Eurographics/ IEEE-VGTC Symposium on Visualization 2008
A. Vilanova, A. Telea, G. Scheuermann, and T. Möller
(Guest Editors)

Volume 27 (2008), Number 3

Density Displays for Data Stream Monitoring

Ming Hao[1], Daniel A. Keim[2], Umeshwar Dayal[1], Daniela Oelke[2], and Chantal Tremblay[1]

[1] Hewlett Packard Laboratories, Palo Alto, CA
[2] Department of Computer and Information Science, University of Konstanz, Germany

Abstract

In many business applications, large data workloads such as sales figures or process performance measures need to be monitored in real-time. The data analysts want to catch problems in flight to reveal the root cause of anomalies. Immediate actions need to be taken before the problems become too expensive or consume too many resources. In the meantime, analysts need to have the "big picture" of what the information is about. In this paper, we derive and analyze two real-time visualization techniques for managing density displays: (1) circular overlay displays which visualize large volumes of data without data shift movements after the display is full, thus freeing the analyst from adjusting the mental picture of the data after each data shift; and (2) variable resolution density displays which allow users to get the entire view without cluttering. We evaluate these techniques with respect to a number of evaluation measures, such as constancy of the display and usage of display space, and compare them to conventional displays with periodic shifts. Our real time data monitoring system also provides advanced interactions such as a local root cause analysis for further exploration. The applications using a number of real-world data sets show the wide applicability and usefulness of our ideas.

CR Categories and Subject Descriptors (according to ACM CSS): I.3.3 [Computer Graphics]: Picture/Image Generation – Display Algorithms; H.5.0 [Information Systems]: Information Interfaces and Presentation – General.

1. Introduction

Large scale time series data exist in many applications, including financial applications, such as sales or currency exchange rates, as well as process monitoring applications, e.g., measuring system performance. Business analysts and service managers often have to digest and visualize long multi-dimensional time series data to understand business and service performance.

For the analysts it is important to see the "big picture". For example, enterprise data warehouse users want to visualize their system performance (e.g., 8 Servers x 16 CPUs) at a glance to identify which CPU in which server is under-or over-utilized for balancing the workload. Store managers want to replay their historical time series data comparing their last year sales with the sales for this year to determine their marketing strategies. Financial analysts want to have the ability to replay the sales time series data to observe particular sales attributes to make their business decisions. System performance analysts want to catch problems in flight to reveal the cause of anomalies and take immediate action before the problems consume too many resources.

There are two main challenges when visualizing streaming time-series data: First, the amount of data to be displayed varies depending on the demands of the applications and tasks. This concerns both, the measurement time interval and the time period that is relevant for the analysis. While in sales time series the measurement time intervals are often days and the relevant time period are years, in analyzing data warehouse performance data a single day may be the relevant period but detailed information for each second may be required. Secondly, such time-series data sets are usually very large and due to the streaming nature constantly new data is added.

Related work

Most existing information visualization techniques are limited to the presentation of a few hundred or thousand items and display screens are filled up quickly with an incoming data stream. Early approaches to deal with this problem were distortion displays pioneered by Mackinlay et al.'s Perspective Wall [MRC91] and high-density displays like Eick's SeeSoft [ESS92]. Eick allowed users to analyze up to 50,000 lines of code simultaneously by mapping each line of code into a thin row for finding interesting patterns. Later Eick addressed high-density display issues in his Visual Scalability paper [EK00]. Since then, many interesting approaches for visualizing large high-density displays have been developed. One of them is the Information Mural [JS98] that creates a miniature version of the information space by using each pixel for more than one data point if this is required by the data distribution.

In [WS99], a new calendar-based visualization of time series data combined with a clustering technique was developed for users to quickly identify changes over time. In [SSJ05], the authors discuss visualizing sets of non-equally spaced time series arising, e.g., in auction bid time series. In our own previous work [HDKS07], we address the problem of space-filling visualizations of many moderate length time series.

