

Toward Using Matrix Visualizations for Graph Editing

Stefan Gladisch*
University of Rostock

Heidrun Schumann
University of Rostock

Martin Luboschik
University of Rostock

Christian Tominski
University of Rostock

ABSTRACT

Node-link and matrix representations are widely applied for graph exploration. However, when it comes to editing graphs, matrix representations are mostly neglected. In this work, we investigate the suitability of matrix representations especially for graph editing. Based on a review of the characteristics of matrices in terms of representation and interaction, we propose first techniques for edge-based editing tasks. We combine our matrix-based techniques with classic node-link solutions in an interactive prototype for touch-enabled graph editing.

1 INTRODUCTION

On a most abstract level, visual methods for graph exploration can be categorized into node-link and matrix representations. Both have their individual advantages and disadvantages and support different exploration tasks [2]. Ideally, node-link and matrix are used in combination for graph exploration [3].

What is widely accepted for graph exploration is not so clear for graph editing. In existing solutions for editing graphs, node-link is often the only type of representation supported. Matrix representations have not been considered so far, although it has been indicated that different visual representations have different strengths and weaknesses for editing tasks as well [1].

Our aim is to ease editing (especially the editing of edges) by extending our considerations to matrix representations. To support editing tasks well, two requirements have to be met. The data must be presented effectively and easy to interact with.

We start with analyzing node-link and matrix representations according to these requirements. It turns out that matrices are better suited for edge-based editing tasks. Based on a set of requirements for these particular editing tasks, we introduce first editing techniques using matrix representations. We developed a prototype where node-link and matrix representations complement each other, and together enable easier editing of graphs.

2 NODE-LINK AND MATRIX REPRESENTATIONS

Node-link and matrix representations have specific characteristics that make them suitable or unsuitable for specific tasks.

In node-link representations, nodes are typically visualized as shapes (e.g., circles) that are connected by lines depicting edges. Provided that a suitable layout algorithm is applied, node-link representations offer an intuitive overview of sparse graphs and support the identification of paths. Dense graphs can lead to severe edge clutter, hampering overview and path-based tasks [2] as well as the identification of existing and non-existing edges.

As node-link representations explicitly show nodes and edges, it is possible to interact with them. While interaction with node shapes is usually easy to accomplish, the interaction with edges can be difficult, because lines are not as easy to pick as shapes. Edge clutter aggravates this issue. This is especially relevant for touch interaction.

Matrix representations directly visualize a graph's adjacency matrix. Nodes are represented as columns and rows of the matrix. Each cell (i, j) of the matrix depicts whether there is an edge between the i -th and the j -th node. When rows and columns are ordered appropriately, matrices can provide a good structural overview of the data and allow us to find existing or non-existing edges quickly. Compared to node-link representations, edge crossings or overlaps cannot occur, and hence, edge visibility is guaranteed at all times, and ambiguities are avoided. However, this comes at the costs of quadratic space requirements. An advantage of matrix representations is that they are particularly effective for dense graphs. However, they are ill-suited for path-based tasks [2].

Concerning interaction, matrix representations have their strength for edge-based tasks. As a matrix cell corresponds one-to-one with an edge, edge insertion and deletion can be designed as interactions that are quite easy to carry out. Moreover, matrix cells are easier to pick than lines in node-link representations. On the other hand, nodes are only implicitly represented as rows and columns, which requires special treatment of node-based tasks.

As we see, node-link and matrix representations offer complementary ways to represent and interact with graphs. While node-link representations are good when the focus is on graph nodes, matrix representations have their strengths for edges. In the next section, we study techniques to better support edge-based graph editing tasks with matrix representations.

3 GRAPH EDITING USING MATRIX REPRESENTATIONS

As indicated, matrices are particularly suited for edge-based editing tasks. Here, we focus on edge insertion and deletion. Our interaction design has to address the following requirements:

- R1: Interactions should be familiar to the user.
- R2: Interactions should avoid editing by accident.
- R3: Interactions should not conflict with other interactions commonly used for exploration tasks.

To meet the first requirement (R1), we rely on matrix cells as primary targets for the interaction and use them as a kind of interactive check boxes. The idea of check boxes is used widely in graphical user interfaces, and users are familiar with it. When an edge exists, its cell is "checked" and filled with a color. When an edge does not exist, the corresponding cell is "unchecked" and left blank. Inserting and deleting edges can now be reduced to checking and unchecking matrix cells. Checking or unchecking are typically triggered by a simple click or tap, which are quite short interactions. However, to avoid editing by accident (R2) and to be conflict-free with respect to interactions for exploration tasks (R3), we require the user to carry out a longer press on a matrix cell. The longer time for the interaction also communicates the significant impact of the interaction, namely the manipulation of the underlying data. When applying a long press on a blank cell, the edge is inserted, and when the long press is applied to a checked cell, the edge is deleted. Figure 1 illustrates the interaction.

For inserting or deleting multiple edges, one can use the technique described above multiple times, but then the editing effort is multiplied as well. In light of R1, one could use traditional

*e-mail: stefan.gladisch@uni-rostock.de

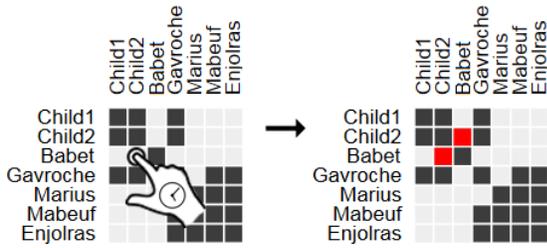


Figure 1: Inserting a single undirected edge (highlighted in red) in a matrix representation using a long press touch-gesture.

