

DataWear: Revealing Trends Of Dynamic Data In Visualizations

Umer Farooq, Dilshad Akhter and Chris North
Computer Science Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061
{ufarooq, dakhter}@vt.edu, north@cs.vt.edu

Abstract

Visualizing data that varies with time differs from visualizing time-independent data – being able to perceive trends as time changes is vital in the former case, whereas in the latter, a reference to a prior visualization need not be mandatory. DataWear maintains the integrity of data with the passage of time by gradually wearing away old data, allowing trends to be visualized by comparing incremental levels of wear in the data. Using this concept of data wear, time does not need to be represented by an axis since wear levels implicitly convey the various time instances.

Keywords: Dynamic data, implicit parameter, fading effect.

1. Introduction

The visualization of time-independent data (that does not vary with time) such as observing the relationships between populations of countries and literacy rate is different from visualizing time-dependent data. In the above case, the point of interest lies at one instance of time, for example, the population of country X corresponds to its literacy rate Y at time T. While this serves as an important piece of information, it is equally imperative to visualize this case as a function of time. For example, literacy rate and corresponding population could vary with time for country X, and the point of interest shifts to the observation of these changing trends with time. DataWear attempts to visualize this type of data, namely time-dependent data, or more informally, dynamic data.

One obvious solution to visualizing dynamic data is to represent time explicitly as one of the axes, as can be achieved in SpotFire [1]. However, this solution

poses two hurdles. Firstly, it constraints the expression of other parameters. Considering the previous scenario of visualizing different attributes of countries, this solution will prevent one of the parameters (population or literacy rate) to be represented on an axis, and rely on alternative representations (such as a 3D representation). Secondly, as the range of time increases, its representation on the axis becomes squeezed and eventually indistinguishable, since the constant length of the axis will have to suffice for a greater quantity of values. DataWear proposes a solution to these problems and furthermore, this technique may be applied to any type of visualization that varies over time.

The remaining content of the paper is organized as follows. Section 2 sheds light on related work that resembles the visualization presented in this paper. Section 3 explicates the technique used to reveal trends of dynamic data in visualizations. Section 4 presents the actual visualization of a sample of dynamic data. Section 5 discusses some interactive features provided by the visualization tool. Section 6 reveals the implementation issues and the algorithm employed to produce the visualization. Section 7 discusses the visualization and its pros and cons. Section 8 throws up research issues and the potential future work. Section 9 presents the summary of this research effort and its contributions.

2. Related Work

There are a myriad of examples of graphical visualizations, such as time-series charts, that encompass the concept of the passage of time [2].

DataWear derives its concept from the Information Mural [3]. This technique supports creation of 2D visualization of large information spaces even when

the number of informational elements greatly outnumbers the available pixels. It creates a miniature version of the information space using visual attributes such as grayscale shading, intensity, color, and pixel size along with anti-aliased compression techniques. Information murals can be used for global views of information spaces. By adding panning and zooming within the global view itself, an information mural can be used as a standalone visualization.

The idea of computational wear in the domain of document processing has been described in [4]. The basic idea behind this is edit wear and read wear. Edit wear represents the edit history of the document while read wear represents its readership history. A technique called attribute-mapped scroll bars have been used to represent the wears onto the document scroll bars in positions relative to line positions. The geometry of the scroll bar is used to interpret the wear marks in relation to the structure of the document. This display technique reuses the precious screen space and also collocates information display with navigation control points.

3. The DataWear Technique

DataWear introduces the concept of representing time as an implicit parameter: a parameter that is not represented explicitly on an axis, rather through the visualization itself. An example of this could be worn paths in grass. As people traverse paths on a green grassy field, the surface tends to wear off gradually and reveal the soil, but as time increases, the grass grows back to retain its original integrity.

The use of time as an implicit parameter leads to the incorporation of a wear effect in the resulting visualization, similar to varying color saturation: the older the data are, the less saturated shade is used to represent them [2]. At the first instance of time, the data is represented with the highest level of grayscale intensity, that is, white. As time increases and data points move to new locations, their paths are highlighted using different levels of grayscale intensity - the latest path being highlighted with higher levels of grayscale intensities and previous paths with decreasing grayscale intensities. Therefore, trends in the data can be inferred by the relative grayscale intensities. It is to be noted here that the oldest data (starting from the first instance of

time) blends in with the black background after a certain threshold time. This technique avoids obscurity with an increased range of time, whilst maintaining a worn away trend of the data. By using this technique, more recent data, which in fact represents more important data, becomes obvious.

