MLMD: Multi-Layered Visualization for Multi-Dimensional Data

Category: Research

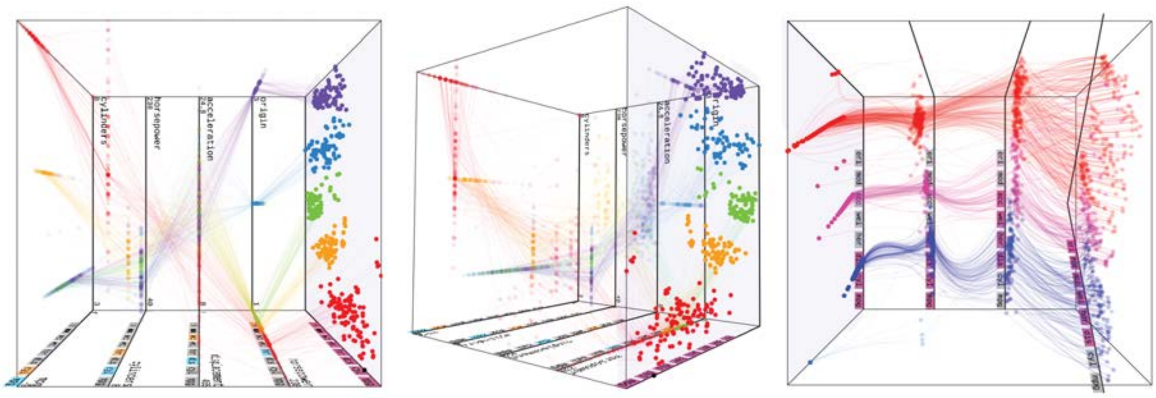


Figure 1: The proposed MLMD method stacks layers of MDS plot and defines a cubic visualization in 3D space. It is well equipped with pertinent interactions that are carefully designed for the user to conveniently achieve layer settings and carry out data exploration tasks throught the visualization including dimension filtering, feature selection, correlation revelation, etc. We implement MLMD on a multi-touch platform to fully take the advantages of gesture operations.

ABSTRACT

Visualization and data mining techniques have for long been laying emphasis on high-dimensional data processing. In this paper, we propose a multi-layered visualization technique in 3D space called 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, which are typically effective in visualizing high-dimensional data. MLMD integrates point based plots and parallel coordinates compactly so as to present more information at a time to help data investigation. The user gradually find the desired dimension set of multi-dimensional data by iteratively editing layer dimensions. We carefully design pertinent user interactions for MLMD method to enable convenient manipulation of the layer properties and views. By using MLMD and its mating interaction techniques, proper dimension settings and in-depth data perception can be achieved, presenting a novel way of perceiving multi-dimensional data in 3D visualization space that coordinates multiple views.

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

Index Terms: I.3.6 [Computer Graphics]: Methodology and techniques—Interaction Techniques; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces (GUI)

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 with the the curse of dimensionality [9]. There exist mature mathematical methods for feature selection and extraction including the classic PCA [22], which generate dimension representatives in order to reduce the original data to computable size. However, without human guidance, mathematical solutions and algorithms for data investigation often fall into data pitfalls or fail concerning processing time limit. On the other hand, mannually selecting dimensions and features of data is not only laborious but also hard to succeed. Designing user-friendly systems that aids human data investigation or incorporates human judgement in algorithm procedure can largely speed up high-dimensional data investigation. Therefore it is recommended, if possible, to integrate and design user interface to handle multi-dimensional data in order to effectively filter out insignificant and undesired data attributes.

Although there exist plenty of methods that visualize multi-dimensional data and present informative results, most of the information visualization methods work only on 2D surfaces including reading materials and monitor displays. Within the category of 2-dimensional data visualzation, scatterplot and its extended version for high-dimensional data, multi-dimensional scaling plot, are both classic point based methods that are widely applied for presentation and data processing. To effectively visualize multi-dimensional data and 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 stacking 2D point based plots. The planes that point based plots, whether scatterplots or MDS plots, lie on are so called plot layers, the stacking of which defines a cubic visualization space, namely, the visualization cube. We name the proposed visualization method MLMD (Multi-Layered visualization for Multi-Dimensional data) according to its principle and purpose.

By the MLMD design, 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, therefore providing a novel way of comparing data attributes and carrying out feature selection or dimension reduction tasks. The design of the multi-layered visualization intrinsically presents parallel coordinates plots on its sides. Looked from the side, the axes of the stacked scatterplots and MDS plots intrinsically present parallel coordinates plots, providing that points representations of the same data item are connected by polylines in 3D space. We lay strong emphasis on the design of the collaborative working mechanism of MLMD in order to yield useful results based on the integrated three metaphors, while preserving the natural power of the three types of visualizations. Consequently, the interaction techniques and user interface are considered in detail so as to coordinate the different types of metaphors for multi-dimensional data investigation purpose. MLMD is implemented on a multi-touch platform in our work to support convenient interactions. The proposed multi-layered visualization method is a novel approach of information visualization targeting at multi-dimensional data that naturally integrates multiple existing metaphors. Besides, it demonstrates the possiblity of using interactions to counter typical obstacles of 3D visualization and reduce gaps between views so as to yield satisfactory visualization results.

The remainder of this paper is organized as follows. Related works are discussed in Section 2. Details of the multi-layered visualization design are given in Section 3, followed by the introduction to the interaction techniques incorporated in the system Section 4. Application cases of the metaphor are demonstrated in Section 5. A discussion on user feedback, method scalability and future work is presented in Section 6. Finally, we draw a conclusion in Section 7.

