Direct Ordering: A Direct Manipulation Based Ordering Technique

BHARAT KALE[a], MAOYUAN SUN[a] and MICHAEL E. PAPKA[a,b]

[a]*Northern Illinois University, DeKalb, IL, USA*
[b]*Argonne National Laboratory, Lemont, IL, USA*

Abstract

The ordering of data can reveal patterns in various data visualizations. It significantly affects the expressiveness of visualizations by making them clear or cluttered. Clutter makes it hard for users to perceive patterns, even for a small dataset. Almost all visualization tools available today support ordering through indirect manipulation techniques where users rely on a widget (e.g., a button or a checkbox) to perform ordering in some predefined way. For example, using widgets for sorting a selected subset of data involves creating widgets to filter the data, highlight the filtered data, and then apply the ordering. This work presents a new interaction technique based on direct manipulation, which supports flexible visual ordering. This direct manipulation-based interaction helps to visually filter the data and then order them. We demonstrate how the proposed interaction technique works with bar charts and matrices, two heavily used visualization techniques that involve ordering.

Keywords

Data Visualization, Direct Manipulation, Visual Ordering, Interaction

1. Introduction

Ordering is a critical function offered by various visualizations (e.g., axes ordering in parallel coordinates [1], entity ordering in multiple lists [2, 3], and matrix ordering [4]). It helps to organize visual elements in a meaningful way, revealing patterns that are hard to see otherwise. A flexible ordering capability potentially supports users exploring data from multiple perspectives. Thus, providing an interactive solution to achieve this goal is vital for the sensemaking of data.

Two types of interaction styles can be used in interactive visualizations: *indirect* manipulation, and *direct* manipulation. The former is enabled by using UI widgets. The latter requires interacting with visual elements directly. The Windows, Icons, Menus, Pointer (WIMP) interfaces created with indirect manipulation techniques require adjusting parameters, which draws user attention away from the visualization. Sometimes creating selections may be a problem for the WIMP interface. For example, users want to select a set of points in a scatter plot, but the points cannot be defined using a simple range filter. To achieve this behavior with indirect manipulation based techniques is tedious, as it involves developing widgets to define filters which are difficult to express, and then developing widgets to provide visual context and finally widgets to specify ordering. Instead, direct manipulation based techniques can offer a better solution by providing simple ways of selecting visual elements (e.g., using an interactive lasso selection [5]). Once a

Joint Proceedings of the ACM IUI 2021 Workshops, April 13–17, 2021, College Station, USA
bkale@niu.edu (B. KALE); smaoyuan@niu.edu (M. SUN); papka@niu.edu (M.E. PAPKA)
0000-0003-3653-228X (B. KALE); 0000-0002-0990-2620 (M. SUN); 0000-0002-6418-5767 (M.E. PAPKA)

selection is finished, it is possible to apply ordering with demonstrative interactions [6].

In this work, we present a direct manipulation based interaction technique to flexibly order visual elements by creating **direct ordering**. Direct ordering enables users to directly interact with the visual glyphs and manipulate the position channel to organize visual elements in a preferred order. Also, it allows applying the interaction on selections in the same way as that on the entire visualization. Applying the interactive ordering to selections (when present) has applications in places such as multiple coordinated views [7], in which selections applied in one view are reflected in other views. It is useful to have an interaction technique that can target interactions only on the selected data parts. In summary, this work highlights the following two contributions:

1) We present a direct manipulation based interaction technique for flexibly ordering visual elements.

2) We demonstrate the proposed technique for a bar chart and a matrix visualization.

2. Related Work

Ordering (also known as *seriation*) involves assigning an order to visually displayed data items. It has been studied in a close correlation with specific type of visualization (e.g., matrix visualizations [8, 4] and parallel coordinates [1]). They focus on improving the ordering algorithms but not on interactions to achieve some desired orders.

