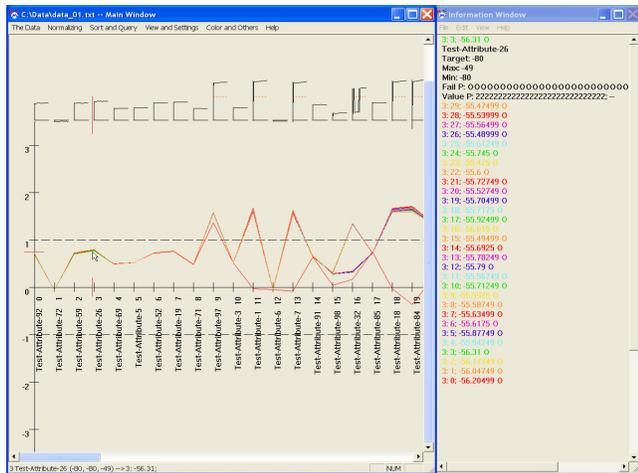


Figure 4. Stable variables

In Figure 4, one can observe a very interesting phenomenon. Some variables, e.g., 18 and 19, fail the tests after the first design modification, and their values remain in the failure range after that. The user can specify a query function on the failure pattern to see if any other test variables display a similar pattern. Figure 5 shows the results of this query. In Figure 5, the values are reordered along the horizontal axis according to their similarity to the query variable. From the main visualization window, one can see that the first fourteen test variables have the same patterns. From the trend figures on top, it is clear that they fail all the tests except the first one. This reveals that during the first design

modification, a certain component change had adversely affected these variables. Furthermore, the failures persist throughout all the subsequent design changes. This tells the user that this component has a major impact on these test variables and, therefore, should be the focus of their attention for the next prototype design.

We can see that the trend figures again play an important role in this finding. Without the trend figures, it is difficult to interpret the main parallel coordinate visualization, due to the overlapping lines.

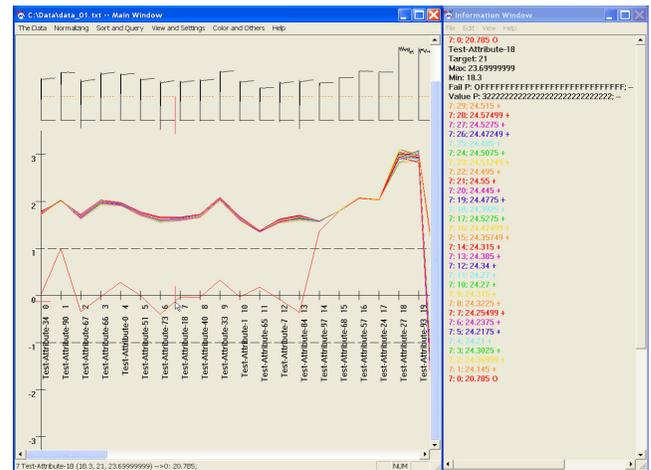


Figure 5. Test variables that failed after the first design change

engineers to select a subset of data for further examination, i.e., the reduced set of data is used in the next data mining cycle. This process has been shown very effective in practice.

One may ask why visualization is needed to filter the data because in many cases filtering can be done easily with a simple procedure. Visualization, however, plays an important role. V-Miner does not act simply as a tool that filters the data using rules. Instead, V-Miner provides an opportunity for the user to interact with the data visually. Through visualization the user can decide whether a sub-population is worth further study. When the user decides to filter the data using a certain set of rules, the first step is to study the rules with respect to their applicable coverage. After the filtering is done, the user is able to see the immediate results, from which it is possible to verify previous assumptions on the data. Also, if the user is not satisfied, it is possible to undo the filter, and try another set of rules. This whole process is possible only with instant visualization and visual data manipulation. Clearly, this process is much more effective than writing a simple procedure to filter the data without the user visualizing the data. Note that before filtering the data, the user can visually manipulate the data. The user may also filter the data by values on certain attributes, etc. Last but not least, all these procedures can be used in conjunction with the visual mining options described above. The resulting sub-data set can be a very complex one, which is not possible to obtain without V-Miner.

We now demonstrate part of this process using a different data set from the one for the first example in this paper. The data set is initially visualized as in Figure 10.