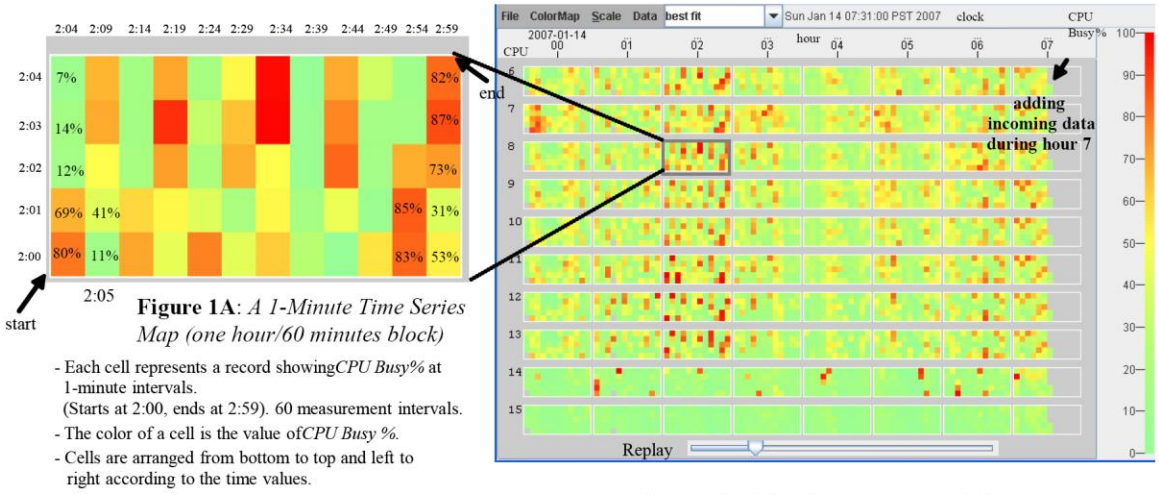


Figure 1A: *A 1-Minute Time Series Map (one hour/60 minutes block)*

Figure 1B: *When the screen is not full: Incoming data is added on the right*

Treemaps in general are also related to our work with respect to their property of space-fillingness. In [FP02], with hardware graphics acceleration and their overlap algorithms, the authors were able to show one million item treemaps and a 40,000 item population map of the US. They also use animation to help users gain understanding across different views. In [YN06], Yost and North describe the results from their visualization scalability study. The results show that by using a combination of perceptual abilities and navigation, users were able to effectively visualize more data on a large display. Our work is also related to focus and context techniques (see [CMS99]) and to the work of Woodruff et al. on constant information density in zoomable interfaces [WLS98].

2. Our Approaches and Evaluation Criteria

Common to the previously related approaches is that they apply a constant information density for displaying large data volume. In this paper, we use cell-based time series maps to visualize large data streams [HDKS07]. In Figure 1A, the color of each cell represents the value of a metric attribute, such as CPU busy. We place multiple cell-based time series in a spreadsheet-like row and column layout. In case of the ***variable resolution density displays*** the size of the cells decreases as more data is read to allow users to get the entire performance/sales data in a single view. For retaining the discovered patterns without movements, we use ***circular overlay displays*** to replace the conventional shifting movements. The circular overlay technique avoids the data shifting movements after the display is full.

In addition, we incorporate some of the previously proposed advanced interaction techniques, such as intelligent visual analytics queries [HDKMS07], to perform root-cause visual analyses. To replay the real-time data, we add a replay slider, which is available to playback portions of the data at any time.

Evaluation Criteria

For the human, two important criteria for measuring the usefulness of the display techniques in monitoring applications are:

Constancy of Display

If large amounts of data are shifted, it is difficult for a user to keep track of patterns. Also, it requires the analysts to re-adjust the mental picture of the data on the screen. It can make analyzing the data more difficult. To measure the constancy of the display, we count the number of pixel changes per time unit. The lower this number, the higher is the constancy of the display.

Usage of Display Space

Using only small parts of the display leads to a distraction from the relevant information that is being displayed. Therefore, to help the analyst to quickly grasp the relevant information, large empty spaces in the display should be avoided. The higher the usage of the display space, the better is the display technique.

Please note that these criteria can only approximate the usefulness of a display technique. For instance, pixel coherence (neighbouring pixels that form a pattern and are therefore perceived as one object and not as single pixels) are not taken into account by the measures. Especially our measurement of the constancy of display is therefore a worst-case estimation of the quality. Depending on the data that is analyzed the problems caused by shifting may be better than our evaluation criteria suggest.