Direct manipulation highlights manipulating displayed visual elements directly. Shneiderman [9] states that visibility of the object of interest; rapid, reversible, and incremental actions are central ideas behind direct manipulation. It reduces user cognitive load by decreasing the distance between the source and target of the interaction [10]. Direct manipulation interfaces are getting popular nowadays as they remove the dependency on extra widgets (e.g., menus and buttons). Sarvghad et al. [11] created an embedded interactive technique to merge and split bars in bar charts and histograms for manipulating data groupings. DimpVis [12] allows direct interaction with various visual encoding channels to navigate visualizations in the temporal dimension. Interver [13] offers dynamic ordering as a user selects a numerical interval via brushing. Saket et al. [6] proposed a new paradigm, visualization by demonstration, which allows directly manipulating graphical encoding and recommends view transformations accordingly. Vuillemot and Perin [14] applied direct manipulation on ranking tables to navigate rows of interest temporally. Saket et al. [15] conducted a qualitative study on scatter-plot, bar chart, and histogram to understand how people convey the intended operations if they use only direct manipulation based interactions. Another interesting application of direct manipulation techniques is the *semantic interaction* introduced by Endert et al. [16]. It highlights that as users directly interact with visual elements, the meanings of such interactions should be considered as "soft data" by a visual analysis system, and such data further steers underlying computation models responsible for information foraging.

Support for ordering visual elements with direct manipulation based techniques has also been studied. Siirtola [17] proposed *reorderable matrix* as an interactive approach to apply Bertin's analysis [18] to physical matrices. Later Perin et al. [19] created *Bertifier* as a comprehensive implementation of Bertin's original ideas. Reorderable matrix [17] has limited functions but it is completely based on direct manipulation. In contrast, bertifier [19] sacrifices directness, to some extent, to support more features and achieve the desired results on multiple objects of interest at a time (e.g., replicating an operation on one row or column to the entire matrix). All these works are done only in the context of matrix visualization. Our goal is to develop an approach that can support flexibly ordering visual elements for a broad set of visualizations.

3. Requirement Analysis

Based on the literature on ordering in visualizations and the role of direct manipulation, we have derived the following requirements that our proposed interaction should provide:

R1: Allow interaction with visual representations. Using visual representations to enable user interaction solves the problem of shifting user focus [20].

R2: Ability to apply the interaction to a selected subset of data in a visualization. Users may want to see entire data in a visualization but emphasize on only parts of the data and further apply interactions to them.

R3: Provide visual cues to help users understand the available actions to achieve a desired interaction [21].

4. Direct Ordering

4.1. Design Considerations

To meet these requirements, we particularly focus on the following design considerations.

D1: Visual marks used in a visualization should be directly moved as desired without disturbing the relationship among data items **(R1)**. For example, moving a glyph to a new position requires adjusting all other glyphs in visualization not to lose the inherent relationship.

D2: Ordering should be supported via simple gestures **(R3)**.

D3: Show visual cues to help user understand *what* to interact with, *where* and *how* to perform an interaction **(R3)**.

D4: Apply the interaction to a selection. In case that there is no selection, use the entire chart as the selection **(R2)**.

4.2. Implementation

We implemented the technique in JavaScript and SVG using D3. The prototype we created for the demonstration uses drag functionality to allow users to interactively organize visual elements. The drag event tracks the state of visual elements to detect cases, such as sorting in ascending or descending order in a bar chart and clustering in a matrix. Also, during the ordering process, visual cues are offered to handle ambiguity. For example, consider two scenarios: 1) moving the tallest bar to the first position, and 2) ordering the entire bar chart in descending order. Both cases require a user to move the tallest bar to the left side. To resolve such ambiguity, as a user starts moving the tallest bar, left and right most boundaries appear to indicate that moving the tallest bar beyond them only triggers the sorting operation (scenario 2) otherwise, a simple position change (scenario 1).

4.2.1. Automated vs. Manual Ordering

While automated ordering has benefits such as speed and reproducibility, it brings problems like interpretability when the number of dimensions is high. It is difficult to include user domain knowledge in an automated ordering process. On the other hand, manual ordering is time-consuming. Hence, direct ordering employs a combination of automated and manual ordering. Direct Ordering first uses an automated approach to provide a preliminary order. Then users can interactively adjust the automated ordering results.

