A Visualization Environment for Multiple Daytime Stock Price Predictions,

Yoshihiko Ichikawa (ichkawa@is.ocha.ac.jp),
T. Tsunawaki (tsuna@imv.is.ocha.ac.jp)),
I. Fujishiro (fuji@is.ocha.ac.jp),
Department of Information Sciences, Ochanomizu University,
H. Yoon (),
CMD Research

September 2002

Y. Ichikawa, T. Tsunawaki, I. Fujishiro, and H. Yoon "A Visualization Environment for Multiple Daytime Stock Price Predictions," in Proceedings of the 2nd VIIP International Conference on Visualization, Imaging and Image Processing (VIIP2002), Malaga, Spain, September 9-12, 2002.

Yoshihiko Ichikawa, Tomoko Tsunawaki
Department of Information Sciences, Ochanomizu University
e-mail: ichikawa@is.ocha.ac.jp, tsuna@im.is.ocha.ac.jp

Issei Fujishiro
Graduate School of Humanities and Sciences, Ochanomizu University
fuji@is.ocha.ac.jp

Hiwon Yoon
Keio University & CMD Research Inc.

Abstract

Visualization of stock exchange market is one of the prominent applications of information visualization. Although there are several research results which visualize long-term stock price movements, visualization of daytime stock price movements is rare. However, it is a very important application domain, since the fact that there is no statistical method of predicting daytime stock prices indicates that the decision making process must be supported by tools for cognitive amplification. Even with the advent of agent-based simulation techniques, it is still a challenging issue because different parameters result in diverse predictions, and so the traders are forced to read multiple predictions quickly. In order to provide the simultaneous presentation of multiple predictions, we have built a system which supports line-charts with a cluttering detection and control mechanism, color-charts with Level-of-Detail control, and a workspace for overviewing the different prediction sets simultaneously. Although it is difficult to evaluate the system quantitatively, the system is novel in the sense that other commercial systems still use simultaneous display of windows each of which shows just a few charts. Since even trained traders cannot scrutinize more than ten windows, providing overviews of dozens of price changes is a very important function of the systems supporting financial traders.

Keywords: *Information visualization, stock price prediction, focus+context view, level-of-detail control.*

1. Introduction

Visualization of stock exchange market is one of the prominent applications of information visualization, and there are several research results that visualize daily or longer movements of stock prices. However, visualization of daytime stock price movements is still a challenging issue, since there is no statistical method of predicting daytime stock prices, and it indicates that the decision making process must be supported by tools for cognitive amplification.

With respect to prediction and modeling of daytime stock exchange activities, recent research efforts in agent-based artificial markets have made them possible to some extent [1]. For instance, the agent-based simulator developed at Keio University and CMD Research Inc. [2] can predict the daytime stock price movement, given the initial states of the agents and parameters regarding the stock itself and the market. The simulator models a daytime price movement by a sequence of states of the agents. The initial state of a prediction is computed from the final state of the sequence which models the preceding price movement. Even with non-statistical methods of predicting stock prices, there are many parameters, and different parameter settings result in drastically diverse predictions. Hence, the traders still have to take into account the multiple results at the same time, and need effective decision-support tools.

Figure 1 depicts a line chart showing twenty-two predictions generated by the simulator mentioned above. It is easy to see that occlusions occur at several areas and the details of each prediction are difficult to read. So, the first purpose of our study is to provide a focus+context view of a set of predictions based on the line chart. Note that every simulation in the set shows one of the possible price-changes of a particlar stock. So, the word “trend” might be more appropriate. We use “prediction” here to avoid unnecessary confusion. Note also that the color figures are available via the following URL:
http://pfp7.is.ocha.ac.jp/˜ichikawa/VIIP02/color-plates.html