In the following, we first introduce and evaluate our circular overlay display technique in section 3 and compare it to a conventional period shift display. Then, in section 4, we present and evaluate our variable resolution density display. In section 5, we then discuss two application examples from

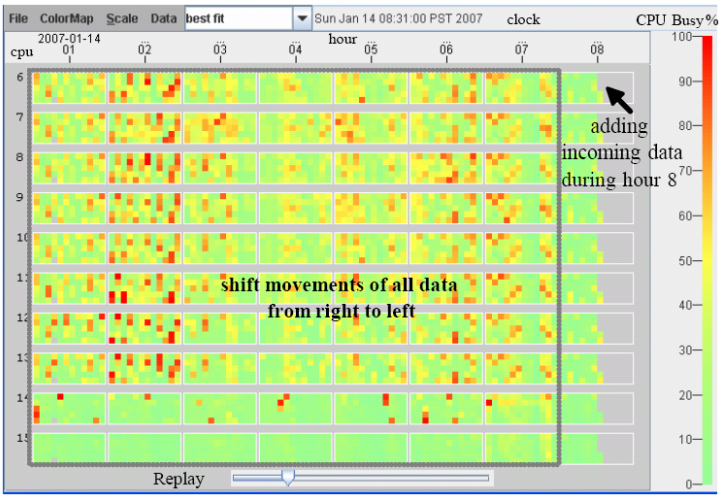


Figure 1C: *After the screen is full: The Complete Shift moves the previous time intervals to the left and drops the leftmost column (hour 0). The incoming data is added on the right.*

system performance monitoring and sale analysis and finally present an informal user study.

3. Constant Density Displays

3.1 Comparison of different techniques

Jump Shift

To visualize large time series data in real time, a simple method is to constantly add pixels at the right until the monitor is full and then shift the whole block of pixels x% to the left. Because this results in an abrupt change of all the pixels from time to time we call this "*Jump Shift*". The larger this jump is, the more empty space is gained and therefore the less often a jump has to be done. On the other hand a larger jump also means that more context is lost and that the empty space in the display is larger.

Complete Shift

A special case of the Jump Shift is to shift each pixel one position to the left each time a new item is added after the screen is full. Although this leads to a permanent movement of all pixels this technique can still be worthwhile to use. If the update rate is high enough the effect of a smooth animation may be achieved. However, this also means that detected patterns constantly move and have to be visually followed.

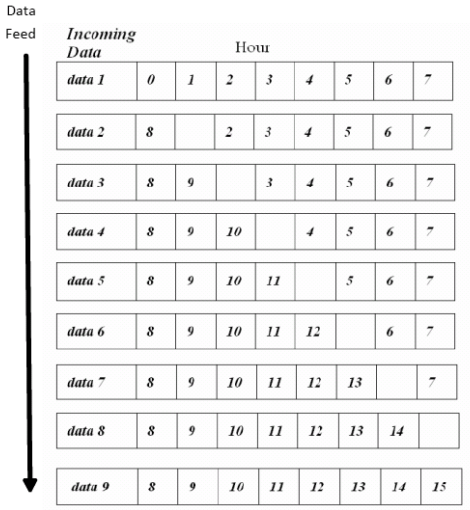
Circular Overlaying

To reduce the number of pixel shift movements, we use a circular overlay technique. When the screen is full, we overplot the oldest two data intervals with the background color and overlay the new interval data into the first of these areas. When the first area is full, we add a gap in the background color to separate the previous time interval data from the new data. This process is repeated until the screen is full again. The overall process of this technique is illustrated in Figure 2A and Figure 2B.

The circular overlay algorithm is as follows:

```
function(numberCols, pixelPerCell) {
    colCounter = 0;
    pixelCounter = 0;
    while (data) {          //read incoming data stream
        if (pixelCounter mod pixelPerCell == 0) { //column is full
            colCounter ++;
            if (colCounter mod numberCols == 0) {  //wrap around
                drawGap(1); drawGap(2);
            } else if (colCounter > numberCols
          & !( colCounter mod numberCols == numberCols -1)) {
             //middle column, neither the last column, nor one of
             //the first numberCols columns
                drawGap((colCounter +1) mod numberCols);
            }
        }
        drawData(colCounter, pixelCounter);
        pixelCounter++;
    }
}
```

In comparison to the Jump Shift the usage of the display space is better with Circular Overlaying and more context is provided (assuming reasonable percentages for the jump such as 25-75%). With respect to the amount of context that is provided, the Complete Shift would be the best one of course. How much context in fact is lost with Circular Overlaying and Jump Shift depends on the number of time intervals that are displayed in parallel representing the size of the jump. Since we never move any pixels but only overlay the oldest ones, the problem of abrupt large changes in the display (respectively the permanent movement of the Complete Shift) can be avoided. However, the advantages of the Circular Overlaying are counterbalanced by the loss of the intuitive left to right time line.



