Visualizing Dense Dynamic Networks with Matrix Cubes

Benjamin Bach, Emmanuel Pietriga, Jean-Daniel Fekete

Visualizing Dense Dynamic Networks with Matrix Cubes

Benjamin Bach* — INRIA
Emmanuel Pietriega† — INRIA, INRIA Chile.
Jean-Daniel Fekete‡ — INRIA

ABSTRACT

Visualizing static networks is already difficult, but exploring dynamic networks is even more challenging due to the complexity of the tasks involved; one visual encoding will hardly fit all tasks effectively, hence multiple complementary views are needed. We introduce the Matrix Cube, a visualization and navigation model for dynamic networks that results from stacking adjacency matrices, one for each time step in the network. It builds on our familiarity with cubes in the physical world and offers intuitive ways to look at, manipulate and decompose them. We describe a set of operations to decompose the Matrix Cube and interact with the resulting views.

Keywords: Dynamic Networks, Matrices, Metaphors

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical User Interfaces (GUI)

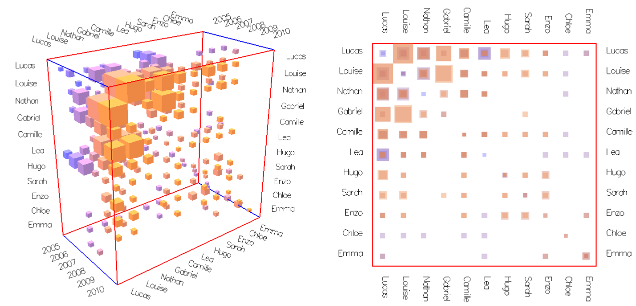
1 INTRODUCTION

Dynamic networks are networks whose topology and attributes change over time: vertices enter and leave the network, edges get added and removed, and values associated with vertices and edges vary with time. Such changes can be observed at a low level of abstraction, looking at individual elements, or at a higher level, observing how clusters emerge and disaggregate over time.

Currently, exploring dynamic networks requires complex interfaces that rely on multiple visual representations such as node-link diagrams, timelines, animations [8, 1], and small multiples [3, 7] coordinated using brushing and linking. Small multiples can provide an overview of the network over time but are often too small for detailed analysis. Animations require users to navigate along time and mentally relate elements across steps. Other techniques used the third spatial dimension as a time axis so as to show both topological *and* temporal information simultaneously [6, 4]. However, this approach suffers from the usual pitfalls of 3D visualization: complex virtual navigation, and issues of visual occlusion that make the representation illegible when networks become dense.

The *Matrix Cube* is designed to replace these multiple representations with a unified visual and mental representation upon which we build a simple yet powerful interface for exploring complex and dense temporal networks. It consists of a space time cube that represents dynamic networks by stacking adjacency matrices, one for each time step of the network. Visualization and interaction with the cube builds on our familiarity with basic geometric objects in the physical world and offers intuitive ways to look at, manipulate and decompose it. Rather than directly visualizing the three-dimensional , our interface derives meaningful and readable two-dimensional views by employing simple operations that could be performed on a tangible cube: rotation, projection, slicing, pruning. We rely on this tangible metaphor to help users construct a mental map of the dynamic network. The supports all of the conceptual views commonly encountered in dynamic network visualization, such as small multiples, aggregated images and time-multiplexing. We further provide novel, richer views that are helpful for exploration yet easy to relate to the cube.

*e-mail: benjamin.bach@inria.fr
†e-mail: emmanuel.pietiga@inria.fr
‡e-mail: jean-daniel.fekete@inria.fr



(a) (b)

Figure 1: Matrix Cube showing a co-authorship network. (a) 3D view, (b) rotation and projection on the Nodes × Nodes plane (authors).

2 MATRIX CUBE

Matrices have been shown to outperform node-link diagrams for many tasks on dense networks [9]. Our work investigates the possible benefits of using matrices for visualizing dynamic networks.