2 Related Works

Previous works have laid emphasis on multi-dimensional data visualization. The possibities of extending 2D information visualization into 3D have been explored. Interaction techniques also play an important role in the proposed system.

2.1 Scatterplots and MDS Plots for Multi-Dimensional Data Visualization

The point based plots stacked together determine the foundamental structure of the proposed visualization, including scatterplots and MDS plots. As part of the basic vocabulary of 2-dimensional data visualization, scatterplots are widely applied in many fields serving presentation, research and education purposes and normally a set of scatterplots can visualize multi-dimensional data. Distances between points in scatterplots are easy to perceive and exact dimension values can be read. Multi-dimensional Scaling (MDS) [4] algorithms aim at minimizing the difference between the dissimilarity matrix of the plot in 2D space with the origin dissimilarity matrix. Point distances in MDS plots reflect abstract distances in high dimensional space and are useful for differentiate data items. However, both methods have drawbacks in dealing with multi-dimensional data. The presentation capability of scatterplot is restricted to 2 dimensions, requiring a large number of plots to cover the all the dimensions. The MDS plot may result in an unavoidable information loss and the dimension amelioration process is laborious and time-consuming.

To solve these problems, scatterplots and MDS plots are usually combined with other visualization metaphors to enhance system capability, resulting in multiple-view visualizations. Many works have focused on visualizing multi-dimensional data and can be divided into five categories including standard 2D/3D displays, geometrically transformed displays, icon-based displays, dense pixel displays and stacked displays [23] [6]. Methods that integrate multiple views have been proved effective in visualizing multi-dimensional data. Baldonado et al. [1] give a set of guidelines on when and how multiple-view system should be used. To overcome the limitations of 2-dimensional scatteprlots [7] , applying multi-dimensional scaling for high-dimensional data is typically effective in many cases. The study of scaling metrics has been a research hot-spot since early time [30]. Parallel coordinates plot [20] expands the dimension domain that a general display holds and is already a widely accetped technique for visualizing high-dimensional data. The combination and integration of point-based plots and parallel coordinates plot is of great concern in recent studies and has been exploited by previous works. The SPPC [32] method scatters points into the parallel coordinates and use curves to connect the scatterplots and therefore, generate smooth perceptional results between the two metaphors (Figure 2(a)). Holten's method [18] embeds scatterplots between each pair of parallel coordinates axes and rotates the scatterplot axes by 45 degrees (Figure 2(b)). Viau et al. [27] propose a hybrid method that integrates scatterplot matrix and parallel coordinates to better visualize a network (Figure 2(c)), in which rotation is applied for switching between different states, allowing the user to view the matrix in a distinctive 3D perspective. In our work, MLMD naturally incorporates parallel coordinates views based on stacked scatterplots and MDS plots (Figure 2(d)). The idea of curved connection between layers is also introduced.

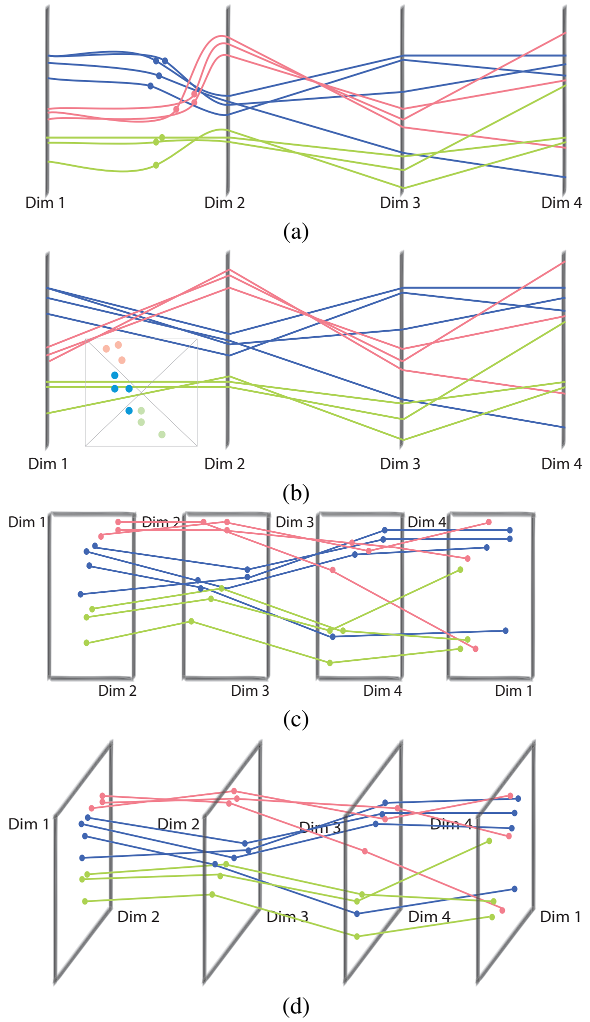


Figure 2: The comparisons between different integrations of scatterplot, MDS plot and parallel coordinates. (a) SPPC apply interpolated curves to connect scatterplots and MDS plots with parallel coordinates; (b) Holten's method embeds rotated scatterplots between each pair of parallel coordinates; (c) The Parallel Scatterplot Matrix maps parallel coordinates on scatterplot matrix; (d) In our design, the scatterplot layers are set parallel with each other, in which case depth information are utilized and stacked plot view is achieved.