Circular Overlay Display over Time

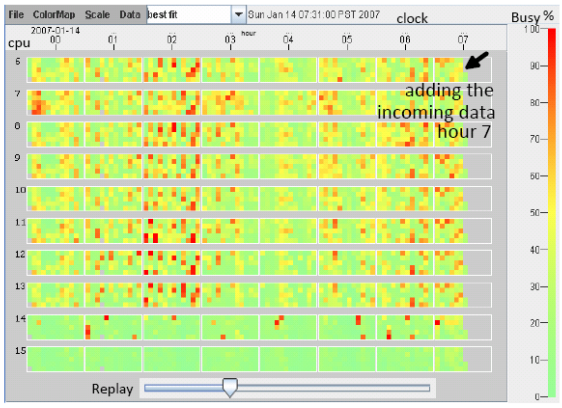


Figure 2A: *Before Circular Overlap*

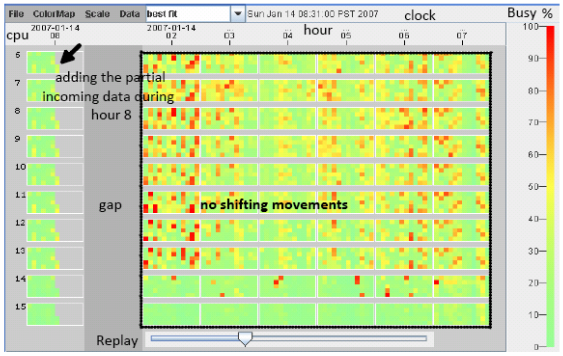


Figure 2B: *After Circular Overlap*

Overlay the oldest time interval (hour 0) with the incoming time interval (hour 8) and insert a gap at hour 1 to separate the new data from the previous data.

3.2. Evaluation of Constancy of the Display

Using the circular overlay technique, we can display large time series data in one single screen without shifting the data. There are two major advantages:

- retain most of the data at same display location to enable users to remember the already discovered patterns.
- improve system performance by reducing the number of shifting movements.

We decided to compare only the *Complete Shift* and the *Circular Overlay* technique and not to include the *Jump Shift* since its advantage of less changes is also provided by the *Circular Overlay* that additionally has the positive characteristic not only to minimize the movement of the pixels but to avoid it completely.

To compare the *Complete Shift* with our *Circular Overlay* technique, we measure the constancy of the display by determining the number of pixels that have been shifted up to time step *t*. The number of pixels that has to be shifted at time *t* is calculated as follows:

Complete Shift:

$$f(t) = \begin{cases} r & \text{if } (t \bmod p \neq 1) \vee (t \leq c * p), \\ c * r * p & \text{else} \end{cases}$$

Circular Overlaying:

$$f(t) = \begin{cases} 2 * r * p + r & \text{if } (t-1) \bmod (c * p) = 0 \\ & \text{//wrap around} \\ r * p + r & \text{if } t \bmod p = 1 \wedge \neg[(t-1) \bmod (c * p) = 0] \\ & \text{//go to next column} \\ r & \text{else} \end{cases}$$

where *c = number of columns*
r = number of rows
p = number of pixels per cell.

Plotting both functions over time for our example data

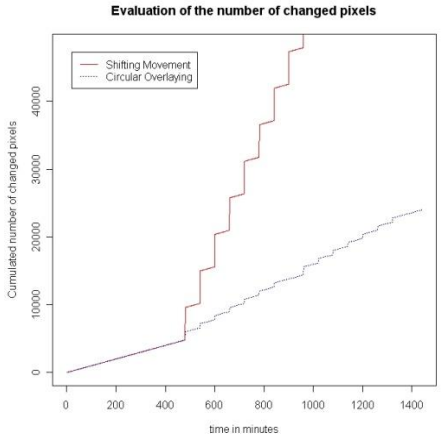


Figure 3: *Number of changed pixels over time. The plot shows the significant difference between the two algorithms.*

shown in Figure 2B (c = 8, r = 10, and p = 60) shows the significant difference of the two functions (see Figure 3).