Each matrix represents a *time slice* that corresponds to the network's configuration in one time step. cube shows a co-authorship network: two of the cube's dimensions correspond to authors; the third dimension corresponds to time (years in this particular case). Each cell inside the cube represents a collaboration between two authors on a paper during the corresponding year. The larger the cell, the more papers the two authors have collaborated on during that year. Cell color depends on the cell's position along the time axis, providing a redundant encoding of this information that helps users orientate themselves when rotating and decomposing the cube: blue cells correspond to early time steps, orange cells to later ones.

The Matrix Cube represents the central visual metaphor and serves as a pivot when switching between any two of the possible views. Its 3D representation provides an overview of the dataset, allowing users to estimate the number of nodes and time steps, edges, distribution of edges and their density. It enables them to visually identify topological and temporal clusters, as well as possible outliers. However, this type of 3D view has numerous drawbacks, some of which were mentioned earlier: visual occlusion, tedious virtual navigation, and distorted sizes due to perspective projection. Our interface lets users derive more readable and meaningful two-dimensional views from the cube by interacting with it, letting them switch between views whenever tasks require a different perspective on the data. Animations are played between any two views to show the corresponding transformations performed on the cube, helping users keep oriented within the representation and understand the relationships between the views.

3 VIEW DESIGN SPACE

Various views can be derived from the matrix cube by applying a combination of operations to it: rotation, projection, slicing, filter-

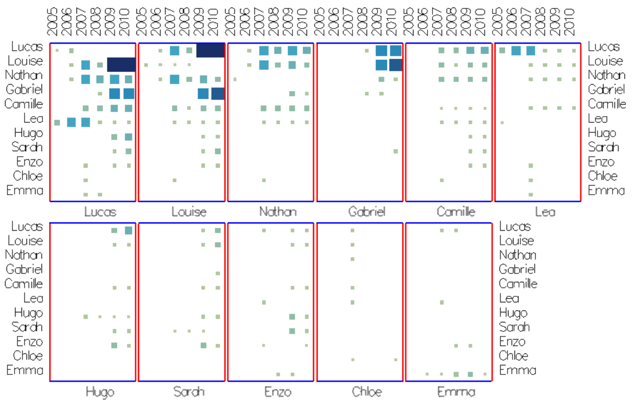
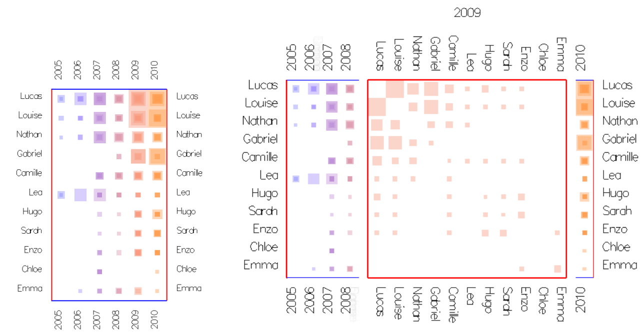


Figure 2: Small multiples of vertex slices using value cell encoding. Cells are colored using value encoding to highlight differences in weight across slices.



(a) Original *Time* × *Nodes* projection

(b) *Time* × *Nodes* projection + slice rotation

Figure 3: Focus and context view of a time slice. (a) Time × Nodes projection, (b) after rotating time slice for 2009.

ing, layout and flip-through.

3D and Rotation — The 3D view can be rotated, panned and zoomed freely. When hovering cells, the corresponding labels for nodes and time get highlighted to make their identification easier.

Projection — front shows the cube, rotated and projected so that it shows the adjacency matrix with all *time slices* superimposed using alpha blending. Cells inside the cube become translucent, and provide a summary of the connections between any two nodes over time. Topological clusters emerge from reordering rows and columns. Thanks to color encoding, clusters and connections can be roughly situated in time. When rotating the cube 90 degrees around its vertical axis, a projection as shown in side is obtained. Rows still show nodes, but columns represent time, one column per time step. Cells in this view summarize all connections of a node over time steps. This view makes it easier to track the overall evolution of each node's degree, and to identify time steps (years in our example) that feature the denser or sparser networks.