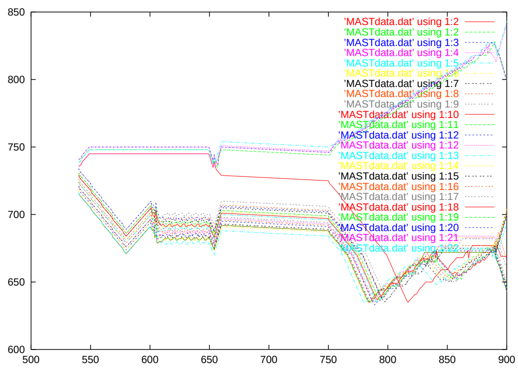

Figure 1. A line chart showing 22 stock-price predictions

Figure 2 shows the color chart of the price-changes of the same set of predictions. Each price-change is represented by a rectangular region colored according to the value. Negative and positive changes are colored blue and red, respectively, while green indicates no change. The intensity of the region in the display represents the absolute value of the change. The horizontal position and width represent the time period between the compared trades, and the vertical position corresponds to the prediction. All color-bands are aligned so that they share the same time axe. Being compared with the chart in Figure 1, it is easier to see the trends of each prediction. However, when the available vertical space is limited or the number of predictions to display increases, the color chart becomes harder to read than the line chart. Hence, the second purpose of this study is to provide the color chart with Level-of-Detail control, so that according to the available horizontal space or the needs of the user, some vertically aligned predictions are merged to depict the overall trends of them.

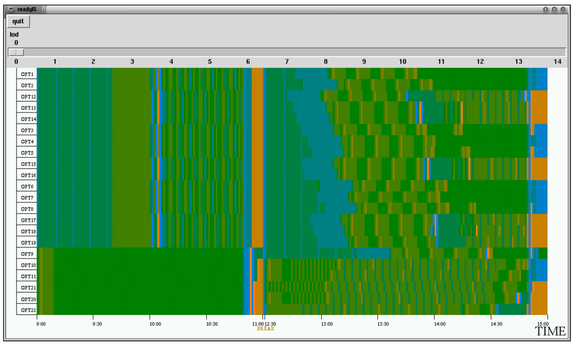

Figure 2. A color chart showing 22 price-change predictions

As mentioned above, the predictions depend heavily on the simulation parameters. Hence, we also provide a workspace in order for the user to be able to browse the visualization results of multiple prediction sets and to see the difference attributed to the parameters.

The rest of this paper is organized as follows. The next section explains related work and the reason why we do not want to use three-dimensional visualization. Section 3 discusses the focus+context view of line charts. Section 4 provides the LoD parameter of color charts, and how a color-band is merged into a texture. Section 5 then explains the workspace for providing an overview of multiple prediction sets simultaneously. The last section concludes and also addresses future work.

2. Related work

One of the earliest attempts to visualize financial market data is *worlds within worlds* [3], where the function of six variables representing *European options* is visualized. The system uses overloaded visualization structure, i.e., three-dimensional orthogonal spaces are embedded into a global similar space. More precisely, with one parameter value fixed, the three axes of the global space are mapped to the first three variables. Each point in this space has a three-dimensional orthogonal space where the two axes are mapped to the rest of variables, and the third axis is mapped to the function value. In the internal visualization space, the function values are visualized as a surface. Since all of the internal spaces cannot be shown simultaneously, the system allows the user to selectively make them visible. By navigating the visualization space and selectively making internal spaces visible, the user can compare various options.

Another example of visualization of financial market data can be seen in [4], which proposes the concept of information animations. The paper shows that a combination of three-dimensional visualization space, well-designed glyphs and their layouts[1], and interactive navigation provides comprehensible visualization of a huge number of data regarding Toronto stock exchange market.

There is an apparent trade-off between the use of three-dimensional visualization space and quick decision making. A perspective view of a three-dimensional space shows much more objects simultaneously. However, occlusion inherent to the perspective views requires the user to interact with the system in foraging necessary data. Figure 3 shows our previous work visualizing daytime stock price changes in a three-dimensional space [5]. Due to the occlusion, the user must move the probe surfaces in finding the details of the prices. In this paper, therefore, we adhere to the use of two-dimensional visualization space.

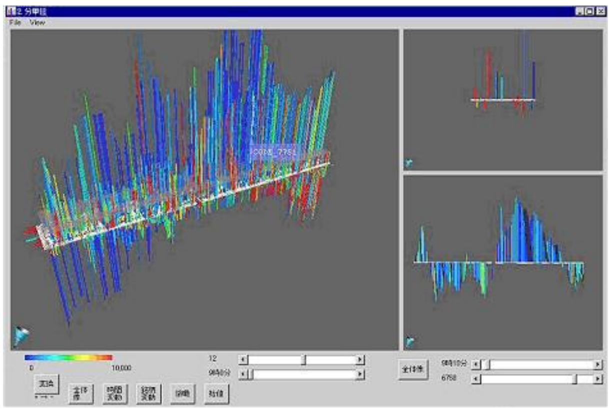

Figure 3. Three dimensional visualization of daytime stock prices.

VisDB [6] is a pixel-oriented visualization system. In the paper, time series of stock prices are visualized. It is easy to find a pattern inherent in the data set, while the details of the data are difficult to see.