DataWear also arrests the problem of intersecting data points. Intersecting paths are important because they represent the paths that are traveled most frequently. Since worn paths represent the movement of data between points, intersections of these paths are likely. The intersected paths are distinguished from normal paths by higher levels of grayscale intensities.

The wear effect reveals how the data has changed over time, but preserves the primary view of the normal visualization of the current state of the data. This visualization is effective for seeing movement patterns at two levels of scale:

- *Micro level:* movement pattern of an individual data point. An example would be a comet trail. A comet moves from one point to the other, and the most recent path (between the last two points) is the most highlighted, whereas the previous paths wear away as the comet traverses distance with time.
- *Macro level:* movement pattern of many data points over a long period of time. An example of this is reflective of the previous one in which the surface of a green grassy field wears away with time as people traverse over it.

DataWear lays emphasis on the paths that are most recent and most traveled, which is the focus of interest in the visualization of dynamic data.

4. Examples of Visualization

4.1. Sample Data

The visualization is based upon stock market data consisting of four attributes with arbitrary ranges and units: *Share ID* ranging from 1 to 100, *price* of a share ranging from \$15 to \$114, *number of transactions* (number of shares bought or sold) ranging from 151 to 13426 shares, and *time* ranging from 1 to 100 days. The share IDs correspond to actual share names. Using our visualization tool on

this sample of randomly generated dynamic data, we attempt to reveal trends in this data and show the effectiveness of the DataWear concept.

4.2. Visualization Tool

The initial visualization is a black screen of spots with options on the right side of the applet. It displays the first day's data by default. Hundred spots appear (corresponding to hundred shares) on the screen, with x-axis as price and y-axis as number of transactions (figure 1).

To see how these data points move to new locations, see figure 2 for visualizing the state on the second day. The red spots represent the current state of the data points (day two). The data wear effect can be noticed since the starting path of each data point is colored with low grayscale intensity and the end path (near the red spot) is almost white. The low grayscale intensity paths will eventually wear away and blend with the black background, since these paths will be more insignificant as compared with the more recent data.

The effectiveness of this visualization is realized with a large increase in the number of days. Figure 3 depicts the visualization for day 46. Different levels of grayscale intensities can be seen, and trends identified. For example, in the price range from \$81 to \$99 and number of transactions from 10,500 to 13,375 shares (see top-right green bounding box in figure 3), it can be inferred that approximately ten shares are represented (ten spots) under this range category. Moreover, a lot of fluctuation has occurred in this range category, which from a shareholder's point of view, might imply that the lower bound for price is \$81, that is, whilst these ten shares fluctuate between the prices of \$81 and \$99, they do not go below \$81. Assuming that those shares do not incur heavy losses for that particular shareholder (since the lower bound of \$81 is bearable), these shares are not to be disposed off.

Consider a novice speculator who wants to enter the share trading business. He wants to experiment with shares at a low risk, and thus wants to invest in shares whose prices are low. Assuming that day 46 is the current instance of time, it is noticed that a lot of transactions (buying and selling) are taking place