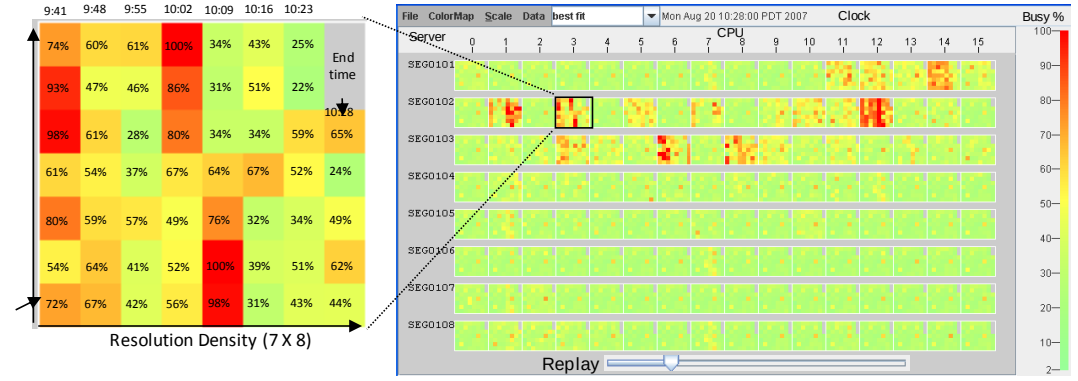
Using Complete Shift we have to change 91,040 pixels within one day (given one value per minute), but only 23,980 pixels when using our Circular Overlay technique.

4. Variable Resolution Density Displays

All display techniques discussed so far show a fixed amount of information. In this section we propose a variable resolution density display technique. Depending on the number of data to be displayed and the available screen space, the display automatically adjusts the resolution of a single data value. At the beginning of a time series, the number of records is low; therefore, the resolution density on the display is low. The resolution grows as more data are read in by the system. In Figure 4A, the user is looking at a low resolution density for the time frame 9:35 to 10:28, whereas in Figure 4B, the user is looking at more data in a higher resolution density display for the time frame 9:35 to 13:34.

4.1 Automatic Layout Algorithm

Given a fixed width and height for a cell the automatic layout algorithm determines at each step the best resolution for a data value. Starting with an initial pixel size as much



- Each cell represents a record showing *CPU Busy%* at 1-minute intervals (start at 9:35, end at 10:28).
- The color of a cell is the value of *CPU Busy %*.
- Cells are arranged from bottom to top and left to right according to the time values.
- The size of a cell changes according to the current resolution density.

Figure 4A: *Low Resolution Density (7x8) from 9:35 to 10:28*

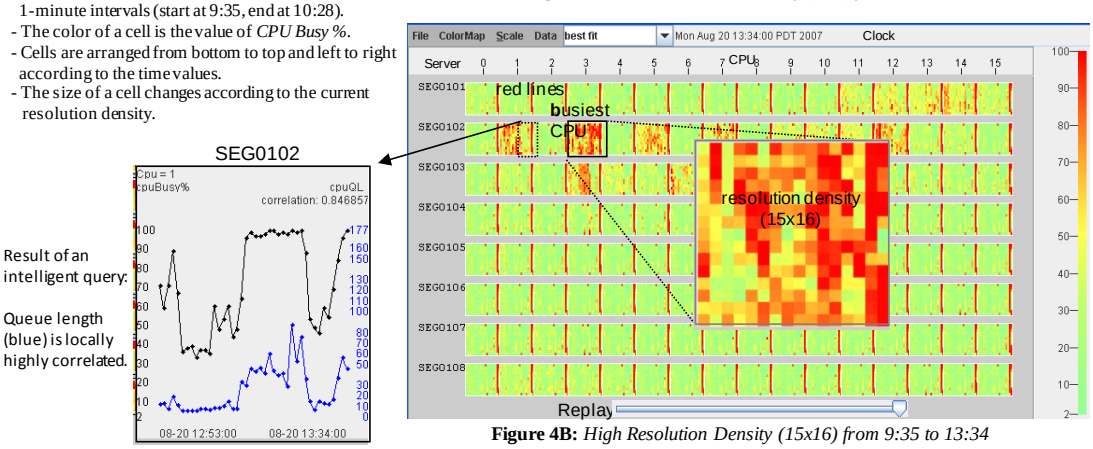


Figure 4B: *High Resolution Density (15x16) from 9:35 to 13:34*

Figure 4: *Two Variable Resolution Density Displays (16 CPUs x 8 Servers)*
Each CPU/Server contains variable numbers of measurement intervals from 9:35 to 13:34
(column: CPU (0, 16), rows: Servers (SEG0101-SEG0108); color: CPU Busy %)

incoming data as possible is added until the cell if full. Note that pixel size corresponds to the number of pixels used to present one data value. Then, the next resolution is determined in a way that it is smaller than the one before but still uses the display space as good as possible.