Another interesting attempt to visualize stock prices is reported in [7]. In the paper, a technique converting a color-band into a texture is used to provide the overview of a large number of stock prices. The conversion process is depicted in Figure 4. Note that in the figure, the left color-band is considered to have three by eight pixels for

[1]The author of the paper uses the term “sign” instead of “glyph”.

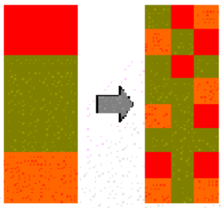

Figure 4. The conversion of a color-band into a texture

brevity. The image on the right-hand side is generated by randomly moving pixels comprising the left image without overlapping. The ratio of the area covered by pixels of a particular color to the overall area in the generated image is equal to that in the original color-band. In addition, this property almost holds even if a sub-area of the generated image is chosen or the generated image is juxtaposed repeatedly. The generated texture can be mapped to a tiny glyph, and so a large number of categorical data can be displayed in a restricted space. Since the texture can be mapped to any shape, generated texture can be used in topographic maps.

3. Line charts with cluttering control

The line chart shown in Figure 1 is difficult to read due to cluttering and overlapping. Since these are different phenomena, our system treats them differently:

- How many predictions are overlapped is represented by the color of line segments.
- The periods of time with occlusion occupy more horizontal space than the original one.

The resulting image is shown in Figure 5. Several periods of time after noon are enlarged so that the details of the line charts can be read easily. To make them salient, the background of enlarged areas is colored pale sky blue. Figure 6 shows part of another improved line chart which well illustrates how the colors are used to represent the degree of overlapping. In the implementation, we use red, yellow, green, blue and black for the colors of line segments. The degree of overlapping decreases in this order.

The visualization algorithm can be split into three steps:

(1) for each point of time t, make a histogram h_t of predicted stock prices;

(2) for each point (t, p) such that $h_t(p) \neq 0$, draw a filled rectangle centered on (t, p) on a virtual gray-scale drawing space. The intensity of the pixel of the rectangle decreases gradually from the center to the edges. In drawing the rectangle, the intensity of a pixel of the canvas is increased by the intensity of the corresponding pixel in the drawn rectangle;

(3) for each pixel at (t, p) in the virtual canvas, check if the intensity is greater than a given threshold.

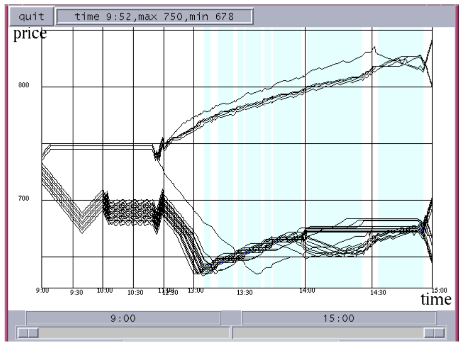

Figure 5. An improved line chart: overlapping is represented by the color attribute, and the periods of time with cluttering occupy more horizontal space than usual.

Every period of time starting from such a point to the next trade point of time occupies more horizontal space. The color of a line segment starting from (t, p) is determined based on $h_t(p)$.

The threshold value and the ratio of an enlarged horizontal space to the normal space can be interactively changed. The algorithm resembles the technique used in Information Mural [8]. In our algorithm, however, the virtual canvas is used for determining the degree of cluttering and is not rendered directly.

Other interactions such as zooming and panning are supported by providing the sliders through which the user can specify the visible period of time.

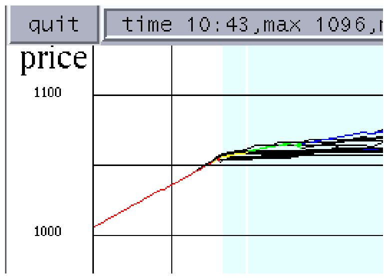

Figure 6. Another example of an improved line chart: overlapping is represented by the color attribute of line segments. Only relevant part is shown.

4. Color charts with LoD control

The color-chart shown in Figure 2 is composed of horizontal color-bands each of which corresponds to a particular prediction. Since all the predictions in one set share the trade points of time, we can consider that the chart is composed of vertical color-bands each of which is associated with a particular period of time. By applying the method depicted in Figure 4, these color-bands can be converted into textures. The resulting image is shown in Figure 7. The generated image can be shown with much smaller horizontal space, due to the inherent property of textures (see Figure 8).