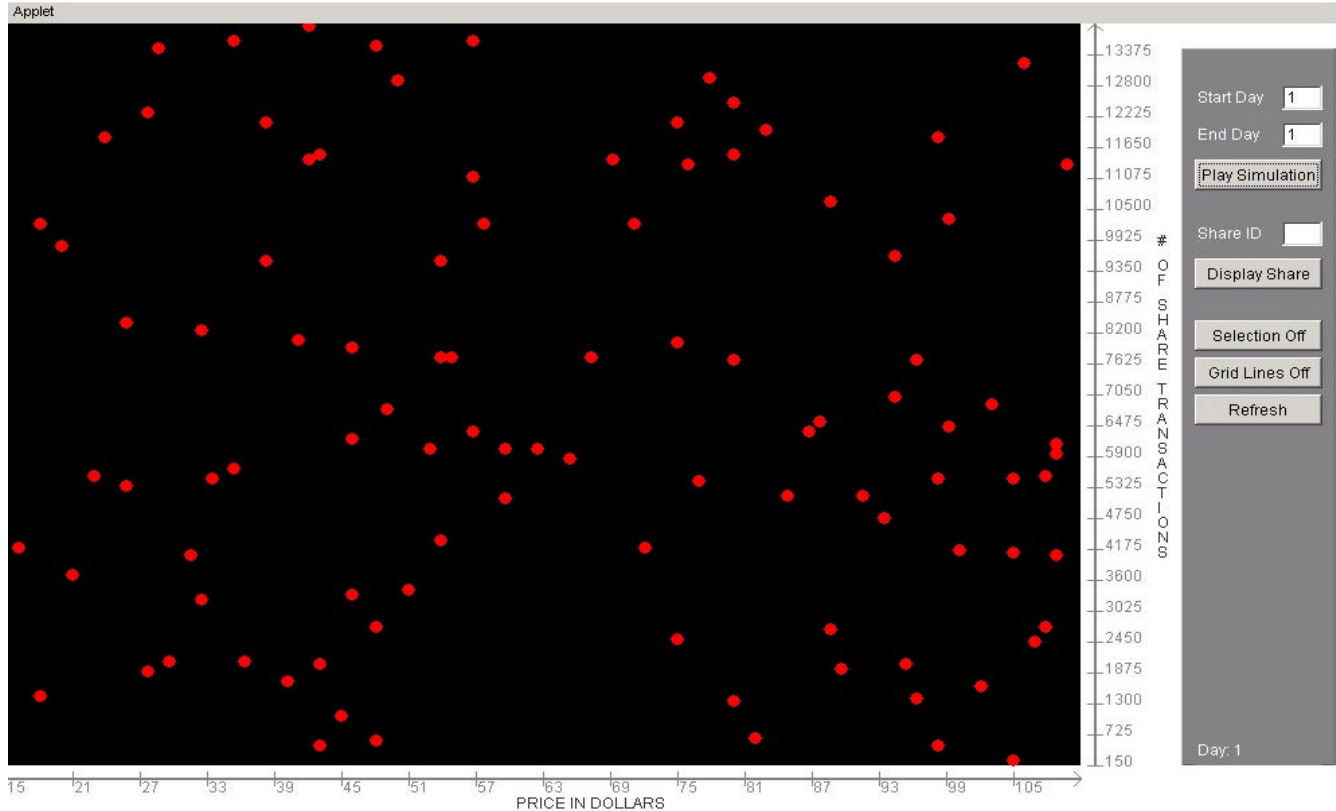


Figure 1: Visualization of stock market data for day 1

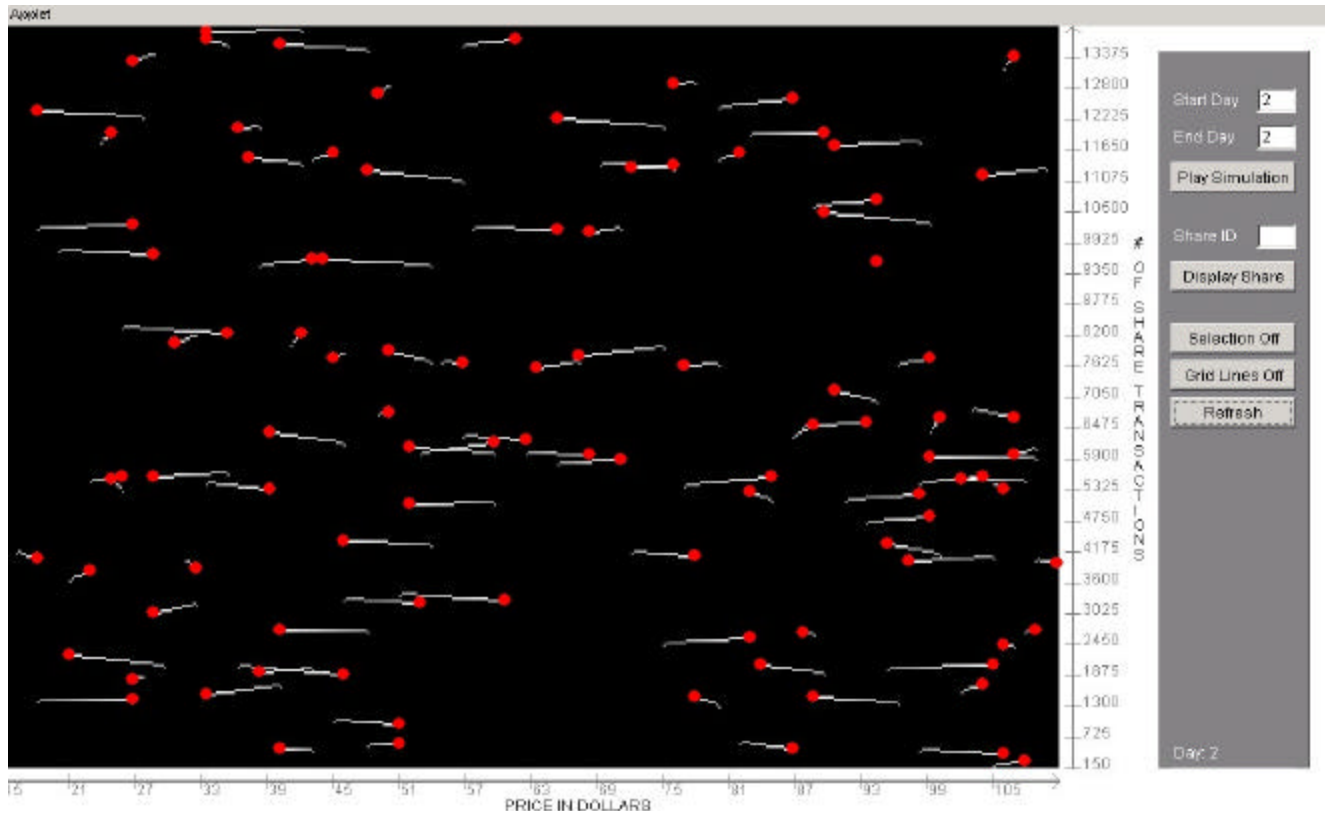


Figure 2: Visualization of stock market data for day 2

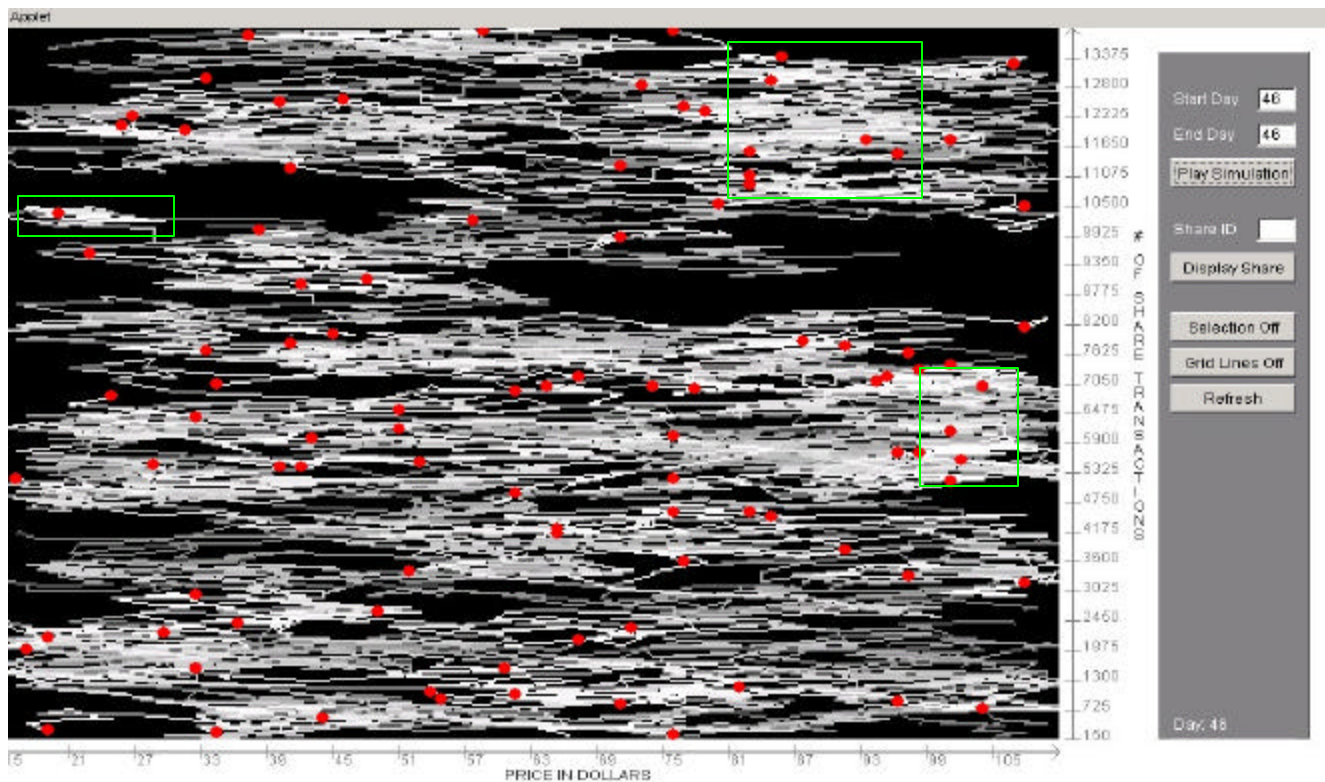


Figure 3: Visualization of stock market data for day 46

with the share in the price range from \$15 to \$27 and approximately 10,000 share transactions (see upper-left green bounding box in figure 3). Noticing that the share in this price range is being traded frequently, the novice spectator decides to invest his money by buying that share.