2.2 3D Information Visualization

In MLMD, the plot layers are stacked along the z-axis of the screen, defining a cubic volume in 3D visualization space. However, 3D information visualizations bring difficulties. Occlusions are likely to occur and effective interaction methods that can resolve the incoherence between physical operations and their visual results are still under development and most existing approaches are immature. The limitation of human spatial perception is another problem, based on which Tim Dwyer [10] gives an approach to 3D information visualization using a 2.5D design attitude. The study indicates that 3D visualizations can help analyzing complicated relations in multi-dimensional data but 2D is favored for observing relationships between a certain pair of attributes. Concerning both possibilities and obstables, research on 3D information visualization is of high interests. Wegenkittl et al. [28] use extrusion to visualize trajectories of higher dimensional dynamical systems. Johansson [21] shows that insights on the relations between a single dimension and other dimensions can be achieved by rearranging the other axes around the target axes. Fanea et al. [13] present an interactive 3D integration of parallel coordinates with star glyphs to provide quick impressions on data distribution and make visualization easier to read. Vislink [8] 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. [12] to maintain the visual coherency during the dimension switches of scatterplots.

2.3 Interaction Techniques

The problems of 3D visualization call for pertinent interaction methods designed for the visualization purposes. To fully take the advantage of the multiple-view properties and counter the 3D information visualization problems, we carefully design the multi-touch interactions for the system. The possibility of presenting a scatterplot on a multi-touch device is mentioned by Heilig et al. [17] considering multi-user collaboration. Besides basic cutting and lasso brushing for selecting data items on parallel coordinates, a number of ways have been developed for convenient manipulation. For example, Hauser et al. [16] propose angular brushing as an effective way to brush and highlight data items. Kosara [24] demonstrates the mechanism of 1- to 4-finger operations on parallel coordinates on a multi-touch deivce. These methods can be extended to 3D information visualization with modifications. Viewing 3D visualization involves 3D navigation. Yu et al. [31] show how 3D navigation should be performed on 2D multi-touch devices, though with a major concern of scientific visualizaiton. The DabR, a multi-touch system designed by Edelmann et al. [11] for intuitive 3D navigation, is a typical example of how multi-touch device is applied for 3D interations. As the proposed MLMD includes multiple scatterplot and MDS plots, as well as parallel coordinates, to be handled and also requires 3D navigation, a complicated set of operations are defined. To smoothly connect different stages of interactions and lower the requirements of gesture memorizing, circular menus inheriting the ideas of pie menu [19] and its variants [25, 15] are incorperated into our system.

3 Visualization Design

In this section, we introduce the proposed stacked MDS method and its cubic visualization space. The properties of the stacked plots are discussed and the benefits of changing viewing perspectives are stated. We focus on the functionality of the metaphor in this section and the details of interactions that manipulate the visualization are introduced in the following section 4.

3.1 Design Logics

Figure 3 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, framing a visualization cube. 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 according to their relative positions with the front view. 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, depending on the number of its included dimensions. 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 trough 3D space. The intersection points of the 3D parallel coordinates polylines with the plot layers are determined by data values of certain data attributes. When an axis of parallel coordinates is part of an MDS plot, the intersection point presents no exact data value but only an abstract location that reflects the result of multi-dimensional scaling. Viewed from the front or back, the stacked plots go into the screen (Figure 3(a)). Viewed from the other sides, layers stay perpendicualr to the screen (Figure 3(b)). The projections of all the point representations on the front and back faces present precise calibrated MDS plot view (Figure 3(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 3(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. Since the same type of views functions similarly, we simply distinguish the cube faces by stacked plots views and parallel coordinates views in the following sections. Along the axes of parallel coordinates, or the borders of the stacked plots, we render colored tag to provide quick overview of the dimension settings of the layers. Pink-color tags indicate MDS plot layers. A cyan tag represents the dimension on a scatterplot's horizontal axis, while an orange tag represents the vertical dimension.

As the major intent of the system is to provide the user with a convenient method to compare the results generated from different settings of multi-dimensional scaling dimensions, we fully exploit the advantage of stacked layers, where the connections of the point representatives indicate potential correlation of the data attributes. To overcome the common 3D clutters and occlusions appeared where lines are dense (Figure 4(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 4(b)). The dimension settings of the inserted layers are determined using linear interpolation. The MDS plots of the cube apply force directed layouts[5] so as to fully utilize the positions achieved during the last configuration as the user may modify the dimensions from time to time. Since multiple force-directed MDS plots are likely to generate results with rotation offset considering the uncontrolled rotation, methods including the ICP is proposed [2]. 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 manual rotation of the layers. Rotation helps remove the clutters (Figure 4(b)). As the plots are rotated, the inserted intermediate states are recaculated to give real time respond and help the user determine whether the current view presents meaningful information. To prevent the system from generating tangled plots, when interpolating the invisble middle layers, we initialize the point positions using linearly interpolated values of the point coordinates on nearby layers at the beginning of the MDS algorithm. This techinque ensures that an automatic layout is achieved without too much misleading biased MDS rotation. To summarize, all the properties of the interpolated layers are determined by the following equation, where t is the scaled parameter