4.2.2. All vs. Selections Ordering

The proposed technique applies ordering to visual glyphs corresponding to all data points if either the selection has all the data or there is no selection at all. With selection, the technique can restrict the ordering only to visual elements in the selection.

In next two sections, we present the implementation of direct ordering in bar charts and matrices. Our implementation gives initial evidence that it is possible to understand user intention based on user interactions and is a first step toward exploring the visualization by demonstration [6] interaction paradigm.

5. Use Case: Bar Charts

A bar chart encodes data values with height and position of bars. It is used to compare values, so ordering is necessary. As ordering in a bar chart is in one dimension, dragging along one axis can achieve a desired result.

5.1. Changing Bar Position

To change position of a bar, a user drags the bar to a desired position **(D1)**. When the cursor is hovered on a bar, it gets highlighted indicating that a user can interact with it **(D3)**.

Sorting is a particular case of ordering, where all items are ordered in either an ascending or descending order. While users can perform sorting by dragging each bar, it is time-consuming. We supported sorting with a simple gesture of dragging the tallest bar to the leftmost (descending) or rightmost (ascending) end of a bar chart **(D2)**. Figure 1 (A) shows the sorting operation in ascending order. Moreover, animated transitions are applied to help users understand the changes.

Figure 1: Direct ordering in a bar chart: ordering all items (A), changing the position of a bar (B), ordering all items in-between two selected items (C), ordering neighboring selected items (D), and ordering non-neighboring selected items (E).

5.2. Applying on Selections

Direct ordering allows applying it to a selection in the same way as that to an entire bar chart. Figure 1 (C, D, and E) shows how direct ordering works on a selection in a bar chart. The selections, in figure 1, are created by clicking on individual bars. With a selection, as user starts moving a bar, boundaries are shown beside the left and right most bars in the selection. This enables sorting a selection by dragging the tallest bar (in the selection) beyond the boundaries of this selection **(D4)**. This avoids dragging the tallest bar to the boundaries of the entire chart even when the area of interest is very small. Also, in a selection, the bars within the selection can be either contiguous or dis-contiguous. In either case, only bars in a selection will be ordered (i.e., in a case of dis-contiguous bars, the sorting does not make the selected bars contiguous as is shown in Figure 1 (E)).

5.3. For Grouped Bar Charts

Direct ordering can be applied to a grouped bar chart, as shown in Figure 2 (B1-B4). Groups can be sorted by selecting a criteria like a common sub-type across groups (see Figure 2 (B1-B2)). In Figure 2 (B1), there is no specific ordering in groups. Once a bar is selected, *65 Years and Over* in this case, bars that belong to the same type across other groups are highlighted. Now treating each group as a single bar in a simple bar chart, the groups can be rearranged using the group containing the tallest bar among the selected bars. Figure 2 (B2) shows the result of sorted groups in a descending order of *65 Years and Over*. Moreover, we can propagate the ordering applied to bars in one group to all other groups. In Figure 2 (B3), bars in the group *CA* are sorted in descending order and the same order is propagated across all other groups (see Figure 2 (B4)). This follows the paradigm of visualization by demonstration [6], as the system learns from user interaction in one group and applies that to others.

6. Use Case: Matrix

Matrix has been used to display network data and tabular data. The data values, in case of tabular data, and relationships, in case of network data, are encoded in cells using colors or visual glyphs. Bertin's matrices [18] is a typical example. Ordering is critical in a matrix to reveal clusters, and it is performed in two dimensions. Hence the proposed approach needs two drag operations, one along each dimension, to achieve the desired result.

6.1. Rearranging rows/columns

To re-position a row/column, a user needs to drag and drop it. Figure 3 (A, B) shows dragging a column and a row to new positions, respectively. It acts as a fundamental interaction and allows manually adjusting the results of an automatic ordering. We used the initial dragging direction to restrict the drag operation to either row or column. For example, if an initial dragging is close to moving up or down, we use only row dragging, otherwise column. In dragging, animated transitions are used to help users understand and follow changes.