These snapshots represent one instance of time, and practically, the interactivity of the play simulation button is to be exploited. By specifying start day and end day, the play simulation button will animate the data, first displaying the start day's data and then gradually moving on to the next day's data until end day is reached. By playing the simulation from day 1 to 46, the shift in trends can be seen gradually. For instance, in figure 3, such a simulation revealed that the shares in the price range from \$99 to \$105 and number of transactions from 5325 to 7050 (see lower-right green bounding box in figure 3) were originally in a lower price range, and with the increase in time from day 1 to 46, the price has increased with almost no change in the number of transactions. This may be vital information to the shareholder, as one might interpret that even though the prices are rising, the number of share

transactions is stable, thereby indicating that shareholders are not selling the shares since the prices are expected to go even higher in future.

5. Interactive Features

This visualization tool provides other interactive features. If one is interested in particular shares, that is, the share ID (corresponding to the name of a share) is known, specifying the share ID and pressing the display share button will highlight those shares. Figure 4 shows shares 73, 37, and 50 highlighted. The green color also varies in intensity from low to high corresponding to old and new data respectively. Moreover, the labels for share IDs are indicated on top of the starting point of the share on the first day, so the trend can be observed over the whole range of days. Pressing the refresh button will bring the visualization to the original state (without the highlighted shares).

If one wishes to know the share ID of a particular share from the visualization, the selection button may be used. By pressing the selection button, it goes into the selection on mode, and by moving the

