

MLMD: Multi-Layered Visualization for Multi-Dimensional Data

Bowen Yu[†] & Richen Liu & Xiaoru Yuan^{‡1}

¹School of EECS and Key Laboratory of Machine Perception (MOE), Peking University, Beijing, China

Abstract

In this paper, we propose and implement a multi-layered visualization technique in 3D space called Multi-Layered Visualization for Multi-Dimensional Data (MLMD) with its corresponding interaction techniques, for visualizing multi-dimensional data. Layers of point based plots are stacked and connected in a virtual visualization cube for comparison between different dimension settings. Viewed from the side, the plot layers intrinsically form parallel coordinates plots. MLMD integrates point based plots and parallel coordinates compactly so as to present more information at a time to help data investigation. Pertinent user interactions for MLMD method to enable convenient manipulation are designed. By using MLMD and its matching interaction techniques, proper dimension settings and in-depth data perception can be achieved.

Keywords: Multi-Layered Visualization, Scatterplot, Parallel Coordinates, 3D Information Visualization

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and techniques—Interaction Techniques

1. Introduction

The rapidly increasing information and the exploding dimensionality have posed huge challenges on data investigation. Finding the correlations between data attributes is becoming more effort-consuming [Don00]. There exist mathematical methods for feature selection and extraction including the classic PCA [Jol02], which generate dimension representatives in order to reduce the original data to computable size. However, without human guidance, mathematical solutions often fall into data pitfalls or fail concerning processing time limit. On the other hand, manually selecting dimensions of data is not only laborious but also hard to succeed.

Within the category of 2D data visualization, scatterplot and its extended version for high-dimensional data, multi-dimensional scaling plot, are both classic point based methods and data processing. To make full use of the advantages of the two metaphors, we extend the presentation style of 2D point based plots into 3D visualization space by stack-

ing 2D point based plots. We name the proposed visualization method MLMD (Multi-Layered visualization for Multi-Dimensional data) according to its principle and purpose.

Due to the 3D layout, scatterplots and MDS plots generated from different dimension settings can be viewed together so as to determine the influence of dimension groups or dimension weights on the plot results. 3D layout is more intuitive in our design, looked from the side, the axes of the stacked scatterplots and MDS plots intrinsically present parallel coordinates plots, if points representations of the same data item are connected by polylines in 3D space. The problems of 3D visualization call for pertinent interaction methods for the visualization purposes. To fully take the advantage of the multiple-view properties and counter the 3D information visualization problems, we requires many interactions on a multi-touch platform which can provide us various touch gestures and intuitive feelings. The proposed multi-layered visualization method is a novel approach of information visualization targeting at multi-dimensional data. It demonstrates the possibility of using interactions to counter typical obstacles of 3D visualization.

[†] Now in New York University

[‡] Email: xiaoru.yuan@pku.edu.cn

2. Related Works

Previous works have laid emphasis on multi-dimensional data visualization. The possibilities of extending 2D information visualization into 3D have been explored.

The point based plots stacked together determine the fundamental structure of the proposed visualization, including scatterplots and MDS plots. Scatterplots are widely applied in many fields serving research and education purposes. Multi-dimensional Scaling (MDS) [BG05] algorithms can be used to minimize the difference between the dissimilarity matrix of the plot in 2D space and the origin dissimilarity matrix. However, both methods have drawbacks. The scatterplot is restricted to 2 dimensions, requiring a large number of plots to cover all the dimensions. The MDS plot may result in an unavoidable information loss and the dimension amelioration process is laborious. Parallel coordinates plot [ID90] expands the dimension domain that a general display holds and is already a widely accepted technique for visualizing high-dimensional data. The SPPC [YGX*09] method scatters points into the parallel coordinates and use curves to connect the scatterplots and therefore, generate smooth perceptual results between the two metaphors. In our work, MLMD naturally incorporates parallel coordinates views based on stacked scatterplots and MDS plots.

Occlusions in 3D visualizations are likely to occur, effective interaction methods that can resolve the incoherence between physical operations and most existing approaches are immature. Vislink [CC07] shows how the user can display multiple 2D visualizations, reposition and reorganize them in 3D, and display relations between them by edges propagation. MLMD is an extended version of Vislink considering the plane number, but it more specifically applies scatterplots and MDS plots onto the layers rather than graphs and networks. Cubic visualization space has once been proposed by Elmqvist et al. [EDF08] to maintain the visual coherency during the dimension switches of scatterplots.

3. Visualization Design

In this section, we introduce the proposed stacked MDS method and its cubic visualization space.