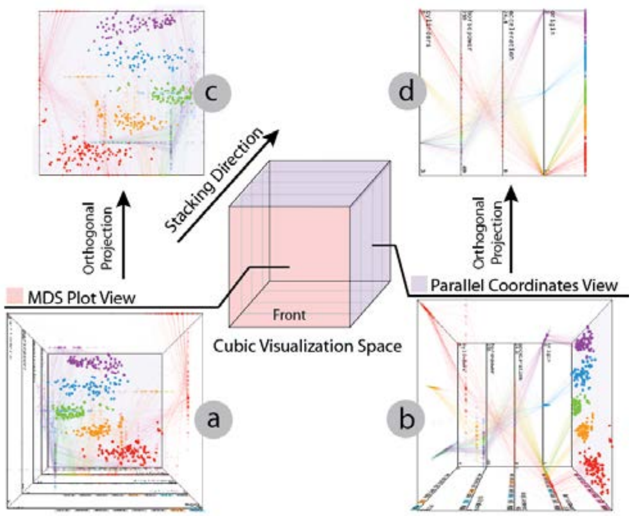
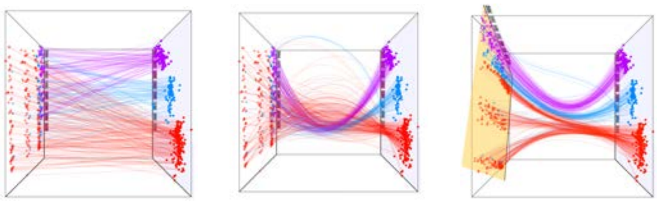


Figure 3: Design Logics: (a) Stacked plots views: When viewed from the front or the back face, MDS plots and scatterplots are stacked along the z-axis of the screen. Two handlers of the FocusBoard enable FocusBoard dragging in MDS views (enclosed by blue edges). (b) Parallel coordinates views: Parallel coorindates composed of 3D polylines serve as interaction ports for layer selection and editing. The half-transparent light blue layer denotes the FocusBoard position (enclosed by blue edges). (c) Orthogonal projection of stacked plots calibrate the high-dimensional projection results and sets up a better view for comparisons; (d) Orthogonal projection of parallel coordinates shows the visualization results in a traditional parallel coordinates metaphor.

of the interval between the layers to be interpolated (from layer1 to layer2):

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



(a) (b) (c)

Figure 4: (a) Clutters encountered in 3D parallel cooridinates. (b) Inserted invisible layers with linearly interpolated dimension settings are placed between layers to form control points of the curved connection, which help the user peceive the plot outpus under intermediate settings. (c) Rotation assits the curved parallel coordinates to better show the diversities of MDS results with various dimension selections and weight values. Curved connection avoids parallel coordinates occlusion.

3.2 View Alternation

The stacked plots views show how the distribution of point representations vary according to different dimension settings, while the parallel coordinates views maintain the traditional advantages of parallel coordinates including high-dimensional information visualization and cluster and correlation identification. Alternating the viewing perspective of the visualization enables the user to quickly switch between the two view categories, and therefore provides extra functionality from the two combined metaphors.

Occlusions bring interference in item selection in 3D visualization. Although the user can adjust the opacity of different layers to avoid visual occlusions and clutters, determing which layer the user wants to modify is no trivial task except when an extra list for selection is provided, which would otherwise undesirably increase the system redundancy. The parallel coordinates views of the cube, in addition to their normal usage, support convenient plot layer manipulation in 3D as the borders of stacked layers are exactly the axes of parallel coordinates. Viewed from parallel coordinates views, the plot layers are roughly perpendicular to the screen and are easily distinguished. To help the user manipulate objects on a specified layer, we design a FocusBoard for layer manipulation as shown in Figure 3(a) and (b) (enclosed by blue edges). The rendered plot on the FocusBoard will always be highlighted. With the FocusBoard, layer selection becomes straightforward. The user first move the FocusBoard to the desired position from the parallel coordinates views and then carry out corresponding layer editing operations. After satisfactory results are achieved, the FocusBoard can be switched off to give an overview of the visualization. Besides, when moving the FocusBoard through the stacked plots, the user can trace the change of positions of data items on consecutive plot layers (Figure 3(b)), which enables the user to acquire knowledge on the data distribution. What's more, zooming or moving the parallel coordinates axes changes the depths of plot layers. More specifically, 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 from the angle of data investigation. This focus+context technique is typically useful when the data contains numerous dimensions, which could lead to dense layers of visualization. By applying axes zooming or moving, the user can filter out uninterested plots and focus on informative MDS plot and distinctive dimension groups. In addition, similar to inverting axes of traditional parallel coordinates, under different rotation configurations of layers, interesting results such as correlation of dimensions can be revealed. 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, including colors and alpha values.

4 Interaction Methods

The interactions of the visualization includes basic navigation methods for changing viewing perspecitves, brushing and layer editing. To carry out the desired modification, several interaction steps have to be taken in a specific order, i.e. firstly Select a layer and then Add a dimension to it. We apply circular menus to make smooth transitions between different types of interactions as well as reduce the user's burden of memorizing complicated gestures. Basically, interactions are in a touch-move-release flow. Touch initiates an interaction round as a pop-up circular menu is shown. Release ends the interactions and put the selected operation into effect. Release at any unrecognized point outside the menu cancels the current action. At any time, the current operation will be shown at the top-left corner by a operation hint to help the user understand what is going on. Details of interactions are given by type in the following subsections.