The automatic layout algorithm is as follow:

```
function(widthCell, heightCell) {
      pixelSize = detInitPixelSize(widthCell, heightCell);
      pixelPerCell = detNumber(widthCell, heightCell,
                                    pixelSize);
      pixelCount = 0;
      while (data)        //read incoming data stream
            if (pixelCount == pixelPerCell) {   //cell is full
                  pixelSize = detPixelSize(widthCell,
                              heightCell, pixelSize);     //see *
                  drawAllPrevPixels(pixelSize, pixelCount);
            }
      drawPixel(pixelSize, pixelCount);
      pixelCount++;
}
```

* we use prime factorization of widthCell and heightCell to determine the next pixelSize that is smaller than the last one and best exploits the space of the cell

4.2 Evaluation of the Usage of Display Space

The alternative to using multiple resolutions for the pixels would be to keep one specific pixel size during the whole process. In this case, as soon as the display space is full, we can do a periodic shift of all pixels, discarding the first one and adding the new data at the end (as discussed in the previous section). However, such shift operations are very expensive with respect to the number of changed pixels. By using pixels of size 1x1 we avoid introducing shift movements as long as possible but run into the problem that it takes a long time to get the display filled. Furthermore, decreasing the pixel size also decreases the visibility of the information units. This means that we have a trade-off between using as much of the display space as possible to display the information units versus avoiding too much change that forces the analyst to re-adjust the mental picture of the data on the screen.

To evaluate our approach and compare it to the simple *Complete Shift* (as described in the previous section), we again measure the constancy of the display by counting the number of pixels that have changed until time step *t*. We also measure the use of the display space by calculating the number of pixels that are used to display information.

For Figures 5 and 6, we calculated both evaluation measures for our example data set in Figure 4 with 16 columns, 8 rows, and a cell size of 64x60. In Figure 5 time is plotted against the number of used pixels. We can see that using our *Variable Resolution Display* technique (black curve) the display is quickly full and oscillates at high levels. In comparison, the *Complete Shift* algorithm using a pixel size of 1x1 (green curve) needs half of the considered time interval to get the display only filled half. The second half of the curve is the same as for our variable resolution display curve. When the *Complete Shift* is applied with a larger pixel size (e.g., 2x2, blue curve), the display is filled much faster and optimally uses the display space for the rest of the time but induces a high degree of shifts. By further increasing the pixel size we can even get steeper curves and thereby further improve the use of the display space but radically worsen the constancy of the display.

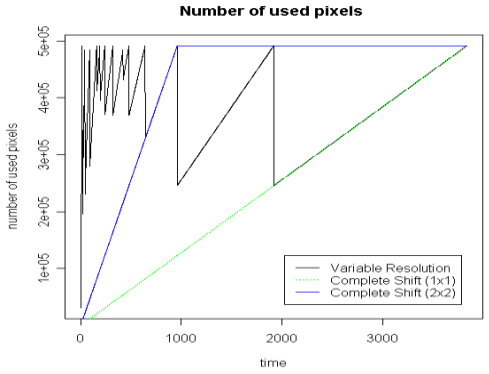


Figure 5: *Number of pixels that are used to display information. Our Variable Resolution Display technique is able to quickly fill the display and then oscillates at high levels.*

Figure 6 shows that increasing the pixel size that is used in the *Complete Shift* algorithm leads to a dramatic increase in the number of changed pixels. The blue curve, showing the *Complete Shift* for a pixel size of 2x2, needs to change about 170 times as many pixels in the observed time interval than the black curve, representing our *Variable Resolution Density* technique, which has a slope that is rather moderate. Our technique represents a good compromise for maximizing the number of used pixels while at the same time minimizing the number of changed pixels.

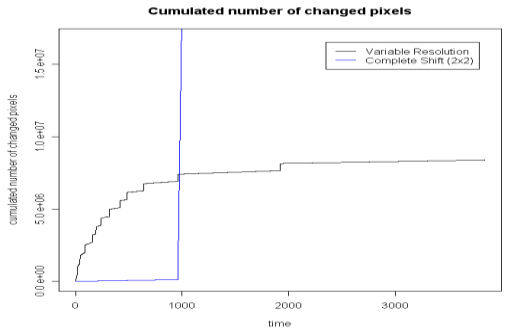


Figure 6: *Number of changed pixels. It can be clearly seen that the Complete Shift algorithm with a pixel size of 2x2 results in a very high number of changed pixels. Note that the blue curve linearly increases with the very steep increase, starting around time step 1000 and is two orders of magnitude larger than our Variable Resolution Display.*