Layout as small multiples — A common technique to show spatio-temporal data is to use small multiples, one per time step. Since time in the matrix cube is discrete, decomposition is straightforward. The cube gets sliced into adjacency matrices, one for each time step. The matrices are juxtaposed, allowing for pairwise comparison and individual analysis. vertex-sm shows *node-slices* (as opposed to *time-slices*) as small multiples for our earlier example. Cell color is now mapped to cell size to allow for better cross-slice comparison. Each node-slice shows the dynamic ego-network for each node, enabling the comparison of individual connection patterns across nodes of the network.

Focus+Context — Small multiple representations do not scale to large networks or networks with many time steps. By projecting the cube on either side, users can thence rotate individual time slices (in a node × node projection) or node slices (in a time × node) projection. Rotating a slice can be compared to rotating a single book on a book shelf; after rotation, the slice can be observed individually while context of the current view is preserved. f+c shows the rotation of a time slice from the view in side.

Slide Show — Yet another way to explore individual slices is to present them one at a time, such as in a flip-book or a slide show. From the 3D view, users can select any single node slice or time slice, consequently hiding all other slices. They can then rotate the cube to obtain a projection that shows only the selected slice. Pressing the *up* and *down* arrow keys switches to the next or previous slice, allowing for the direct comparison of two adjacent slices.

4 Discussion

We believe that matrix cubes provide an interesting alternative to node-link diagrams for the visualization of dynamic networks, especially for dense networks. Decomposing the cube and obtaining matrices allows for occlusion free legibility. Another strength of the approach is that it relies on an easy-to-understand, yet powerful unified metaphor based on the manipulation of cubes.

A crucial feature of our matrix cubes are the interactive navigation and decomposition techniques. The original 3D representation should mainly be seen as a pivot rather than an actual view (beyond providing an overview of the dataset). We currently use simple commands to switch between views. The design space of views that can be derived from the cube is quite rich, actually, and one of the next steps is to investigate which input modalities, such as touch, can enable users to more efficiently manipulate matrix cubes and switch between views on them. Further we investigate alternatives to translucent cells in the projected views to visually represent quantitative information about edges using novel encodings for temporal glyphs [10, 5, 2].

References

[1] J. Ahn, M. Taieb-Maimon, A. Sopan, C. Plaisant, and B. Shneiderman. Temporal Visualization of Social Network Dynamics: Prototypes for Nation of Neighbors. In *Proc. of Social Computing, Behavioral-Cultural Modeling and Prediction conference*, pages 309–316, 2011.

[2] B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete. Weighted Graph Comparison Techniques for Brain Connectivity Analysis. In *Proc. of CHI*, pages 483–492. ACM, 2013.

[3] I. Boyandin, E. Bertini, P. Bak, and D. Lalanne. Flowstrates: An approach for visual exploration of temporal origin-destination data. *Computer Graphics Forum*, 30(3):971–980, 2011.

[4] U. Brandes and S. R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization*, 2(1):40–50, Mar. 2003.

[5] U. Brandes and B. Nick. Asymmetric relations in longitudinal social networks. *IEEE TVCG*, 17(12):2283–2290, Dec. 2011.

[6] T. Dwyer and P. Eades. Visualising a fund manager flow graph with columns and worms. In *Proc. IV*, pages 147–152, 2002.

[7] M. Farrugia, N. Hurley, and A. Quigley. Exploring temporal ego networks using small multiples and tree-ring layouts. In *Proc. ACHI*, 2011.

[8] C. Friedrich and P. Eades. The marey graph animation tool demo. In *Proc. Graph Drawing*, pages 396–406. Springer-Verlag, 2001.

[9] 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, 2005.

[10] J.-S. Yi, N. Elmqvist, and S. Lee. Timematrix: Visualizing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction*, 26(11-12):1031–1051, 2010.