At this point, the engineers have a set of rules from a data mining tool. These rules are loaded into V-Miner, and the user can select the rules to be used from Figure 11.

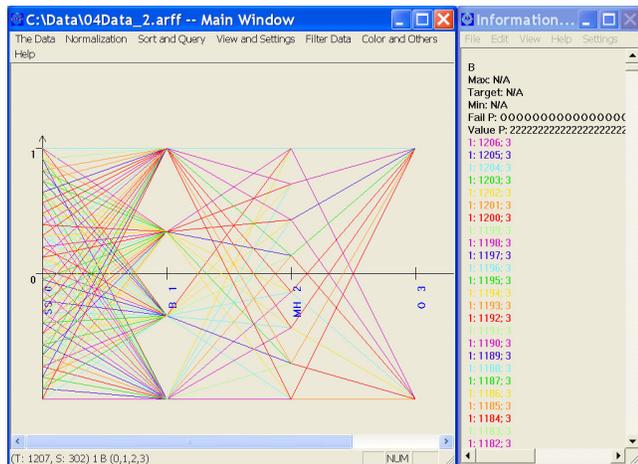


Figure 10. Initial visualization of a data set.

The user has the option to use one rule or multiple rules to filter the data. The user can also apply the rule(s) to those data records currently on screen or to all the data, or to add those data that satisfy the selected rule(s) back to screen.

We show an example in Figure 12. The filtering is done using two rules on the data set. If the user finds some interesting features from the results, i.e., the segment of the data, it is possible to save the data and study them using other data mining tools or do

further mining using V-Miner. In this way, the data mining task can be done in a recursive manner, which leads to a finer granularity of mining.

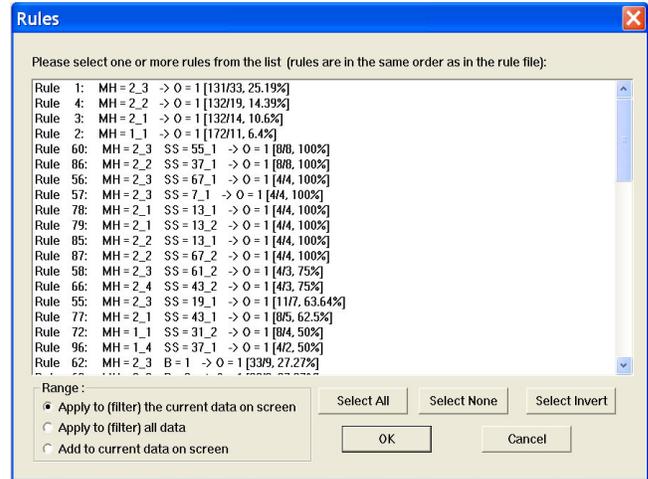


Figure 11. Select the rules to be used.

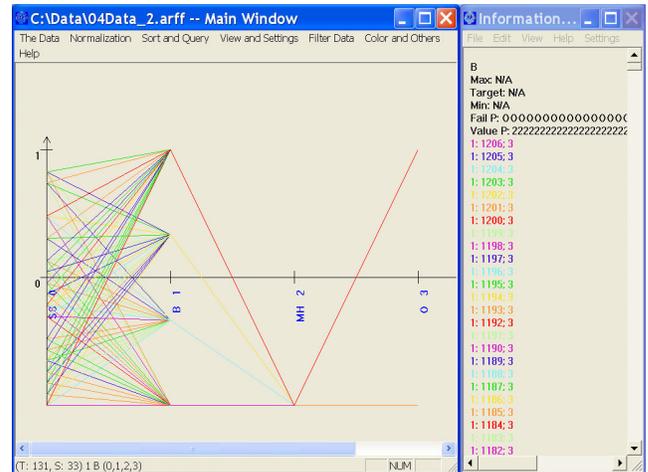


Figure 12. Visualization of after data filtering using rules.

4.6 Application Feedback

The visualization system described in this paper has been used by engineers at the Motorola Personal Communications Sector factory in Flensburg, Germany, and at Motorola Labs in Germany and the U.S. for more than a year. During this time, they have provided feedback regarding the application of the V-Miner software and have reported that the tool indeed helps them find useful patterns and information from product test data.

- Thanks to the enhanced normalization and visualization, it is possible to identify variables with prominent changes in a single glance at the V-Miner window.
- It is easy to see the failure patterns of variables and to group related variables.
- Querying and sorting functions are used frequently to find ad hoc patterns that are of interest to the decision makers.