6.2. Using Visual Similarity

A selected set of rows/columns are automatically organized by visual similarity [19]. We use differences in visual encodings instead of raw data values to compute similarity. The visual similarity computation algorithm takes two steps. First, it takes visual encoding values of a selected set of rows or columns as input vectors. For example, as each column is encoded independently, we used the circles' radius as the input vectors for computing distance. Second, using the Euclidean distance metric, it computes an optimal order that minimizes the sum of distances between consecutive vectors [22].

To enable multi row/column ordering, we used a design similar to *Crossets* [23]. Selecting a row/column header and dragging to consecutive row/column headers highlights and adds them to a selected subset, which is then sent to the similarity calculation algorithm. In Figure 3, (C) shows column based ordering and (D) presents row based ordering. As updates are incremental, applying a row based ordering followed by a column based or vice-versa organizes a matrix.

The result shown in Figure 2 (C3) is obtained by applying a 2D ordering to the matrix shown in Figure 2 (C1). First, the matrix is ordered by visual similarity across all the columns (Figure 2 (C2)), and the resulting matrix is ordered again using all rows. Finally, if a user wants to tweak the automated

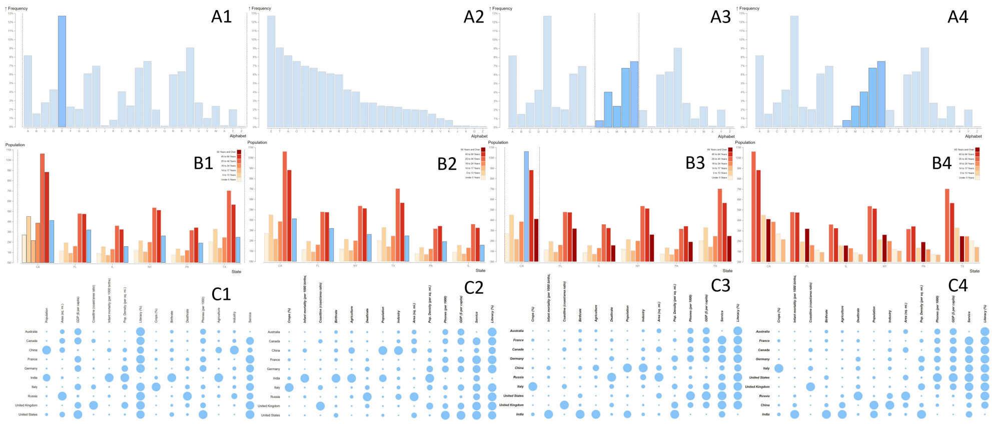

Figure 2: Direct ordering on a bar chart (A1-A4), a grouped bar chart (B1-B4), and a matrix (C1-C4): ordering all bars in a descending order using the tallest bar and the bounds (A1-A2), ordering selected bars in an ascending order using the tallest bar in the selection and the bounds (A3-A4), ordering all groups in a descending order of selected type using the group with the tallest bar of the selected type and the bounds (B1-B2), propagating the order of bars in one group to all groups using the tallest bar in the selected group and its bounds (B3-B4), ordering all columns in a matrix based on visual similarity (C1-C2), ordering a matrix with an automatic ordering algorithm – first by visual similarity of columns and then rows to reveal clusters (C3), manually fine-tuning automated ordering results (C4).

ordering results, the manual ordering that we supported helps. Figure 2 (C4) shows some of the rows from the Figure 2 (C3) are manually adjusted. This function offers more flexibility.

7. Initial Expert Feedback

To evaluate direct ordering, we showed it to a domain expert in information visualization and visual analytics. The feedback indicated the usefulness of the technique in terms of flexibly organizing visual elements. During the evaluation, similar to those displayed in Figure 2, three demonstrations are explored: *1) ordering bars in a bar chart, 2) ordering groups and bars with in a group in a group bar chart, and 3) ordering rows and columns in a matrix*. Key positive comments from the feedback include:

"It's really good for me to flexibly order any bars. This explicit manual ordering implies user intention and capturing that will be helpful." [**demos 1, 2**].