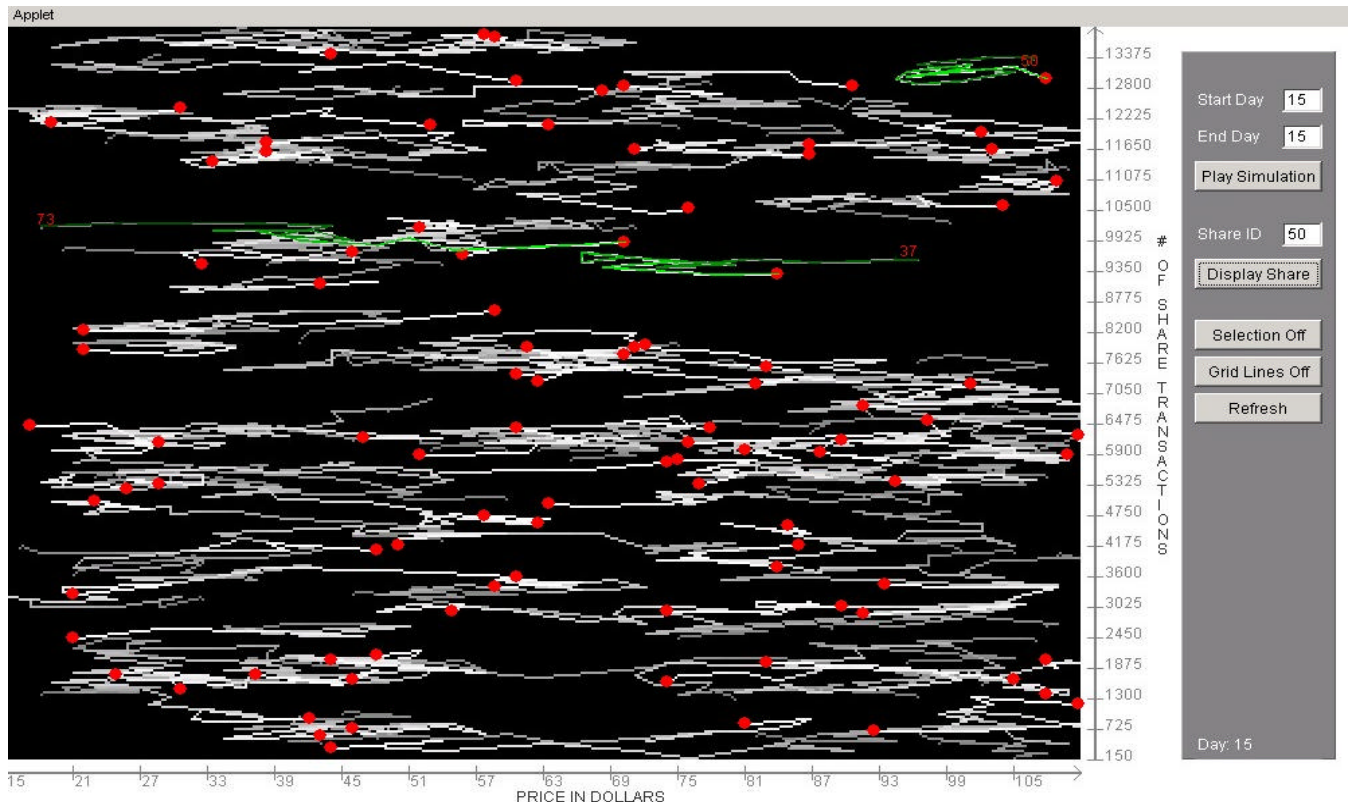


Figure 4: Shares 73, 37, and 50 highlighted

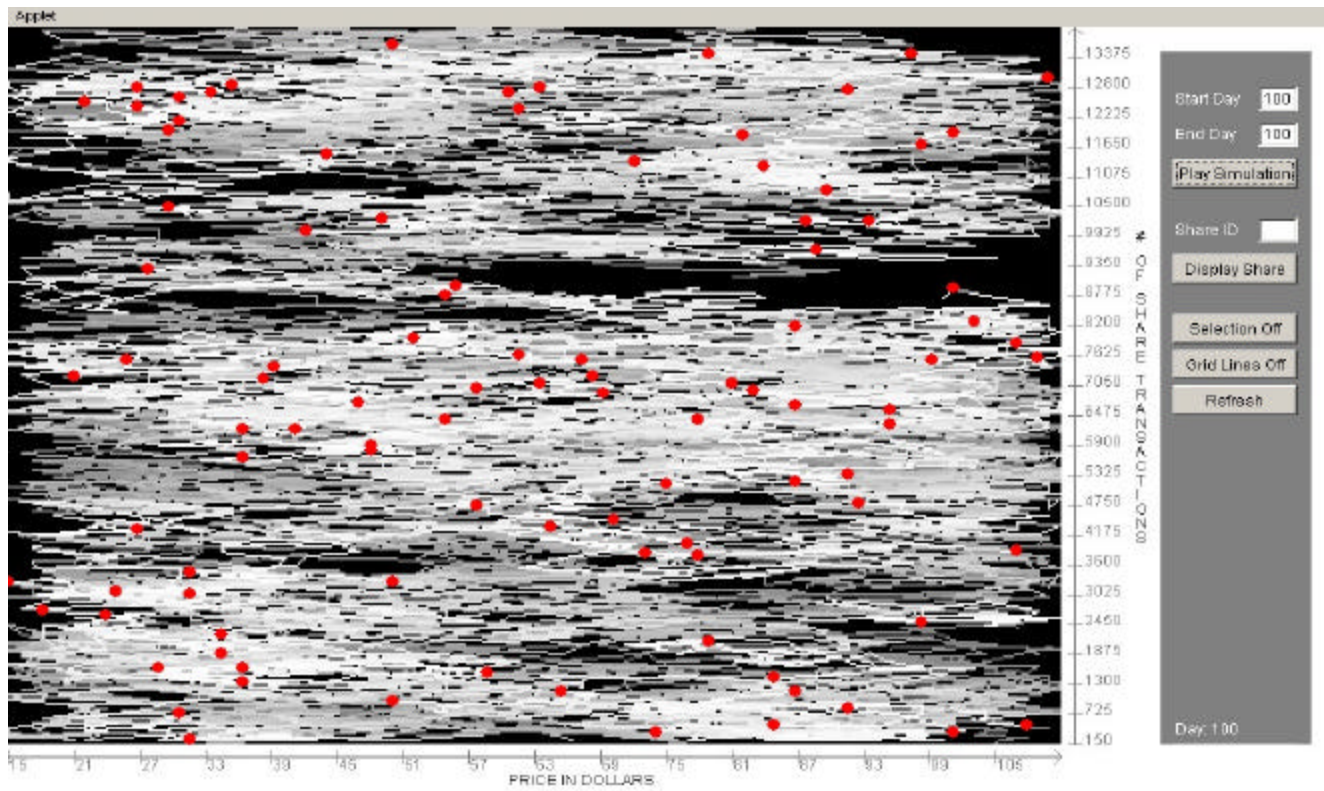


Figure 5: Visualization of stock market data for day 100 (without grid lines)