3.1. Design Logics

Figure 1 gives an overview of the stacked-layer structure of the MLMD method. Plot layers are initially placed along the z-axis that goes into the screen. For descriptive clarity, we define the cube face originally displayed on the screen as the front view and the others as left, right, top, bottom and back view respectively. Notice that this setting can be altered when needed by simply changing the stacking direction.

Each plot layer shows a scatterplot or an MDS plot. Every point based plot contains one point representation for each data item. The points on different layers representing the same data item are connected by polylines that go through 3D space. Viewed from the front or back, the stacked plots go into the screen (Figure 1(a)). Viewed from the other

sides, layers stay perpendicular to the screen (Figure 1(b)). The orthogonal projections on the front and back faces present precise calibrated MDS plot view (Figure 1(c)), while the projections of all the segments of the polylines present roughly standard parallel coordinates on the other 4 faces of the cube with some specialties as the MDS axes could not present exact data values (Figure 1(d)). As a result, the designed metaphor presents stacked plots views on 2 of the cube faces and 4 parallel coordinates views on the others, dividing the views into two categories. The front view belongs to the stacked plots view category at the start-up phase of the system.

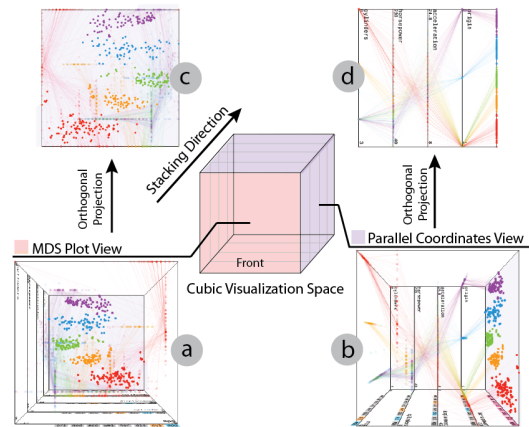


Figure 1: Design Logics.

To overcome the common 3D clutters and occlusions (Figure 2(a)), we propose an interpolation method that inserts invisible layers between MDS plots so as to quickly reveal the intermediate state between the original adjacent layers (Figure 2(b)). The dimension settings of the inserted layers are determined using linear interpolation. The MDS plots apply force directed layouts [Cha96] so as to fully utilize the positions achieved during the last configuration as the user may modify the dimensions from time to time. In our design, we focus on utilizing the user's interaction so as to save the computational resources of the system and therefore, we enable rotation of the layers which helps remove the clutters (Figure 2(c)). To summarize, all the properties of the interpolated layers are determined by the following equation, where t is the scaled parameter of the interval between the layers (from layer1 to layer2):

$$Prop(t) = t * Prop_{layer2} + (1 - t) * Prop_{layer1} \quad (1)$$

3.2. View Alternation

Alternating the viewing perspective of the visualization enables the user to quickly switch between the two view categories. Occlusions bring interference in item selection in

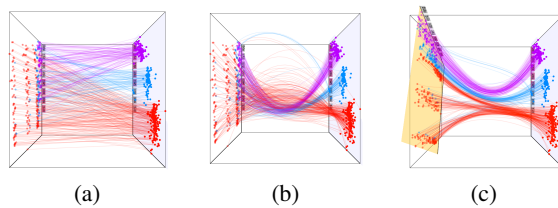


Figure 2: (a) Clutters encountered in 3D parallel coordinates. (b) Inserted interpolated invisible layers are placed between layers to form control points of the curves. (c) Rotation assists to better show the diversities of MDS results and avoid occlusion.

3D visualization. To help the user manipulate objects on a specified layer, we design a FocusBoard for layer manipulation as shown in Figure 3(b) (enclosed by blue edges). The rendered plot on the FocusBoard will always be highlighted. With the FocusBoard, layer selection becomes straightforward. We reorganize the depths of plot layers inside the cube after axes relocation on the parallel coordinates view, and hide the layers that are outside the visualization focus area. Consequently, modifying the axes positions can be viewed as semantic zoom. This Focus+context technique is typically useful when the data contains numerous dimensions. By applying axes zooming or moving, the user can filter out uninterested plots and focus on informative MDS plot and distinctive dimension groups. In our system, brush operations are naturally linked on all views since they refer to the same data items, making it easy to adjust visual effects from any view of the system.

4. Interaction Methods

The interactions of the visualization includes basic navigation methods for changing viewing perspectives, brushing and layer editing. We apply circular menus to make smooth transitions between different types of interactions and reduce the user's burden of memorizing complicated gestures.