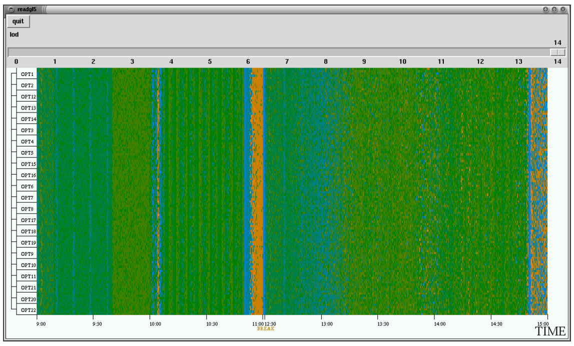

Figure 7. A texture band generated by the conversion method shown in Figure 4.

Figure 8. A horizontally cropped part of the texture band shown in Figure 7.

The difference between Figures 2 and 7 is “drastic” in the sense that individual properties have completely diminished in Figure 7. However, if we can hierarchically group the predictions, then we can get LoD control of the conversion by selectively applying the conversion process to the groups according to a particular control parameter.

In order to have a clustering hierarchy, our implementation utilizes the clustering program distributed with XmdvTool [9]. The program is based on Burch's clustering algorithm which uses the Euclid distance. In the first step of the algorithm, for each prediction, a cluster containing the prediction is constructed. Then, in the next step, the algorithm gradually and repeatedly merges near clusters into a new cluster until the root cluster is generated. The result is a hierarchy of clusters, each of which is associated with the radius of the cluster. Every leaf cluster has the radius of size zero, and is associated with one of the predictions.

By traversing the hierarchy, the predictions can be listed in a way that, for every cluster, all its members are listed contiguously. Figure 2 lists the predictions according to this order from top to bottom. It is easy to observe that there are two clusters of level one. The one is composed of upper seventeen predictions, while the other is composed of the rest.

The radii of the clusters are used for LoD control. That is, every cluster whose radius is smaller than a particular threshold is displayed as the generated texture-band instead of a contiguous list of color-bands. The display algorithm in a C-like pseudo language is shown in Figure 9.

Figure 10 shows the mixture of texture-bands and a color-band. How predictions are grouped is indicated in the area near the left periphery. In this chart, there are three clusters: the cluster composed of top-most ten predictions, the one composed of the next six predictions, and the one composed of the bottom five predictions.

```
/*
 *Globals:
 *  cls is an array recording clusters;
 *  n is the number of clusters;
 *  ry is the horizontal drawing position;
 *Input: idx is the id of the cluster.
 */
void dft_display(int idx)
{
  int   i,r;

  /* case 1: the cluster is a leaf */
  if (cls[idx].np == 1) {
    /* draw he color-band of the prediction */
    display_line(cls[idx].id+1, ry++);
    return;
  }
  /* case 2: the cluster radius is smaller
   than the control parameter */
  if (cls[idx].rad < lod) {
    /* convert the cluster into to a texture-
       band.  The function returns the number
       of the included predictions */
    lf = conversion(idx);
    /* draw the texture-band */
    display_texture_band(ry);
    /* move the drawing position */
    ry += lf;
    return;
  }
  /* case 3: otherwise, recursively apply the
      routine to each of the children */
  for (i = 0; i < n; i++)
    if (cls[i].pid == cls[idx].id)
      dft_display(i);
}
```

Figure 9. The display program of predictions of price-changes in a C-like pseudo language.

These are shown as texture-bands, while the prediction named OPT9 is shown as a color-band.

5. A workspace

The visualizers described above can display one set of predictions at a time. However, in practice, the user wants to view several predictions simultaneously in order to see the effects of changing parameters. The agent-based simulator uses thirty-three parameters in total, but many of them are determined by the stock in question, the market where the stock is traded, or the date of trading. So, the workspace we have built allows the user to change the stocks, the position of selling investor agent (INVSPOS), and the position of the buying investor agent (INVBPOS). We have chosen these parameters, since the parameters regarding the market-maker agent and the day-trader agent are easier to guess by the trader. For each set of parameters, the simulator generates a set of twenty-two predictions, which is visualized by either of the visualizers.