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 convenient navigation methods including trackball, orthogonal projection, orthogonal rotation and free navigation. Trackball is a classic method suitable for manipulating rendered 3D object. The user can use trackball to rotate the visualization cube in the virtual space, which is carried out in a dragging manner, in order to switch from one view to antoher. When the user rotates the visualization cube, the transitional state between two views plays a vital role in maintaining the perceptional consistency as the user can track interested data items to explore their locations in two different views. However, 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 perceptional 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. Dock offers shortcuts to default viewing perspectives as manually rotating the screen heading to perpendicular position is hard to achieve. Besides, to preserve the properties of traditional parallel coordinates as well as calibrate the comparison results, orthogonal projections of both MDS views and parallel coordinates views can be enabled when the view is docked, as shown in Figure 3(c)and(d). Orthogonal projection yields a more precise presentation of data values. In free navigation mode, the visualization cube does not dock after its orientation is modified. Zoom and translation are available in free navgation, making more part of the 3D space accessible to the user. A pinch gesture triggers a zoom and a pan gesture triggers a translation in free navigation mode. 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 perspecitves between two types of views, which preserves critical perceptional coherency, and is quite useful for dealing with clutters and occlusions.

.

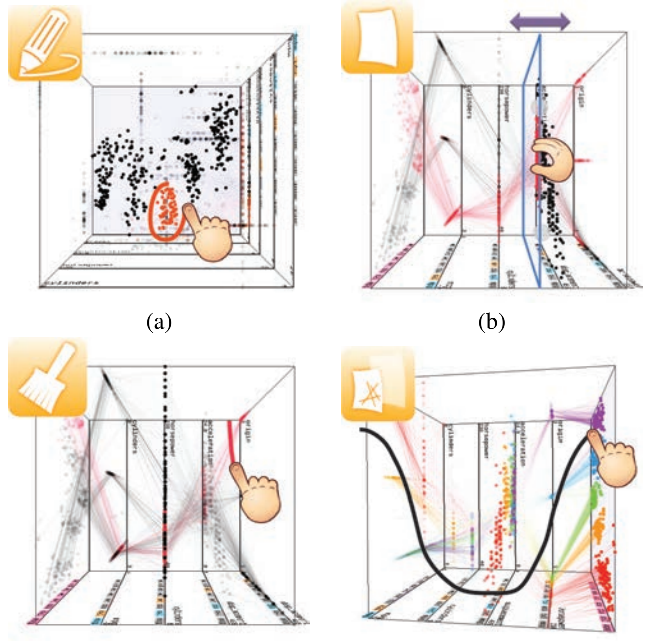
4.2 Brushing

Brushing and selecting data items from a large dataset help further understand valuable subsets of the data. Brush operation is typically more intuitive on a multi-touch device than on a monitor. To fully take the advantage of the convenience of touches for brushing purposes, we define several types of brushes including lasso brush, cutting brush and alpha brush for different adjustments. To enter the brushing mode, the user expand the toolbox at the top-left corner of the screen.

We provide lasso brush on the FocusBoard for coloring the scatterplots or MDS plots (Figure 5(a)). The FocusBoard position can be set by making vertical pan gestures on the parallel coordinates view (Figure 5(b)) or dragging the FocusBoard handlers (Figure 3(a)). For parallel coordinates, cutting brush gives instant feedback after the user's touch track intersects a parallel coordinates segment (Figure 5(c)). As the opacities of rendered items 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 the point based plots and parallel coordinates. Performing an alpha brush is similar to doing a cutting brush. Instead of coloring data items, alpha brush changes the alpha value of intersected layer based on the intersection position. The user can modify the alpha values of all the layers by drawing one curve, which reduces the time cost of alpha adjustment (Figure 5(d)).

4.3 Layer Editing

Changing dimensions is of utmost importance for interactively optimizing the dimension groups, clusters, and multi-dimensional scaling results. The dimensions of layers can be modified at any time during the investigation process. We define a set of operations including Add, Remove, Move and Switch operations for this purpose. Adding dimension and removing dimensions adjust the dimension group of an MDS plot. If carried out on a scatterplot, an Add operation generates a resultant MDS plot with 3 dimensions.



(a) (b) (c) (d)

Figure 5: (a) Lasso Brush: The user draws a lasso on the FocusBoard to color the scatterplots or the MDS plots. (b) FocusBoard: The user uses two-finger panning to move the FocusBoard on multi-touch devices. Two-hand collaboration is easy to carry out in on our design. (c) Cutting Brush: The brush colors data items when it intersects parallel coordinates segments; (d) Alpha Brush: The user draws a curve to simultaneously set alpha values for different parallel coordinates segments (also for the plots). Alpha value of the endpoints of parallel coordinates on the middle layer is lower than the others;

Conversely, if a Remove operation is done on a 3-dimensional MDS, it is switched back to a scatterplot. Switching dimension allows the user to change the horizontal or vertical dimension of a scatterplot. Moving dimension allows the user to move one dimension from an MDS to another scatterplot or MDS, reducing the cost of an equivalent remove-add operation pair to better suit the user's dimension adjustment requirements. 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 from the visualization, 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 coordinages segments by curves, the control points of which are calculated from the settings of the two layers by Equation 1. Besides, due to the 3D properties of MLMD, rotating a layer can generate potential results that are helpful to interpreting parallel coordinates and point based plots. As an MDS plot does not reflect actual values of attribute, rotating an MDS layer reconfigures its cluster positions and can reveal more information combined with parallel coordinates. What's more, parallel coordinates segments may cluster better at a specific rotation angle. All the possible layer editing operations will be shown in a pie menu (Figure 6(a)) after the user touch one layer in the parallel coordinates view to enter layer editing mode. The user then moves her finger to any of the neary menu items to perform desired operations. With the help of flow menu mechanism [15], 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:

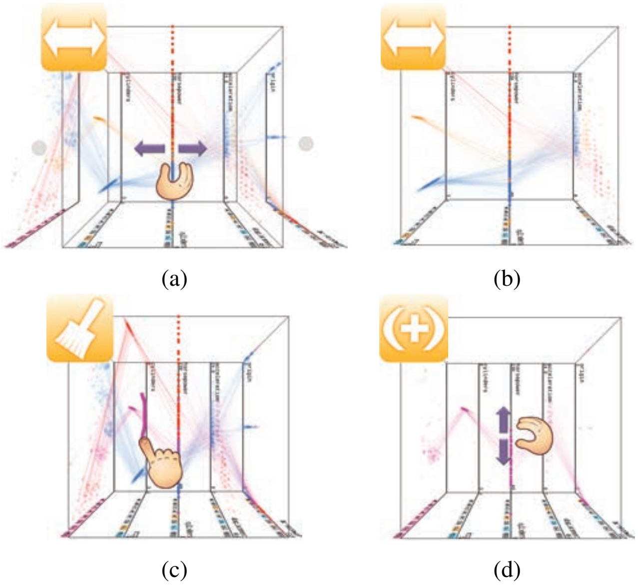
- Begin: In parallel coordinates view, touch a layer to enter layer editing mode, and a circular menu is popped up. (Figure 6(a))
- Select: Move the finger to select desired operations in the menu. Do any of the following based on the selection.
- Interpolate/Copy/Remove/Interpolate Layer: These operations receive instant feedback.
- Add/Remove Dim: Vertically move the finger to select the dimension (Figure 6(b)) to be added/removed. Release.
- Move Dim: Vertically move the finger to select the dimension to be moved. Then horizontally move the finger to select the targeted layer. Release. (Figure 6(c))
- Switch Dim(Scatterplot Only): Move the finger to the axis tag to decide which axis to be replaced with. Then vertically move the finger to select a dimension.
- Move Layer: Horizontally move the finger put the layer to the desired position. (Figure 6(d))
- Rotate Layer: Vertically move the finger to achieve the disired angle of roation. (Figure 4(c))
- Edit Weight: Vertically move the finger to select a dimension. Then horizontally move the finger to adjust dimension weight. (Figure 6(e))

The dimension tags will be highlighted when a corresponding dimension is selected, and labels showing the names of the selected dimension and its neighbor dimensions will pop up to better inform the user of her current operation (Figure 6(d)). A red slot on the label (Figure 6(e)) will show the current weight value of the selected dimension on MDS. The tags of MDS plots also provide weight settings overview by proportionally coloring the tag areas (Figure 6(e)). When moving and rotating a layer, or moving and switching a dimension, instructive hints including transparent temporary results and arrows will be shown to help the user achieve her purpose (Figure 6(b)(c)).

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 trackings. Focus+context methods such as Magiclens [3], TableLens [26], EdgeLens [29] and Fisheye [14] have been proposed to provide detailed information as well as preserve the overall picture. We integrate two types of semantic zoom in our system to help the user filter out irrelevant data and focus on interested items.

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 map the concerned layers to the visualization cube and hide other layers in order to fully utilize the cube space. As illustrated in Figure 7, after the user performs a pinch gesture, the central three layers are relocated to the cube range, 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 redisplays all the data, therefore removing unconcerned objects and saving more space for the focused data items. Pinch gestures are easy to make on a multi-touch device, in which case the advantages of designed interactions are fully exploited.



(a) (b) (c) (d)

Figure 7: Two types of semantic zoom. (a)(b) Use two-finger pinch gesture to zoom scatterplot layers. Resultant layers that are outside the cube will be hidden to provide more space for focused layers. (c)(d) After a selection is done, a following two-finger pinch gesture is recognized as a semantic zoom, which hides all the unselected items for further investigating interested data items.

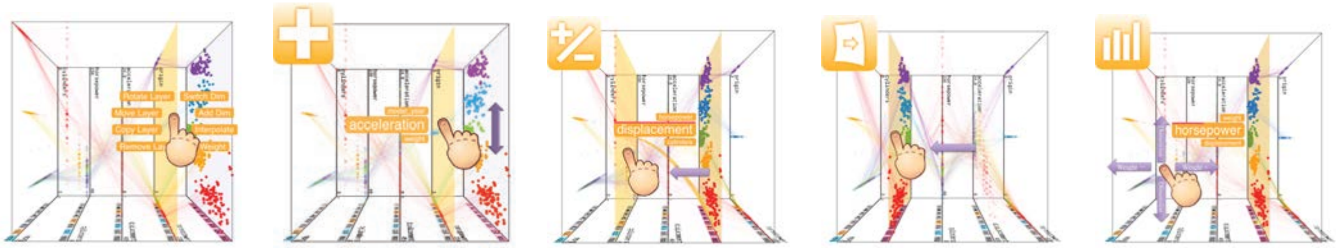
5 Application Case

Dimension selection and reduction are employed in many fields including mathematics solutions to real problems and computer simulations. We demonstrate a case of data investigation task in terms of feature selection, which exemplifies the application of MLMD in information processing.

The dataset in this case is the car data containin 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 forms five distinct clusters. To simplify the case description, we suggest that the user focuses on the bottom two clusters, which are colored red (Figure 8(a)). The task is to find out a set of representative dimensions that result in the difference between the two clusters.