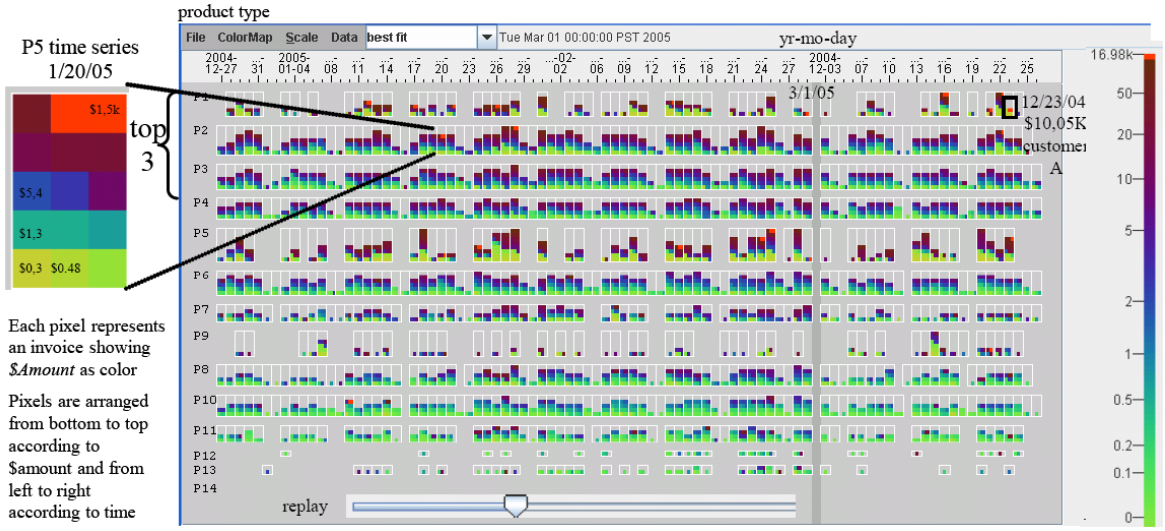


Figure 7: *Replay of product sales time-series (columns: yr-mo-day; rows: product type; color: $amount) Sales time-series from 12/3/04 to 3/1/05 to observe the 1st Quarter of 05*

5. Applications and User Study

5.1 Enterprise Data Warehouse CPU Performance

Our technique eliminates the limitations of common time series charts by enabling more data and more time intervals to be seen in a single display. Figure 4B, for example, shows a single view of the customer's performance data (8 Servers x 16 CPU). Users can visually compare the 128 time series to analyze changes, patterns, and exceptions at a glance.

To answer the question "Which system resource causes the CPU busy?", the user can rubber band an area and use a local correlation query to determine attributes which are closely related within the selected time interval (e.g., *Queue length*, *Disk I/O*, and *Memory Size*). This application combines the variable density display with our intelligent visual query approach [HDKMS07] (see insets in Figure 4). Users can also query from the rubber-band areas to visualize detailed information for further analysis. At time 13:34 in Figure 4, the user can instantaneously discover that CPU 3 in Server SEG0102 has the highest busy % (red, orange). The user can also quickly recognize a high CPU usage (red lines) across all CPUs for all servers.

5.2 Replaying Three Years Sales Data

For sales analysis, store managers would like to discover new patterns and relationships in the sales data. Examples of sales analyses that they typically need include the following:

1. How have our sales grown in the last three years?
2. Which are the three top products? What are their sales patterns?
3. What can we do to retain our top customers?

To answer the above questions, store managers need to replay their sales data. Figure 7 shows a snapshot of the replay sequence in which the first quarter sales data from 12/3/04 to 3/1/05 are visualized. The following information can be seen:

- The overall sales in 05 are much higher than the sales in 04 (more red and burgundy in 05, higher bars).
- The top three products are displayed in the first three rows, many high $Amount sales (red). P1 has the highest sales on 12/23/04 (marked).

5.3 User Study

To compare the circular overlaying display with a traditional shifting method, we conducted an informal user study with 9 participants from IT Services, Data Center, Finance, and Research Labs. The goal of this study is to evaluate both methods with respect to the following three tasks: (1) finding patterns, (2) finding anomalies, and (3) easiness of interaction. For the first two tasks we considered the factors: *display technique* x *visualized time range*. The dependent variable was time. Additionally, at the end of the study we asked the participants about their preferences.

The user study led to the following observations:

1. Data Stream Continuity

   When the last column is full the circular layout continues the visualization of the data stream in the first column which results in a break of the timeline. We observed that this causes difficulties for the users when two (in the timeline neighboring but in the display disjoint) columns have to be compared.

2. Screen Stability

   Some participants reported difficulties with the shifting method when the task required them to search for

patterns and follow them up. They appreciated the higher screen stability of the circular layout technique.