"This technique can help with matrix ordering, such as after an automatic matrix sorting, user manually adjust the order. If such user adjustment can be analyzed and sent back to the automatic ordering algorithms, then we can iteratively order the whole matrix." [**demos 3**].

Along with the positive comments, suggested improvements are listed as follows.

"Why does it only support using the highest bar, not the shortest one? Finding the highest bar needs cognitive effort." [**demos 1, 2**].

"It would be better to use gestures for ordering neighboring bars." [**demos 1, 2**].

"Can I brush on the y-axis, and ordering bars based on my selection?" [**demos 1, 2**].

"Why does it only allow me sorting neighboring rows/cols? The ordering behavior for non-neighboring rows/cols should be consistent with that in bar chart." [**demos 3**].

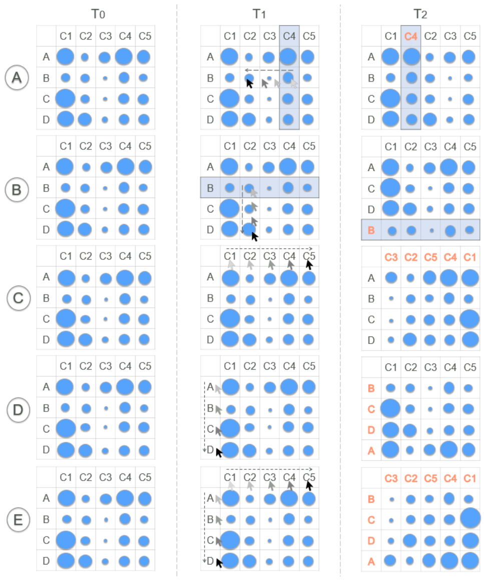

Figure 3: Direct ordering in a matrix: manually changing the order of a column (A), a row (B), ordering all columns (C), all rows (D), and all columns followed by all rows (E) by visual similarity.

8. Conclusion

We presented *direct ordering*, a direct manipulation based interactive technique to order visual elements in a visualization. We implemented the technique on bar charts and matrices and evaluated the technique by gathering an initial expert feedback. The feedback shows interest and potential usefulness of the proposed interaction technique in terms of its flexibility and ability to understand user intention. We believe our proposed technique can help design and build intelligent interactive systems, e.g., designing consistent ways of organizing visual elements in well-aligned 1-D and 2-D layouts, enriching the design space of semantic interactions and visualization by demonstration by contributing novel direct manipulation based techniques to order visual elements, and help in designing techniques to understand user intention by capturing and analyzing user interactions and recommending new possible transformations matching user interest. While this work demonstrates the use of interactive ordering, there are few limitations that need to be studied further.

First, test possible generalizability of direct ordering to a variety of visualizations. This work can be extended to conduct a study on applying it to diverse visualizations (e.g., parallel coordinates, pie charts, and radial bar charts). Similarly, conducting a study to understand its application for existing visualization tools and analyzing the differences in flexibility and ease of use helps further identify trade-offs of using it.

Second, the initial evaluation of direct ordering came from one domain expert only. To gain an in-depth understanding of this technique's impact on the interactive user interface, we need to conduct user studies in the future, using diverse real-world datasets. One key area we plan to focus on moving forward is understanding how to capture user intention based on user interactions. We could then recommend further possible interactions and transformations that might be of interest to the user. Thus, instead of merely updating a view after the user actions, we can empower the user interface with intelligence to suggest users with possibilities.

9. Acknowledgments

This research was supported in part by the NSF Grant IIS-2002082 and the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357.

References

[1] L. F. Lu, M. L. Huang, T.-H. Huang, A new axes re-ordering method in parallel coordinates visualization, in: 2012 11th International Conference on Machine Learning and Applications, volume 2, IEEE, 2012, pp. 252–257.