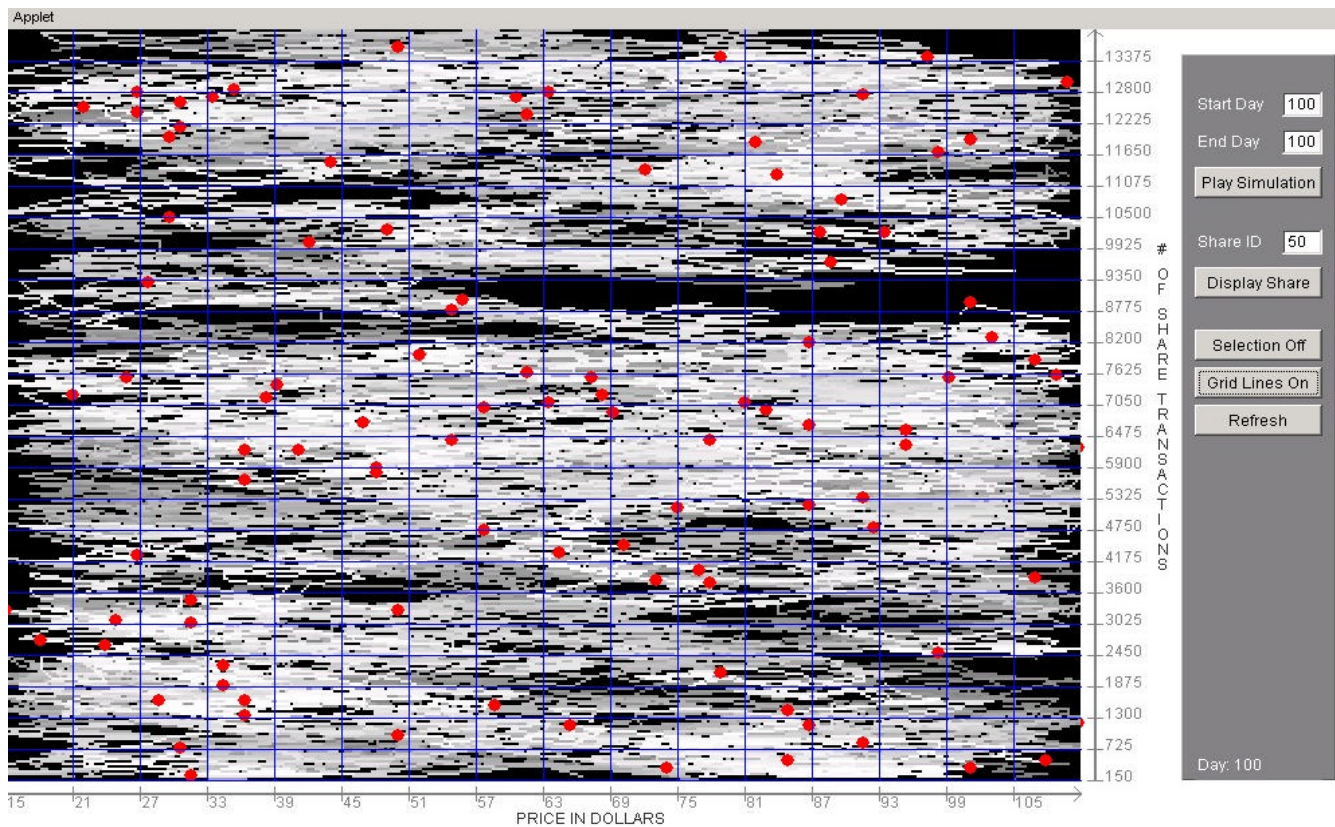


Figure 6: Visualization of stock market data for day 100 with grid lines

mouse over a data point, the respective share will be displayed in the share ID textbox. Displaying this share will highlight the movement of this share from day 1. Pressing the selection button again will revert back to the selection off mode.

For finding out specific values of shares, the grid lines option is provided. By pressing the grid lines button, lines are drawn for the values on the x and y-axes. Figure 5 and 6 represent the visualization for day 100 without and with grid lines respectively. This allows the user to match the data point to an exact value corresponding to the axes. Pressing the grid lines button again removes the grid lines.

6. Implementation Issues

6.1. Algorithm for the wear effect

The data consists of a number of data points (D1, D2, ...). Each data point D1 (x1, y1) is accessed from a database and normalized to get a corresponding pixel point P1 (x1, y1). As time increases, a data point may move from D1 (x1, y1) to D1 (x2, y2). The position of the data point D1 (x1, y1) at the next instance of time, which is D1 (x2, y2), is also fetched from the database and normalized to get pixel point P2 (x2, y2). All the pixels on the line P1 P2 are calculated. For each pixel on this line, in descending order from P2 to P1, the increase in weight is calculated by normalizing the total number of pixels of the line to the weight of ascending range W1 to W2. Initially, all pixel points have a weight of zero. While increasing the weight we checked to see if the amount exceeds the maximum weight at that step. If so, then we chopped the weight to that maximum value. After assigning the weights to all data points, the pixels' weights are normalized to get a corresponding color value in gray scale. An image of dimension I x J is created from all these pixels and a scaled version of this image is then mapped onto the applet screen of dimension I*2 x J* 2.