- The visualization tool has helped to confirm rules found by other data mining tools, such as rule induction with DTE [12]. Furthermore, V-Miner is able to find some knowledge that cannot be found by these tools, e.g., variables that are correlated with each other, and failure patterns in sets of multiple variables.
- The visualization system significantly speeds up the data mining process.
- Using rules to filter the data in V-Miner facilitates the process. The engineers can use V-Miner and their favorite tools together and recursively to mine for finer details.

5. CONCLUSIONS

In this paper, we have introduced the V-Miner visualization system and described its application for mining mobile phone design and test data. V-Miner implements two important extensions to classical parallel coordinate visualization. Figures that show trends of the variables are added into the visualization to summarize sequence-dependent trends in the data. An edit distance based technique is also included to rearrange and group variables so that interesting patterns can be easily identified, i.e., variables showing similar patterns are viewed in adjacent sections of the visualization. V-Miner also has a query engine that enables the user to specify patterns to be mined and displayed. Furthermore, the system can use rules to filter the data visually. This allows the use of data mining results from other tools in V-Miner. V-Miner, together with the other data mining tools, enables a data mining cycle, in which the user can narrow down the raw data for focused analysis or mining. Experimental results and feedback from Motorola engineers, who have used the V-Miner software for more than one year, indicate that the proposed methods are both powerful and easy to use.

Two areas for future work have been identified. The first is to add functionalities to V-Miner so that it can also analyze time-series data. The second is to use V-Miner to guide product testing. Traditionally, data mining is performed after a large amount of data is collected. However, the problem with this approach is that the data collected may not be the most appropriate for the application, and consequently few interesting patterns may be found. Thus, it is desirable to develop a process in which testing (which generates data) and data mining are done concurrently. The key advantage of this approach is that testing is not done blindly. Instead, it is focused on the problems discovered from mining previous test data.

6. REFERENCES

- [1] Agrawal R. and Srikant R. "Fast algorithm for mining association rules" VLDB-94, 1994.
- [2] Agrawal R, Psaila G, Wimmers E.L, Zait M. "Querying Shapes of Histories". In Proceedings of the 21th VLDB Conference. 1995.
- [3] Alexakis A, Deftereos M, Samiotakis Y. "The Parallel Coordinates Visualiser (PARCOVI)". In New Techniques & Technologies for Statistics. NTTS 98.
- [4] Ankerst M., Berchtold S., Keim D.A.. "Similarity Clustering of Dimensions for an Enhanced Visualization of Multidimensional Data". IEEE Symposium on Information Visualization, InfoVis98. 1998.
- [5] Baeza-Yates R.A. "Algorithms for string matching: A survey". ACM SIGIR Forum, 23(3-4):34--58, 1989.
- [6] Goel A, Baker C., Shaffer C.A., Grossman B., Haftka R.T., Mason W.H., and Watson L.T. "VizCraft: A Multidimensional Visualization Tool for Aircraft Configuration Design". In Proceedings of IEEE Visualization'99, San Francisco, CA, October 1999.
- [7] Inselberg A., Dimsdale B. "Parallel Coordinates for Visualizing Multi-Dimensional Geometry". In Proceedings of the Computer Graphics Intl. Conf. 1987.
- [8] Keim D.A., and Kriegel H.-P. "VisDB: Database Exploration using Multidimensional Visualization". IEEE Computer Graphics and Applications, 14(5):40--49, 1994.
- [9] Lee H-Y., Ong H-L., and Singh K. "Visual Data Exploration Using WinViz." Proceedings of the First International Conference on Knowledge Discovery and Data Mining, 1995.
- [10] Ma S., Hellerstein J. "Ordering Categorical Data to Improve Visualization". IEEE Symposium on Information Visualization, InfoVis99. 1999.
- [11] Quinlan J. "C4.5: program for machine learning". Morgan Kaufmann, 1992.
- [12] Zhou C., Nelson P.C., Tirpak T.M., Xiao W., and Lane S.A., "An Intelligent Data Mining System for Drop Test Analysis of Electronic Products Manufacturing". IEEE Trans. on Electronics Packaging Manufacturing. Vol. 24 No. 3, pp. 222 -231, July 2001.