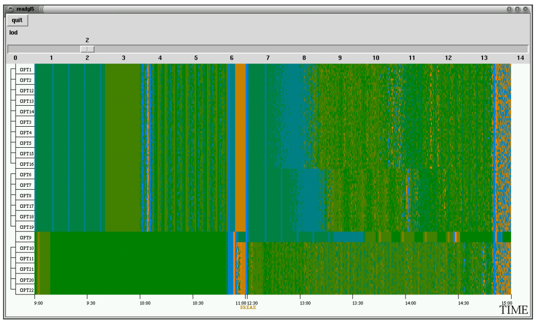

Figure 10. A color-band display when the LoD parameter is 2: three non-leaf clusters are depicted as text-bands, and one leaf cluster is depicted as a color-band.

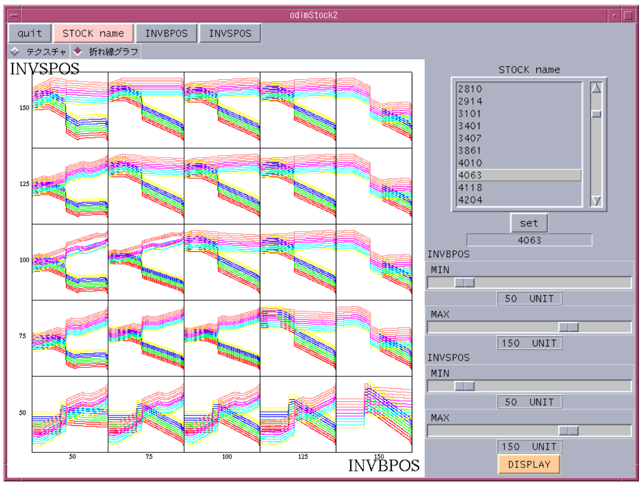

Figure 11. A snapshot of the workspace

Figure 11 shows the snapshot of the workspace, where the user specifies one stock and the ranges of INVBPOS and INVSPOS. The design of the workspace is based on dimensional stacking [10]. Hence, the visual structure of the figure is as follows: the outer horizontal and vertical axes are mapped to INVBPOS and INVSPOS respectively, and the inner horizontal and vertical axes are respectively mapped to time and the stock price variable.

The mapping of the outer axes can be changed by clicking the buttons located at the upper-left area. In the snapshot shown in Figure 12, the visualizer used to generate the thumbnails is the same as the previous one, while the outer horizontal and vertical axes are mapped to INVSPOS and the stock variable, respectively. Below these buttons, two radio buttons are located so that the user can change the visualizer. In the snapshot shown in Figure 13, the texture-band visualizer is chosen instead of the line-chart visualizer. In this image, sixteen texture-bands representing 352 predictions in total are shown. The parameters of the visualizers are not manipulated in this workspace. Instead, when the user clicks any of the "thumbnail" visualizations, the actual visualizer pops up and begins interaction with the user.

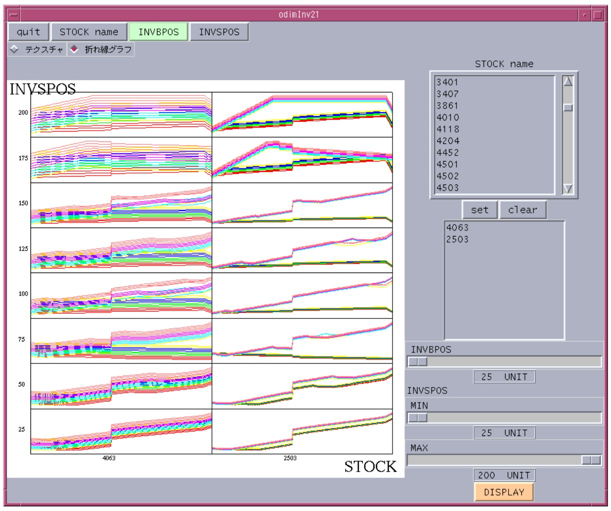

Figure 12. A snapshot of the workspace: INVSPOS and STOCK are mapped to the outer vertical and horizontal axes respectively.

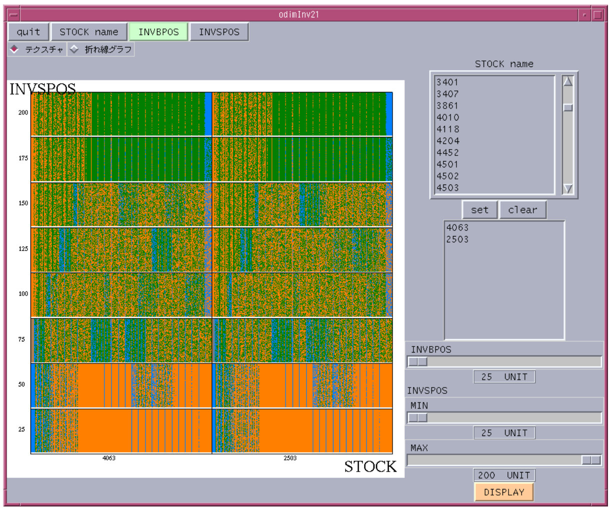

Figure 13. A snapshot of the workspace: the texture-band of selected predictions are simultaneously displayed.