One of the major problems faced during the early implementation is that large number of intersections occluded the current day's data. We minimized this problem by keeping the weight of intersecting points lower than the maximum possible weight values of that time. This is done by checking through the "if condition" as given in the pseudo-code of the

algorithm. The wear effect algorithm is outlined below.

```

For each instance of time {
  For each data point {
    - Normalize to get the
      corresponding pixel P1 (x1, y1)
    - Get the position of the same data
      point at the next instance of time
    - Normalize this data point to get
      corresponding pixel P2 (x2, y2)
    - Get all the points between P1 and P2
      For each point from P2 to P1 {
        - Get the weight by
          normalizing the number
          of pixels to the weight
          range of W1 to W2,
          which is in descending
          order
        - Add this calculated
          weight to the current
          weight of the pixel
        - If weight > starting weight + W2
          Weight = starting weight + W2
      }//end for
    }//end for
  For each pixel in the image {
    - Normalize its weight against
      color range
    - Set the pixel to that color
  }//end for
} //end outer for

```

6.2. Performance of algorithm

Since the randomly generated data is stored in a sequential manner and not indexed, the interaction with the visualization tool is not real time. For this reason, preprocessing on the data was performed, in which indexed data was generated for visualization. The performance of the algorithm depends on the number of data points, time for which the data is to be displayed, the path traversed by each data point (from P1 to P2), and the width and height of the screen where data is to be displayed.

6.3. The underlying architecture

The visualization is presented in a Java applet. The software framework is a two-tiered client server application. The data is stored in a remote database.

The applet communicates with a server (on the same machine as the database) using Remote Method Invocation (RMI) to get the data from the remote database. The server in turn gets the data from the database using Java Database Connectivity (JDBC).

7. Discussion

The visualization effectively represents trends in dynamic data. The visualization tool simply provides an initial implementation framework for exemplifying the concept of wear. With its current features and advantages, there are a number of drawbacks and other important interactive tools that need to be incorporated.

There may be a conflict in the representation of the latest data and the paths that intersect. Since both visualizations are represented with higher levels of grayscale intensity, it could potentially create misinterpretations. One solution to this is to use two different colors for representing normal and intersecting paths. For example, normal paths could be represented in red intensities, and intersected paths with blue intensities. As the intersection increases on a specific path, an increased blue intensity will result in a mixture with the red color, thereby giving a purplish effect. More the intersections, brighter the intensity of purple. The trends can easily be visualized since color representations for normal and intersected paths are distinct.

The time before a path wears away completely is fixed. A customizable wear threshold should be provided to the user for specifying the time duration for a normal path to completely wear away. Even with such a facility, zooming needs to be incorporated to give accurate and detailed information about the individual data points.

8. Future Work

In addition to resolving the weaknesses of the data wear concept outlined in the previous section, there are features that need to be incorporated to the visualization tool itself. The visualization tool should provide an interface for the user to visualize any data using the concept of DataWear. The user would connect to his or her own database using the

interface, provide the appropriate fields to be displayed, their ranges, constraints, and the time duration, resulting in a customized visualization of the dynamic data to reveal trends. Since the application is fully integrated with the Internet using an applet, the user will not be required to download any application, and major load of the work shall take place on the server side.

9. Conclusion

This paper presents the concept of visualizing trends in dynamic data using the wear effect. With the passage of time, most recent data and traveled paths are highlighted using higher levels of grayscale intensities and older data faded away using lower grayscale intensities, which gradually blends in with the black background. The visualization tool is just a demonstration of this concept. It can be used in any static data visualization for revealing the effects of dynamic data by showing the wear effect in the background of the visualization.

Acknowledgements

This work was accomplished during the CS 5984 (Information Visualization) course executed in Spring 2001 at Virginia Polytechnic Institute of State University by Dr. Chris North.

References

- [1] Ahlberg, C., Wistrand, E., "IVEE: An Information Visualization and Exploration Environment", *Proc. IEEE Information Visualization '95*, pp. 66-73, (1995).
- [2] Andrienko, N., Andrienko, G., Gatalsky, P., "Supporting Visual Exploration of Object Movement", *AVI 2000*, pp. 217-220.
- [3] Jerding, D., Stasko, J., "The Information Mural: A Technique for Displaying and Navigating Large Information Spaces", *Proc. IEEE Symposium on Information Visualization*, pp. 43-50, (October 1995).
- [4] Hill, W. C., Hollan, J. D., "Edit wear and read wear", *ACM CHI '92*, pp. 3-9.