3. Task Completion Time

   The task completion times provided mixed results. The time measurements indicate that the users are faster with the circular display unless the task requires them to cope with the break in the timeline. It would be interesting to know whether the effect would be alleviated after some time of usage and an increased familiarity. We plan to address this question in a more detailed user study.

4. User Preferences

   After the study the users were asked to rate both methods for each task. Table 1 provides the average user preference ratings. The results indicate that there is a slight preference for the shifting method when the task is to find patterns in the data whereas to find anomalies the participants preferred the circular overlay as a display technique. Regarding the easiness of interaction with the display the circular overlaying technique also was superior to the shifting method.

   Overall, there are six users that prefer the circular overlaying display, two users that prefer the shifting display and one user who likes both displays.

Task	Shifting Method	Circular Overlay
Finding Patterns	2.5	2.35
Finding Anomalies	1.25	2.625
Interacting Display	0.875	2.25

Table 1: *The Average User Preference Ratings (Average, 3 is the best rating)*

Because our limited user study leaves a number of questions open, we consider doing a more detailed, quantitative user study with a larger number of participants in the future.

6. Conclusions

In this paper, we presented variable resolution density displays, a new method for visualizing very long time series data streams. The combination of the high density displays and the proposed circular overlay techniques enables users to see the "big picture" of real-time information without being overwhelmed by the constant shifting of the display. At the same time, users are able to drill-down to the detailed information and perform root-cause detection. Our real-time visualization of data warehouse and sales data will not only help customers in balancing the workloads but also increase their ability of quickly tracking invoices and identifying key deficiencies. In the future, we plan to develop other real-time visualization methods and refine our understanding of benefits and disadvantages by exploring data sets in a wide variety of applications. In addition, we will do further research to help the user to quickly discover interesting developments, for example, by an automatic highlighting of change points.

Acknowledgements

The authors wish to thank Mei Chun Hsu of HP Laboratories for her encouragement, T Ls Dutt for providing suggestions and data.

References

[CMS99] S. Card, J. D. Mackinlay, B. Shneiderman. *Information Visualization Using Vision to Think*. Morgan Kaufmann, pp. 307-309, 1999.

[ESS92] S. G. Eick, J. L. Steffen, E. E. Summer Jr. Seesoft-A Tool for Visualizing Line Oriented Software Statistics. *IEEE Transactions on Software Engineering.* 18(11): 957 - 968, 1992.

[EK00] S. Eick, A. Karr. Visual Scalability. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):44-58, 2000.

[FP02] J. Fekete, C. Plaisant. Interactive Information Visualization of a Million Items. *Proc. of IEEE Symposium on Information Visualization*, pp. 117-124, 2002.

[HDKMS07] M. Hao, U. Dayal, D. A. Keim, D. Morent. Intelligent Visual Analytics Queries. *IEEE Symposium on Visual Analytics Science and Technology,* pp. 91-98, 2007.

[HDKS07] M. Hao, U. Dayal, D. A. Keim, T. Schreck. Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. *Proc. of Eurographics/IEEE-VGTC Symposium on Visualization,* pp. 27-34, 2007.

[JS98] D. F. Jerding, J. T. Stasko. The Information Mural: A Technique for Displaying and Navigating Large Information Spaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3): 257-271, 1998

[MUN04] T. Munzner. BinX: Dynamic Exploration of Time Series Datasets across Aggregation Levels. *IEEE Symposium on Information Visualization*, 2004.

[MRC91] J. D. Mackinlay, G. G. Robertson, S. K. Card. The perspective wall: detail and context smoothly integrated. *Proc. of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology*, pp 173-176, 1991.

[SSJ05] A. Aris, B. Shneiderman, C. Plaisant, G. Shmueli, W. Jank. Representing Unevenly-Spaced Time Series Data for Visualization and Interactive Exploration. *Proc. of International Conference on Human-Computer Interaction*, pp. 835-846, 2005.

[WLS98] A. Woodruff, J. Landay, M. Stonebraker. Constant Information Density in Zoomable Interfaces. *Advanced Visual Interfaces 1998*, pp. 19-28, 1998.

[WS99] J. J. Van Wijk, E. R. Van Selow. Cluster and calendar based visualization of time series data. *Proc. of IEEE Symposium on Information Visualization*, pp. 4-9, 1999.

[YN06] B. Yost, C. North. The Perceptual Scalability of Visualization. *IEEE Transactions and Computer Graphics*. 12(5):837-844, 2006.