**N. Henry, J.-D. Fekete, and M. J. McGuffin. NodeTrix: a hybrid visualization of social networks. *IEEE TVCG journal (IEEE InfoVis’07 Proc.)*, 13(6):1302-1309, 2007.**

Social networks are graphs where the vertices are actors and the edges are relationships. Social networks are commonly known to be globally sparse and locally dense. Thus, a big challenge of social network visualization is to obtain a readable representation for both the overall sparse structure of a social network and its dense communities. In this paper, they proposed a novel visualization called *NodeTrix*. NodeTrix is a hybrid representation of the overall structure of the network based on the node-link diagram where communities can be represented as adjacency matrices. In other words, Intra-community relationships use the adjacency matrix representation while inter-community relationships use normal links. NodeTrix is built using the InfoVis Toolkit and uses its rendering mechanism to create the visualization. They designed a set of interaction techniques to support the manipulation of NodeTrix, and also proposed an animation for smoothly transitioning between node-link diagrams and matrices. Finally, they have illustrated the effectiveness of NodeTrix with a case study of the InfoVis publications data.

**N. Henry and J.-D. Fekete. MatLink: Enhanced matrix visualization for analyzing social networks. *Proc. of the 11th IFIP TC13 International Conference Interact’07*, LNCS 4663:288-302, 2007.**

In this paper, the authors developed MatLink, an enhanced matrix-based graph visualization. MatLink is a matrix representation with links overlaid on its borders and interactive drawing of additional links and highlighting of the cells included in a path from the cell under the mouse pointer. To compare MatLink with traditional matrix-based layout and node-link diagram, they designed a controlled experiment for those 3 visualizations of 6 social networks with 5 different tasks. Five tasks are finding commonNeighbor (an actor directly linked to both two given actors), finding shortestPath (a shortest path linking two given actors), finding mostConnected (the actors with the highest number of relations), finding articulationPoint (a cut point, i.e. an actor linking two sub-graphs), and finding largestClique (the largest set of actors who are all linked to each other). Thirty six subjects participated the experiment. They were asked to complete the tasks correctly as rapidly as possible. The results showed that MatLink significantly outperforms ordinary matrix visualizations for path-related tasks. It also performs as well or better than MAT on other analysis tasks that cannot be effectively performed on NL because of the inherent tendency of social networks to be locally dense. Therefore, MatLink is a good compromise for visualizing and analyzing social networks with a single representation.

**N. Henry, A. Bezerianos, and J.-D. Fekete. Improving the readability of clustered social networks using node duplication. *IEEE TVCG journal (IEEE InfoVis’08 Proc.)*, 14(6):1317-1324, 2008.**

When studying social network visualization, communities can be grouped visually and structurally in a clustered graph. This community representation suffers from two problems: clustering ambiguity and readability. To improve readability and solve ambiguous clustering, they proposed using actor duplications: introducing new nodes to represent aliases of actors. When designing duplication representations, three questions were considered: When to duplicate? How to duplicate? And how to visualize duplications? They investigated two types of duplication: clone and split. An experiment was performed to determine the effects of the different duplications types and designs. The experiment consisted of 6 tasks (actorEstimation, actorConnectivity, communityConnectivity, communityCentrality, articulationPoint, mostConnected), 4 visualizations (noDuplicaion, cloneNode, cloneLink, splitLink), 2 densities (sparse, dense), 2 repetotions, and 12 participants. The results were summarized as a set of design guidelines for node duplications in clustered graph representations. Actor duplication was showed to be very promising in social network analysis and visualization, especially for community centered tasks.

**B. Bach, E. Pietriga, and J.-D. Fekete. Visualizing Dynamic Networks with Matrix Cubes. *Proc. CHI*, ACM, 2014.**

In this paper, the authors introduced the Matrix Cube, a novel visual representation and navigation model for dynamic networks. Matrix Cubes result from stacking adjacency matrix representations of the network at each time step, forming a space time cube. In the cube, a cell exits for each edge between two vertices at a certain time point. They also described how their interactive system for exploration of Matrix Cubes, Cubix, can be used to visualize such dynamic weighted networks by decomposing the cubes into meaningful 2D views. The system implemented seven predefined views: 3DD view, time-projection view, vertex-projection view, time small multiples, vertex small multiples, time-slice-rotation, and vertex-slice-rotation. Among them, two views are more important: time-projection that shows the time steps of the network at a certain time, and vertex-projection that shows the dynamic ego-network of each node. They asked two experts (an astronomer and a neurologist) to use the system to analyze dense dynamic network data from their own work, and got feedback from them. However, the networks they studied in this paper are very small. They did not mentioned the applicability of the system for different scales of networks. In addition, the last two rotation views are not so easy to understand. There is no detail discussion about the slice rotation in neither one of case studies.

**J.-W. Ahn, C. Plaisant, and B. Shneiderman. A Task Taxonomy of Network Evolution Analysis. *IEEE TVCG* 20.3 (2014): 365-376.**

This paper proposed a task taxonomy for network evolution analysis. The authors identified the entities and the properties to be visualized and showed the hierarchy of the temporal features extracted by surveying the existing temporal network visualization studies. Each entity can be classified as the global network, the group, or the node. The properties can be grouped as structural properties and domain attributes. The structural properties reflected the topological nature of the entities. Temporal analysis features defined how to analyze the objects and the properties. They listed tasks by combining the dimensions and their taxonomy and showed examples of the usage of the task taxonomy in a Nation of Neighbors social network analysis project. By comparing the tasks and visualization study examples, they found out which tasks had been covered so far and suggested additions for designing future visualizations. The task taxonomy stated several lessons: the importance of domain attributes, features less explored, higher propensity to the similar individual temporal features, and the lack of means to incorporate different granularity of analysis.

**B. Bach, E. Pietriga, and J.-D. Fekete. GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. *IEEE TVCG* 20.5 (2014): 740-754.**

GraphDiaries is a visual interface designed to improve support for identifying, tracking and understanding changes in dynamic networks. It relies on animated transitions that highlight changes in the network between time steps, thus helping users identify and understand those changes. GraphDiaries features interactive staged animations, non-linear temporal navigation, difference highlighting, small multiples and adapting layout stability. These features can be integrated into any visualization system that supports dynamic node-link diagrams. The design is informed by a taxonomy of tasks related to the exploration of dynamic networks. The tasks are classified based on their categorization along three dimensions: time, type of change and graph entities. GraphDiaries is then evaluated against techniques commonly found in visualization systems for temporal graph navigation. There is a minor increase in task completion time, which is compensated by a significant decrease in error rate in favor of animated transitions.