4.1. Changing Viewing Perspectives

Viewing the visualization from different perspectives is critical to fully understanding the data. We make the 3D visualization space accessible to the user by providing navigation methods including trackball, orthogonal projection, orthogonal rotation and free navigation. When the cube stops at a state that none of its face normals are perpendicular to the screen, the operations that follow can probably result in perceptual inconsistency. To counter this problem, we provide orthogonal rotations. The viewing perspective will be automatically fixed perpendicular to the face that has a normal nearest with the z axis of the screen, which is called a Dock action. In free navigation mode, the visualization cube does not dock after its orientation is modified. The user switches on or off free navigation mode by pressing the corresponding

button on the interface. Free navigation provides opportunities for stopping at intermediate viewing perspectives, which is quite useful for dealing with clutters.

4.2. Brushing

We provide lasso brush on the FocusBoard for coloring the scatterplots or MDS plots (Figure 3(a)). The FocusBoard position can be set by making vertical pan gestures on the parallel coordinates view (Figure 3(b)) or dragging the FocusBoard handlers (Figure 1(a)). For parallel coordinates, cutting brush gives instant feedback after the user's touch track intersects a parallel coordinates segment (Figure 3(c)). As the opacities are important in the cube case and improper opacity settings could result in severe occlusions and clutters, alpha brushing is introduced to enable adjustment on the transparency of plots and parallel coordinates. The user can modify the alpha of all the layers by drawing one curve, which reduces the time cost of alpha adjustment (Figure 3(d)).

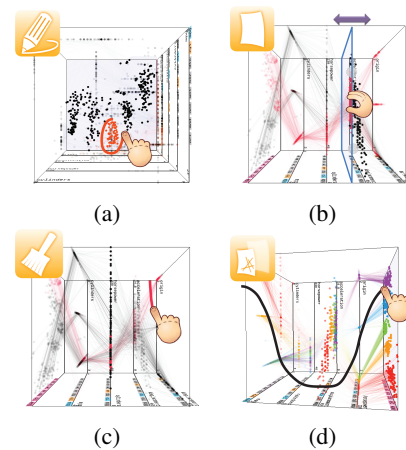


Figure 3: Three types of brushes and Focusboard

4.3. Layer Editing

Changing dimensions is of utmost importance for interactively optimizing the dimension groups, clusters, and multi-dimensional scaling results. We define a set of operations including Add, Remove, Move and Switch operations for this purpose. Taking the advantages of these operations, the user can build MDS plots or scatterplots of any dimension groups on a plot layer. The user may want to copy the layer settings for slight modification or remove one layer, resulting in a Copy and a Remove layer editing operation. To interpolate the interval between two layers, the user can perform Interpolate operation to replace the original parallel coordinates segments by curves, the control points of which are calculated by Equation 1. Rotating an MDS layer reconfigures its cluster positions and can reveal more information combined with parallel coordinates. What's more, parallel

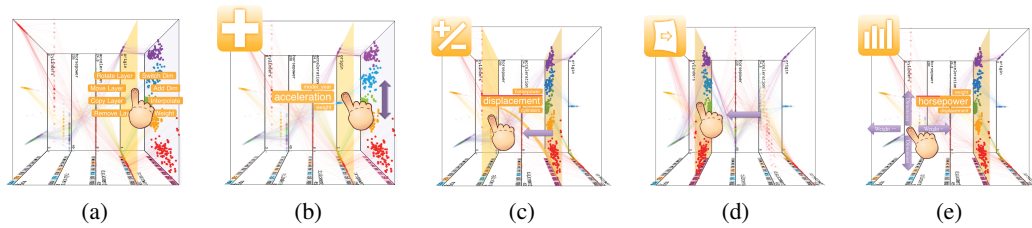


Figure 4: (a) Circular menu popped up. (b) Dimension selection (for Add operations). (c) Move Dimension operation. (d) Move Layer operation. (e) Weight adjustment.

coordinates segments may cluster better at a specific rotation angle. All the possible layer editing operations will be shown in a pie menu (Figure 4(a)) after the user touch one layer to enter layer editing mode. The user then moves finger to any of the near menu items to perform desired operations. With the help of flow menu mechanism [GW00], the user is able to perform a complete round of layer editing within one touch. The corresponding mapping from touch gestures to layer editing operations are as follows:

- Begin: In parallel coordinates view, touch a layer to pop up a circular menu. (Figure 4(a)).
- Select: Select desired operations in the menu.
- Interpolate/Copy/Remove/Interpolate Layer: These operations receive instant feedback.
- Add/Remove Dim: Vertically move the finger to select the dimension to be added/removed (Figure 4(b)).
- Move Dim: Horizontally move the finger to select the targeted layer. (Figure 4(c)).
- Switch Dim(Scatterplot Only): Move the finger to the axis tag to decide which axis to be replaced with. Then vertically select a dimension.
- Move Layer: Horizontally move the finger to put the layer to the desired position. (Figure 4(d)).
- Rotate Layer: Vertically move the finger to achieve the desired angle of rotation. (Figure 2(c)).
- Edit Weight: Horizontally move the finger to adjust dimension weight for selected dimension. (Figure 4(e)).

4.4. Semantic Zoom

Due to the limitation of the size of display, visualization clutter has to be resolved. Traditional zooming would result in changes of viewing perspectives and loss of item tracking. We integrate two types of semantic zoom in our system to help the user filter out irrelevant data and focus on interested items, the irrelevant data will be stacked closely by one pinch gesture. The first type is semantic layer zoom. Through pinching and panning, the user can not only adjust the intervals between axes of parallel coordinates, but also hide other layers in order to fully utilize the cube space. After the user performs a pinch gesture, the central three layers will be relocated to the cube range (Figure 5(a)), which provides a chance for downward investigation. The second type is semantic data item zoom. After the user brushes some data items, the pinch gestures that follow will be recognized as semantic zoom operations. Zoom In hides the unbrushed items and Zoom Out redisplay all the data (Figure 5(b)).

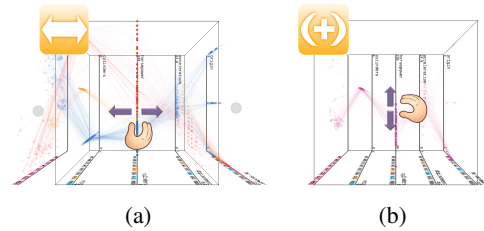


Figure 5: Two types of semantic zoom.

5. Discussion

We ask 12 volunteer students in computer science to test the proposed and implemented tool and we collect their feedbacks. The trials are carried out both on standard PCs and on multi-touch devices. The dataset of our application case is the car data contains 406 data items and 9 attributes including mpg, horsepower, weight, acceleration etc. At the first stage, there is one MDS containing all the dimensions set at the back of the cube. It is easy to notice that on the MDS there form five distinct clusters (Figure 1(c)). More complicated features selection including features generated from linear combination of dimensions could be realized using weight editing function provided in MLMD by field researchers. The volunteer users especially prefer the user-friendly interface of the system and mention that the incorporated interactions work well for the proposed MLMD method.

Based on the users' experiences, it can be concluded that the dimension number should not be greater than 20 and the cube could hold up to 8 layers while still presenting perceivable information. But these limitations are not without solutions. In our system, we have designed semantic zoom for layers so as to gain more space for interested layers.

6. Conclusion

The visualization approach stacks plot layers and naturally present multiple views including scatterplots and parallel coordinates. It can help the user better perceive the properties of various dimensions of multi-dimensional data.

Acknowledgements

This work is supported by NSFC Project (61170204) and Key Project (61232012).

References

- [BG05] BORG I., GROENEN P. J. F.: *Modern Multidimensional Scaling: Theory and Applications*, 2nd ed. Springer, Aug. 2005. 2
- [CC07] COLLINS C., CARPENDALE M. S. T.: Vislink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1192–1199. 2
- [Cha96] CHALMERS M.: A linear iteration time layout algorithm for visualising high-dimensional data. In *Proc. IEEE Vis '96* (1996), pp. 127–131. 2
- [Don00] DONOHO D. L.: *High-dimensional data analysis: The curses and blessings of dimensionality*. Aug. 2000. 1
- [EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1539–1148. 2
- [GW00] GUIMBRETIERE F., WINOGRAD T.: Flowmenu: combining command, text, and data entry. In *Proc. ACM UIST '00* (New York, NY, USA, 2000), UIST '00, ACM, pp. 213–216. 4
- [ID90] INSELBERG A., DIMSDALE B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proc. IEEE Vis '90* (1990), pp. 361–378. 2
- [Jol02] JOLLIFFE I. T.: *Principal Component Analysis*, second ed. Springer, Oct. 2002. 1
- [YGX*09] YUAN X., GUO P., XIAO H., ZHOU H., QU H.: Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1001–1008. 2