[2] J. Stasko, C. Görg, Z. Liu, Jigsaw: supporting investigative analysis through interactive visualization, Information visualization 7 (2008) 118–132.

[3] M. Sun, J. Zhao, H. Wu, K. Luther, C. North, N. Ramakrishnan, The effect of edge bundling and seriation on sensemaking of biclusters in bipartite graphs, IEEE transactions on visualization and computer graphics 25 (2018) 2983–2998.

[4] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, J.-D. Fekete, Matrix reordering methods for table and network visualization, in: Computer Graphics Forum, volume 35, Wiley Online Library, 2016, pp. 693–716.

[5] M. J. McGuffin, I. Jurisica, Interaction techniques for selecting and manipulating subgraphs in network visualizations, IEEE Transactions on Visualization and Computer Graphics 15 (2009) 937–944.

[6] B. Saket, H. Kim, E. T. Brown, A. Endert, Visualization by demonstration: An interaction paradigm for visual data exploration, IEEE transactions on visualization and computer graphics 23 (2016) 331–340.

[7] J. C. Roberts, State of the art: Coordinated & multiple views in exploratory visualization, in: International Conference on Coordinated and Multiple Views in Exploratory Visualization, IEEE, 2007, pp. 61–71.

[8] I. Liiv, Seriation and matrix reordering methods: An historical overview, Statistical Analysis and Data Mining: The ASA Data Science Journal 3 (2010) 70–91.

[9] B. Shneiderman, 1.1 direct manipulation: a step beyond programming languages, Sparks of innovation in human-computer interaction 17 (1993) 1993.

[10] E. L. Hutchins, J. D. Hollan, D. A. Norman, Direct manipulation interfaces, Human–Computer Interaction 1 (1985) 311–338.

[11] A. Sarvghad, B. Saket, A. Endert, N. Weibel, Embedded merge & split: Visual adjustment of data grouping, IEEE transactions on visualization and computer graphics 25 (2018) 800–809.

[12] B. Kondo, C. Collins, Dimpvis: Exploring time-varying information visualizations by direct manipulation, IEEE transactions on visualization and computer graphics 20 (2014) 2003–2012.

[13] M. Sun, G. Convertino, Interver: Drilling into categorical-numerical relationships, in: Proceedings of the International Working Conference on Advanced Visual Interfaces, 2016, pp. 44–51.

[14] R. Vuillemot, C. Perin, Investigating the direct manipulation of ranking tables for time navigation, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 2703–2706.

[15] B. Saket, S. Huron, C. Perin, A. Endert, Investigating direct manipulation of graphical encodings as a method for user interaction, IEEE transactions on visualization and computer graphics 26 (2019) 482–491.

[16] A. Endert, P. Fiaux, C. North, Semantic interaction for visual text analytics, in: Proceedings of the SIGCHI conference on Human factors in computing systems, 2012, pp. 473–482.

[17] H. Siirtola, Interaction with the re-

orderable matrix, in: IEEE International Conference on Information Visualization, IEEE, 1999, pp. 272–277.

[18] J. Bertin, Graphics and graphic information processing, Walter de Gruyter, 2011.

[19] C. Perin, P. Dragicevic, J. D. Fekete, Revisiting bertin matrices: New interactions for crafting tabular visualizations, IEEE Transactions on Visualization and Computer Graphics 20 (2014) 2082–2091.

[20] B. Saket, A. Srinivasan, E. D. Ragan, A. Endert, Evaluating interactive graphical encodings for data visualization, IEEE transactions on visualization and computer graphics 24 (2017) 1316–1330.

[21] D. Norman, Things that make us smart: Defending human attributes in the age of the machine, Diversion Books, 2014.

[22] Z. Bar-Joseph, D. K. Gifford, T. S. Jaakkola, Fast optimal leaf ordering for hierarchical clustering, Bioinformatics 17 (2001) S22–S29.

[23] C. Perin, P. Dragicevic, J.-D. Fekete, Crossets: Manipulating multiple sliders by crossing, in: Proceedings of the 41st Graphics Interface Conference, 2015, p. 233–240.