At first the user uses two layers to view the dimension modification results. After a series of layer editing operations are performed, including using Copy Layer to generate copies of MDS, Add/Remove Dimension to modify the included dimensions of the new layers and Semantic Zoom on layers for plot relocations, the middle layer excludes the right three dimensions, while the bottom layer excludes the left two. It is obvious that on the bottom layer the two clusters are separated better (Figure 8(b)), which indicates that the data items vary more on the right three dimensions: Displacement, Cylinders and MPG. Then the user continues this process to achieve the result that applies only three dimensions to clearly separate these two clusters (Figure 8(c)). Layer rotation helps to make the clustered parallel coordinates segments look closer in this case. Curved interpolation can also be used here.

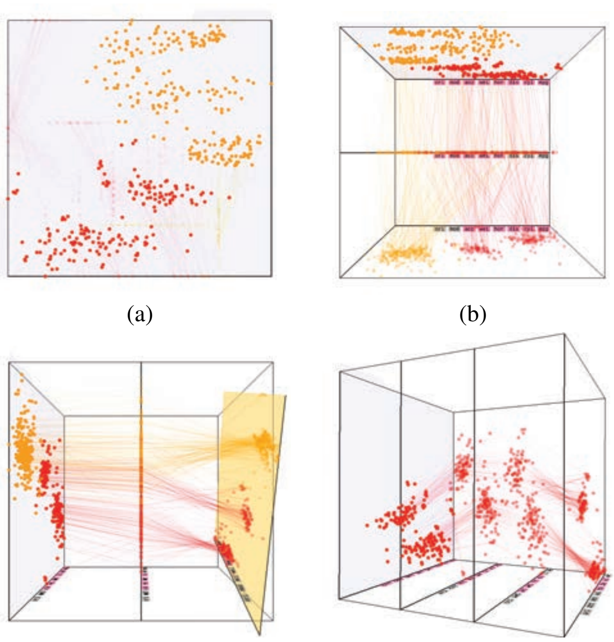
Finally, the user can review the overall picture of the dimension reduction process as is given in Fig. 8(d), in which semantic zoom on data items is applied to hide the uncared data items in order to provide a cleaner view. From left to the right, each layer is a step in the reduction procedure where human efforts are involved to optimize the dimension selection result. This example shows how the cube can help investigators fulfill their feature seletion



(a) (b) (c) (d) (e)

Figure 6: (a) Circular menu popped up. (b) Dimension selection (for Add operations, the dimension selections of the other operations are the same). (c) Move Dimension operation. (d) Move Layer opeartion. (e) Weight adjustment.

task. 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. All though the feature selection case, the interactions designed in our MLMD system largely accelerate the investigation and make the visualization space more accessible to the user.



(a) (b) (c) (d)

Figure 8: An example of applying MLMD to a dimension reduction task. (a) The original five clusters and the focused two clusters(red); (b) The right three dimensions separate the focused clusters better than the left two; (c) Final result that contains three representative dimensions is achieved; (d) All the steps of dimension reduction. Semantic zoom is applied to provide a clear overview.

6 Discussion

Comments received from volunteer users are summarized first in this section. The method scalability is then explained, followed by a discussion on future works based on the users' suggestions.

6.1 User Feedback

We ask a few volunteer users, who are students from the computer science department, to test our proposed visualization method and we collect their feedbacks. The trials are carried out both on standard PCs and on multi-touch devices. Although it takes certain time to get familiar with the system, the test participants all agree that the stacked plot metaphor is a novel approach to visualize multi-dimensional data in termof fully utilizing 3D space for layer comparison and naturally integrate multiple visualization metaphors. Most of the users have claimed that they successfully find out siginificant dimensions of the test data using our visualization. The system helps them determine whether an inclusion of a dimension for multi-dimensional scaling is appropriate and identify important attributes out of the others. While the calibrated stacked plots provide direct impressions on how the current dimension settings work, the parallel coordinates views are mainly responsible for coordinating and adjusting the system parameters. The curved interpolation between layers is a novel visualization feature for the testers, which provides an alternative view in addition to the standard parallel coordinates. Some test users state that the interpolation could be useful in finding out the occluded bundling and revealing desired intermediate settings when applied together with rotation and view alternation. The volunteer users espeically prefer the user-friendly interface of the system and mention that the incorporated interactions work well for the proposed MLMD method and effectively speed up the process of data exploration using MLMD. The frequently popped up circular menus, when working on a multi-touch device, is considered a satisfactory design as it saves the effort of gesture memorizing and at the same time reduce the operation time. The hint icons on operation status are also considered helpful as they enhance the user's feeling that the task being worked on is fully under control.

6.2 Scalability

The scalability of the proposed method is worth studying, which is also an included topic in the test users' feeback. Limited by the size of the display, it is impossible to show all the layers in the visualization cube since it would lead to severe occlusions. When the dataset consists of too many attributes, the dimension tags of the system would also lose efficacy as the size would be too small if equally devided. Therefore, the upper limit of the dimension number of the dataset and the maximum layers the virtual cube could hold are two importance indices of the designed MLMD system. We did not conduct a formal study on this question but based on the users' experiences, it can be basically concluded that the dimension number should not be greater than 20 and the 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 when necessary. Semantic zoom for dimension tags are not included in the current design, but the analogous method applies. However, although the semantic zoom helps resolve the limitations to certain extent, using semantic zooms unavoidably increase the operation time cost and consequently bring more difficulties for data exploration.