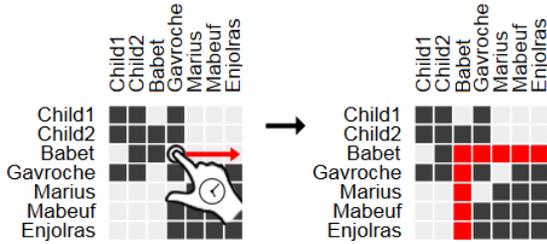


Figure 2: Inserting multiple undirected edges (highlighted in red) using a combination of long press and drag touch-gestures.

techniques for multi-selection, which would result in checking or unchecking multiple cells at once. Examples include interaction with an additional modifier key (e.g., use SHIFT to un/check everything between two consecutive presses) or spatial selection via elastic rectangle or lasso. Yet, modifier keys are not always available (e.g., on tablets or tabletops), and the overhead of using a 2D spatial interaction is not really necessary in our scenario. Therefore, we design the interaction as a simple long press (R2) followed by a 1D-restricted drag (R2). The long press marks the first edge to be edited and the drag along all other cells (row-wise or column-wise) marks further edges to be affected (see Figure 2). This results in inserting/deleting all edges corresponding to the marked cells. An advantage of this technique is that multiple edges can be inserted or deleted quite quickly. A drawback is that rows and columns must be ordered appropriately. As this is not always the case, the approach presented next includes a reordering technique.

4 MATRIX AND NODE-LINK REPRESENTATIONS COMBINED

To demonstrate that combining node-link and matrix representation can effectively support the editing of graphs, we implemented a prototype application for an android tablet using these representations as interlinked views. Both representations support the typical pan and zoom navigation for exploration tasks. The node-link view is equipped with traditional editing techniques for inserting and deleting nodes. The matrix view implements the described interaction techniques for edge editing.

To take full advantage of using both node-link and matrix, the views are linked in a special way. When a set of nodes is lasso-selected in the node-link representation, the matrix rows and columns are automatically reordered so that the selected nodes are placed in the top left corner of the matrix. This makes it easy to apply the multi-edges editing technique, as there is no need to reorder the matrix by hand. Further the matrix cells are enlarged similar to a focus+context technique. This design has been chosen to circumvent common problems with touch precision. When deselecting nodes, the matrix ordering is reset to its original state. This is important to reduce interference with the users mental map of the matrix. Figure 3 shows our prototype during the insertion of multiple edges in the matrix view.

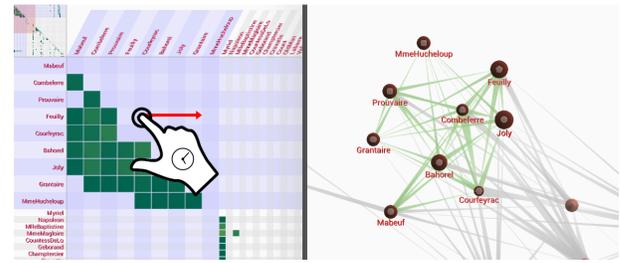


Figure 3: Edge editing in an interlinked view setting. Highlighted nodes and edges have been previously selected in the node-link representation. The matrix representation where the editing takes place, has been reordered accordingly. The underlying graph is directed.

The usefulness of our combined solution shall be illustrated with a small example. Consider a user exploring a social network. As the user’s task is to follow paths between different communities, the node-link view is used in the beginning. During the exploration, the user stumbles upon a community where a special person and its connections to all other persons in the community is missing. To insert this person’s node, the user simply applies a double-tap on an empty position. To insert all missing connections, the user can now benefit from the interlinked matrix view. First, the community is lasso-selected in the node-link view. The matrix is then reordered automatically. Inserting the connections is then as easy as applying the multi-edge technique to the appropriately reordered matrix. Because the matrix communicates the dense structure of the community well, the effect of the editing is immediately visible. After confirming that the community is now correctly connected, the user resets the matrix ordering and continues the exploration.

5 CONCLUSION AND FUTURE WORK

In this work, we investigated matrix representations for graph editing. We compared the characteristics of matrix and node-link representations and concluded that both complement each other in terms of representation and interaction. To better support edge-based editing tasks, we proposed matrix interaction techniques. A working prototype shows that interlinked node-link and matrix representations can be beneficial for graph editing.

For future work, we plan to better integrate node-link-based and matrix-based editing by following a NodeTrix design [3]. We would also like to consider the editing of edge attributes while inserting edges into the graph. An option would be to investigate bi-manual touch-interaction. Based on the improved implementation, user studies can be carried out to quantify the pros and cons of matrices and node-link representations for graph editing tasks.

ACKNOWLEDGEMENTS

We thank Benno Waldhauer for implementing the interactive prototype. This research has been supported by the German Research Foundation (DFG) under grant number DA 1319/3-1.

REFERENCES

- [1] T. Baudel. From Information Visualization to Direct Manipulation: Extending a Generic Visualization Framework for the Interactive Editing of Large Datasets. In *Proc. of ACM Symp. User Interface Softwa. Techn. (UIST)*, pages 67–76. ACM Press, 2006.
- [2] M. Ghoniem, J.-D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135, July 2005.
- [3] N. Henry, J.-D. Fekete, and M. J. McGuffin. Nodetrix: A hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.