6. Conclusion

We have presented two visualizers for a set of daytime stock price movement predictions. The first visualizer extends line charts with a cluttering detection and control mechanism. The second one extends color-band visualization of price-changes with LoD control. A workspace for browsing the visualization results has also been presented. The user of the system can browse the effect of changing parameters easily with the aid of simultaneous application of either of the visualizers to several sets of predictions. The system has been coded in C++ and Tcl/Tk with OpenGL and Togl.

We have not yet performed feasibility study of the system. However, the system is novel in the sense that other commercial systems still use simultaneous display of windows each of which shows just a few charts. Since even trained traders cannot scrutinize more than ten windows, providing overviews of dozens of price changes is a very important function of the systems supporting traders.

Future work is as follows. First, the cluttering detection algorithm must be improved according to the actual needs of the user. Although it might be possible to revise the algorithm from the view point of computer graphics or cartography, more practical approaches based on the users' needs will be effective, since the target domain is restricted to the financial trading. Second, the texture-band generation and display must be accelerated. It takes about 0.5 seconds to redraw the window shown in Figure 7 on a vanilla PC. When the overview window in Figure 12 is redrawn, this amounts to several seconds which is much larger than 0.1 seconds, the maximum permissible delay of interactive systems. Third, the workspace has only three visualization structures. Making it configurable or adding more structures might be necessary to support the actual needs. Lastly, some feasible study must be performed to evaluate the effectiveness of the system. One plausible way is to compare a subject group using the system with another group using the system with LoD control mechanism off. Another plausible way is to ask the subjects to compare the workspace providing overviews through the dimensional stacking technique with a workspace providing pop-up windows.

Acknowledgment

Part of the programs were implemented by Ayumi Honjo and Maki Harada. The authors sincerely thank them for their contribution to this research.

CMD Research Inc. holds all rights on the agent model and the agent-based simulator described in this paper.

References

[1] W. B. Arthur, J. H. Holland, B. LeBaron, R. Palmer, and P. Tayler. Asset pricing under endogenous expectations in an artificial stock market. In *W. B. Arthur, S. Durlauf, and D. Lane (eds.), The Economy as an Evolving Complex System, II*, pp. 14–44. Addison-Wesley, 1997.

[2] H. Yoon, F. Niwa, and T. Tanahashi. The suggestion of computational dynamics II (intraday simulation in financial market (in Japanese). *Transactions of Japan Society for Computational Engineering and Science*, August 2001.

[3] S. Feiner and C. Beshers. Worlds within worlds: Metaphors for exploring n-dimensional data. In *Proceedings of UIST'90*, pp. 76–83, 1990.

[4] W. Wright. Information animation applications in the capital markets. In *Proceedings of IEEE Information Visualization '95*, pp. 19–25, 136–137, 1995.

[5] Y. Yamashita, Y. Ichikawa, I. Fujishiro, and H. Yoon. A visualization system for day trading of stocks (in Japanese). In *Proceedings of the 13rd Data Engineering Workshop (DEWS 2001)*, March 2001.

[6] D. A. Keim, H.-P. Kriegel, and M. Ankerst. Recursive pattern: A technique for visualizing large amounts of data. In *Proceedings of IEEE Information Visualization '95*, pp. 279–286, 1995.

[7] Y. Saito. A visualization technique for extracting rules from series of multivariate data (in japanese). *IPSJ Transactions on Databases*, Vol. 40, No. SIG6(TOD3), pp. 1–11, August 1999.

[8] D. F. Jerding and J. T. Stasko. The Information Mural: A technique for displaying and navigating large information spaces. In *Proceedings of IEEE Information Visualization '95*, pp. 43–50, October 1995.

[9] Y. H. Fua, M. O. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large databases. In *Proceedings of Visualization 99 (Vis'99)*, pp. 43–50, October 1999.

[10] M. O. Ward, J. LeBlanc, and R. Tipnis. N-land: A graphical tool for exploring N-dimensional data. In *Proceedings of CG Internatinal Conference*, June 1994.