6.3 Future Works

We list several possible improvment of the MLMD method as the future works. Though being a new approach, the efficacy of the curved layer interpolation needs to be justified. Its ability of representing middle-state results and suitable curve type (passing the control points or not) are to be studied in addition to understanding its effectiveness of avoiding clutters and occlusions. Besides, alternative coloring criterion can be provided. For example, instead of item-based coloring, layer coloring is also a possible solution to provide better comparisons between different plot layers and is required by some of the testers. Finally, in the current design, layers are stacked into the cube to form an integration. As layers are quite independent components in the proposed system, an informative layer should be abe to be saved and shown at other area of the screen to enable more complicated investigation operations such as marking and comment editing. Extending the layer to present other plot including graph visualization and timeline visualization are also possible. The stacked layers can naturally present hierarchical information if the layers and the connections between which are carefully modified to serve that purpose. As most of the graphs include hierarchical structures, i.e. the social network involves people's working department, interest groups, etc., following the order of the stacked plots to trace downward those hierarchies would be prefferred. We would like to try these possibilities in our future work to extend our method for more complicated data investigations.

7 Conclusion

The visualization approach presented in this paper called MLMD 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 for multi-dimensional data based on its well designed interactions including navigation, brushing and layer editing. The proposed method can be applied for feature selection and dimension reduction, which are common tasks involved in large dataset processing. MLMD is a novel information visualization approach that extends the 2D visualization space into 3D. It compactly and interactively coordinates multiple views, and shows that 3D representation can be useful in the observation of multi-dimensional data when interaction methods are properly designed.

References

[1] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. AVI '00*, pages 110–119, 2000.

[2] P. Besl and H. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239 –256, feb 1992.

[3] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *Proc. ACM SIGGRAPH '93*, pages 73–80, 1993.

[4] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2nd edition, Aug. 2005.

[5] M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Proc. IEEE Vis '96*, pages 127–131, 1996.

[6] W. W.-Y. Chan. A survey on multivariate data visualization. Technical report, Hong Kong University of Science and Technology, 2006.

[7] W. C. Cleveland and M. E. McGill. *Dynamic Graphics for Statistics*. CRC Press, Inc., 1988.

[8] C. Collins and M. S. T. Carpendale. Vislink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1192–1199, 2007.

[9] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. Aug. 2000.

[10] T. Dwyer. *Two-and-a-Half-Dimensional Visualisation of Relational Networks*. PhD thesis, The University of Sydney, 2004.

[11] J. Edelmann, A. Schilling, and S. Fleck. The dabr - a multitouch system for intuitive 3d scene navigation. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, pages 1 –4, may 2009.

[12] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.

[13] E. Fanea, M. S. T. Carpendale, and T. Isenberg. An interactive 3d integration of parallel coordinates and star glyphs. In *Proc. IEEE InfoVis '05*, pages 149–156, 2005.

[14] J.-D. Fekete. The infovis toolkit. In *Proc. IEEE InfoVis '04*, pages 167–174, 2004.

[15] F. Guimbretiére and T. Winograd. Flowmenu: combining command, text, and data entry. In *Proc. ACM UIST '00*, UIST '00, pages 213–216, New York, NY, USA, 2000. ACM.

[16] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Proc. IEEE InfoVis '02*, pages 127–130, 2002.

[17] M. Heilig, S. Huber, M. Demarmels, and H. Reiterer. Scattertouch: a multi touch rubber sheet scatter plot visualization for co-located data exploration. In *Proc. ACM ITS '10*, pages 263–264, 2010.

[18] D. Holten and J. J. van Wijk. Evaluation of cluster identification performance for different pcp variants. *Computer Graphics Forum*, 29(3):793–802, 2010.

[19] D. Hopkins. The design and implementation of pie menus. *Dr. Dobb's J.*, 16:16–26, December 1991.

[20] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proc. IEEE Vis '90*, pages 361–378, 1990.

[21] J. Johansson, M. Cooper, and M. Jern. 3-dimensional display for clustered multi-relational parallel coordinates. In *Proc. IEEE InfoVis '05*, pages 188–193, 2005.

[22] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, Oct. 2002.

[23] D. A. Keim. Visual exploration of large data sets. *Communication of ACM*, 44:38–44, 2001.

[24] R. Kosara. Indirect multi-touch interaction for brushing in parallel coordinates. *Poster Proc. InfoViw '10*, 2010.

[25] G. Kurtenbach and W. Buxton. User learning and performance with marking menus. In *Proc. ACM SIGCHI '94*, pages 258–264, New York, NY, USA, 1994. ACM.

[26] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proc. ACM SIGCHI '94*, pages 318–322, 1994.

[27] C. Viau, M. J. McGuffin, Y. Chiricota, and I. Jurisica. The flowvizmenu and parallel scatterplot matrix: Hybrid multidimensional visualizations for network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1100–1108, 2010.

[28] R. Wegenkittl, H. Löffelmann, and E. Gröller. Visualizing the behavior of higher dimensional dynamical systems. In *Proc. IEEE Vis '97*, pages 119–125, 1997.

[29] N. Wong, S. Carpendale, and S. Greenberg. Edgelens: an interactive method for managing edge congestion in graphs. In *Proc. IEEE InfoVis '03*, pages 51–58, 2003.

[30] P. C. Wong and R. D. Bergeron. Multivariate visualization using metric scaling. In *Proceedings of IEEE Visualization 1997*, pages 111–118, 1997.

[31] L. Yu, P. Svetachov, P. Isenberg, M. Everts, and T. Isenberg. Fi3d: Direct-touch interaction for the exploration of 3d scientific visualization spaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1613 –1622, nov.-dec. 2010.

[32] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1001–1008, 2009.